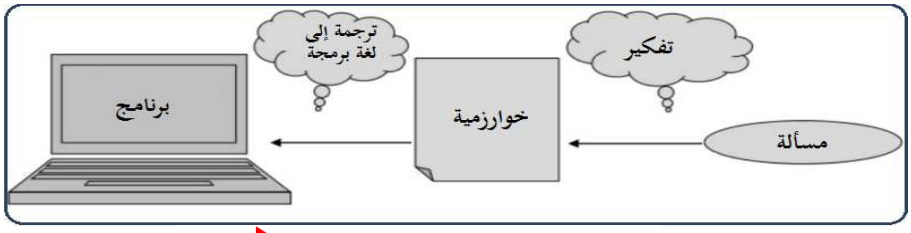
Ministère de l'Enseignement Supérieure et de la Recherche Scientifique
Université d'Alger3
Faculté des sciences Economiques, Sciences Commerciales et Sciences de Gestion
Département des Sciences Economiques

وزارة التعليم العالي والبحث العلمي
جامعة الجزائر 3
كلية العلوم الاقتصادية والعلوم التجارية وعلوم التسيير
قسم العلوم الاقتصادية

مطبوعة موجهة لطلبة السنة الثانية جذع مشترك LMD

بعنوان:

محاضرات في الإعلام الآلي2، تمارين ومسائل محلولة



من إعداد الأستاذ
عبد النور قبايلي

السنة الجامعية 2021-2020

تمهيد:

شهد عالمنا المعاصر تطورا تكنولوجيا هائلا، حيث تسارعت خطواته وتعاظمت، وانتقل من عالم مادي يخضع للعديد من القوانين والأعراف والاختلافات، إلى عالم افتراضي اضمحلت فيه كل الحدود والفروقات، ويرجع الفضل في هذا كله إلى تكنولوجيات الإعلام والاتصال التي سمحت بربط هذا العالم بشبكة عالمية رقمية، فأصبح العالم في خضمها قرية صغيرة متقاربة الحدود والأقاليم، ولعل أهم عنصر وأكثره تأثيرا على هذا التطور الحاصل، هو مختلف البرامج وتطبيقات الحاسوب التي صاحبت هذا التحول نحو العالم الرقمي والافتراضي.

وقد أضحى هذا العصر ينعت في الوقت الراهن بعصر الثورة المعلوماتية أو عصر الثورة الصناعية الرابعة (Fourth Industrial Revolution) (4th IR) والتي تعتبر المعلوماتية عموده الفقري، وحجر أساسه الذي يقوم عليه، فقد تغيرت فيه مفاهيم عدة، إذ لم يصبح الأمي ذاك الذي لا يعرف القراءة والكتابة، بل الأمي في عالمنا اليوم هو الذي لا يعرف التعامل مع برامج وتطبيقات الإعلام الآلي، ولا يعرف التعامل مع مختلف أجهزة وأدوات المعلوماتية.

ووعيا منها بضرورة إلمام كل طلبة الجامعات والمدارس العليا الجزائرية، بمختلف تخصصاتهم وشعبهم، بمبادئ وأبجديات الإعلام الآلي، أدرجت الوزارة الوصية مقياس الإعلام الآلي في مختلف برامج التعليم العالي لكل التخصصات والشعب والمستويات، ولم تشذ تخصصات العلوم الاقتصادية والعلوم التجارية وعلوم التسيير عن هذا الاختيار، فمقياس الإعلام الآلي يعتبر من المقاييس التي يدرسها طالب الليسانس انطلاقا من السنة الأولى ووصولا إلى السنة الثالثة.

وانطلاقا من تجربتي الطويلة في تدريس مقياس الإعلام الآلي2 لطلبة قسم العلوم الاقتصادية، ارتأيت أن أضع بين أيديهم هذه المطبوعة والتي وسمتها بـ محاضرات في الإعلام الآلي2، تمارين ومسائل محلولة، راجيا من العلي القدير أن أكون قد وفقت في تبسيط المفاهيم وتوضيح الأبجديات الخاصة بالخوارزميات، وحتى تتم الاستفادة الكاملة من هذه المطبوعة، لا بد لطلبتنا الأعزاء الاهتمام بهذا المقياس، وذلك من خلال الحضور المنتظم لكافة محاضراته وأعماله الموجهة، مع التحضير المسبق لسلاسل التمارين ومناقشة الحلول المقترحة مع الأساتذة لمحاولة إثرائها واقتراح حلولا أخرى، علما بأنه لا يمكن للكتاب أو المطبوعة أن يحلا محل الأستاذ.

وفي الأخير، أتوجه بالشكر الجزيل إلى كل من مدّ يد العون وساهم من قريب أو من بعيد في إنجاز هذا العمل سائلا المولى السميع العليم القبول والتوفيق، فهو الموفق والهادي إلى سواء السبيل.

د. عبد النور قبايلي

فهرس المحتويات

الموضوع	الصفحة
تمهيد	أ
فهرس المحتويات	I
المقدمة	02
الفصل الأول: مفهوم الخوارزميات والمتغيرات والتعليمات الأساسية البسيطة	
تمهيد	05
I. مدخل إلى الخوارزميات	05
1.I الهدف من استعمال الخوارزميات في العلوم الاقتصادية والعلوم التجارية وعلوم التسيير	05
2.I مفهوم البيئة المعلوماتية	06
3.I تعريف الخوارزمية	07
4.I الشكل العام للخوارزمية	09
5.I مجالات استعمال الخوارزميات	10
6.I تعريف البرمجة والبرنامج	10
II. المتغيرات	12
1.II تعريف المتغير	12
2.II.تعريف الثابت	12
3.II.أنواع المتغيرات	13
4.II.كيفية التصريح بالمتغيرات والثوابت داخل الخوارزميات	16
5.II.كيفية تحليل الصيغ الرياضية في الخوارزمية	17
6.II.استعمال الأولويات لحساب الصيغ الرياضية	17
III.تمارين مقترحة	19
IV. حلول التمارين	19
V. التعليمات الأساسية البسيطة في الخوارزميات–تعليمة التخصيص، القراءة والكتابة	22
1.V تعريف التعليمة	22
2.V تعليمة التخصيص أو إعطاء القيمة (Affectation)	22
3.V. تعليمات القراءة والكتابة أو الإدخال والإخراج	24
VI. تمارين مقترحة	27
VII. حلول التمارين	29
الفصل الثاني: التعليمات الأساسية في الخوارزميات– التعليمات الشرطية	

تمهيد	35
I. التعليمة الشرطية البسيطة	35
I.1 تعريف التعليمة الشرطية البسيطة	35
I.2 كيفية عمل التعليمة	35
II. التعليمة الشرطية المتناوبة	37
II.1 تعريف التعليمة الشرطية المتناوبة	37
II.2. كيفية عمل التعليمة	37
III. التعليمة الشرطية المتداخلة	39
III.1 تعريف التعليمة الشرطية المتداخلة	39
III.2 كيفية عمل التعليمة	40
IV. التعليمة الشرطية selon	44
IV.1 تعريف التعليمة الشرطية selon	44
IV.2 كيفية عمل التعليمة	45
V. تمارين مقترحة	46
VI. حلول التمارين	49
الفصل الثالث: التعليمات الأساسية في الخوارزميات– التعليمات التكرارية	
تمهيد	63
I. التعليمة التكرارية بعداد أو الحلقة بعداد (التعليمة pour / for)	63
I.1 طريقة كتابة الحلقة pour	64
I.2 طريقة عمل الحقة pour	64
II. التعليمة التكرارية بشرط الخروج	67
II.1 التعليمة التكرارية (الحلقة) tant que	67
II.2. التعليمة التكرارية (الحلقة) répéter jusqu'à	71
II.3 كيفية الاختيار بين استعمال الحلقات الثلاثة (pour, tant que, répéter jusqu'à)	74
III. تمارين مقترحة	75
IV. حلول التمارين	80
الفصل الرابع: مسائل محلولة في الخوارزميات	
تمهيد	99
I. المسألة الأولى	99
II. المسألة الثانية	101
III. المسألة الثالثة	104

IV. المسألة الرابعة	107
V. المسألة الخامسة	110
VI. المسألة السادسة	112
VII. المسألة السابعة	116
VIII. المسألة الثامنة	120
IX. المسألة التاسعة	122
X. المسألة العاشرة	126
XI. المسألة الحادي عشرة	128
الخاتمة	132
قائمة المراجع	135

المقدمة

مقدمة:

بالرغم من القدرات الحسابية الهائلة للحواسيب ودقتها المتناهية وقدرتها التخزينية المعتبرة وسرعتها التي لا يمكن مضاهاتها، إلا أنها تبقى مجرد آلة إلكترونية لا تستطيع أن تنجز أي عمل بمفردها، بل تنفذ ما يطلب منها فحسب بكل دقة وأمانة، وذلك من خلال مجموعة من التعليمات تكون على شكل برامج يتم تزويدها بها.

وتعتبر الخوارزميات أهم فروع المعلوماتية وأول مرحلة من مراحل البرمجة، وينسب مصطلح الخوارزميات إلى العالم المسلم أبو جعفر محمد بن موسى الخوارزمي، والذي عاش في القرنين الثامن والتاسع ميلادي (780م-850م) في عصر الخليفة العباسي المأمون، وعرف بالعديد من المؤلفات في الرياضيات والفلك والجغرافية ويعتبر مؤسس علم الجبر الخوارزميات.

وحاولنا من خلال الفصول الأربعة لهذه المطبوعة التطرق إلى تعريف الخوارزميات ومجال استعمالاتها، ثم تناولنا أهم مكونات الخوارزميات والكيفية الصحيحة لكتابتها وكذا كل خصائصها، وهذا انطلاقا من أنواع المتغيرات في المعلوماتية ووصولا إلى مختلف التعليمات المستعملة في كتابة الخوارزميات مرورا بكيفية كتابة مختلف الصيغ الرياضية في المعلوماتية.

وقد تم تخصيص الفصل الأول من هذه المطبوعة لدراسة مدخل للخوارزميات تم من خلاله التطرق إلى الهدف من تدريس هذا المقياس، وبعض التعاريف الخاصة بالبيئة المعلوماتية (environnement informatique) وتعريف الخوارزميات ولغات البرمجة والفرق بينها، وكذا التعريف بالمتغيرات في الخوارزميات، والتطرق إلى مختلف أنواعها وكيفية استعمالها والتصريح بها، كما تم التطرق أيضا إلى تعريف الثوابت والحاجة لاستعمالها والفرق بينها وبين المتغيرة، وعرجنا بعدها إلى طريقة تحليل وحساب مختلف العلاقات الرياضية باستعمال الأولويات، وختمنا هذا الفصل بدراسة التعليمات الأساسية البسيطة في الخوارزميات، انطلاقا من تعليمة التخصيص (Affectation) ووصولا إلى تعليمة القراءة والكتابة أو تعليمة الإدخال والإخراج (Entrée/Sortie).

الفصل الثاني من هذه المطبوعة خصصناه لدراسة التعليمات الشرطية في الخوارزميات، فتطرقنا ابتداء إلى التعليمة الشرطية البسيطة، ثم انتقلنا إلى التعليمة الشرطية المتناوبة، وعرجنا على دراسة التعليمة الشرطية المتداخلة باعتبارها أكثر التعليمات استعمالا، ثم ختمنا هذا الفصل بالتعليمة الشرطية selon لما لها من دور فعال في كتابة الخوارزميات المتعددة الشروط.

الفصل الثالث تم تخصيصه للتعليمات التكرارية، فتم التعريف بالتعليمة التكرارية ذات العداد pour، ثم انتقلنا إلى التعليمات التكرارية بشرط الخروج tant que و répéter jusqu'à.

الفصل الرابع من هذه المطبوعة تضمن مجموعة من المسائل المستقاة من مختلف الامتحانات الخاصة بمقياس الإعلام

الآلي2 لطلبة السنة الثانية جذع مشترك علوم اقتصادية للسنوات الماضية مع الحلول المقترحة، وكذا مناقشة مفصلة لهذه الحلول.

كما تجدر الإشارة أن الفصول الثلاثة الأولى لهذه المطبوعة قد احتوت على أمثلة توضيحية، ومجموعة من التمارين المحلولة التي تساعد الطلبة على الفهم الجيد لأبجديات الخوارزميات، وكيفية تطبيق الجانب النظري على أمثلة تطبيقية مستمدة من الواقع.

الفصل الأول

الفصل الأول: مفهوم الخوارزميات والمتغيرات والتعليمات الأساسية البسيطة

تمهيد:

كما أسلفنا الذكر في مقدمة هذه المطبوعة، فإن الخوارزميات تأسست في القرن التاسع الميلادي من طرف العالم المسلم الخوارزمي، فهو أول من استعمل طريقة تبسيط المسائل الرياضية المعقدة إلى مجموعة من العمليات المتتالية، والتي تؤدي في الأخير إلى حل تلك المسائل، وألف الخوارزمي مجموعة من الكتب والمؤلفات، أشهرها كتاب الجبر والمقابلة والذي ترجم إلى اللاتينية واستفاد منه كل علماء الرياضيات الذين جاؤوا من بعده، ولدوره الفعال في إثراء علم الجبر بطريقته المبتكرة في حل المسائل عن طريق التسلسل المبسط، خُلِّد اسمه وارتبط بالخوارزميات، فلا تكاد تبحث عن مصطلح الخوارزميات بكل لغات العالم حتى يُذكر العالم المسلم الخوارزمي.

I. مدخل إلى الخوارزميات

سنحاول من خلال هذا الجزء التطرق إلى مجموعة من المفاهيم والتعاريف الخاصة بالخوارزميات وكيفية كتابتها وطريقة استعمالها، ودواعي استخدامها.

I.1 الهدف من استعمال الخوارزميات في العلوم الاقتصادية والعلوم التجارية وعلوم التسيير:

عند اقتنائنا لجهاز إلكتروني أو كهرومنزلي لا بد من وجود دليل استعمال يوضح الخطوات التي نتبعها من أجل تركيب الجهاز وتوصيله بالكهرباء وتشغيله تشغيلا صحيحا يضمن حسن سيره، كما يتبع الكثير منا خطوات معينة ومرتبة ودقيقة من أجل تحضير أكلات معينة أو حلويات تقليدية أو عصرية، وقد يصادف بعضنا أشخاصا يبحثون عن موقع معين فيقدم لهم يد العون من خلال إرشادهم حول الكيفية التي تمكنهم من الوصول إلى وجهتهم دون غموض أو التباس، في كل هذه الحالات نحن نستعمل الخوارزميات ولكن دون أن نشعر أننا نستعملها.

الهدف من استعمال الخوارزميات في ميدان العلوم الاقتصادية والعلوم التجارية وعلوم التسيير، هو أن يصبح الطالب ملما بمبادئ الخوارزميات، بحيث يستطيع عند الانتهاء من دراسة هذا المقياس، أن يكتب خوارزميات متوسطة التعقيد تكفل له حل مسائل متعلقة بميادين اختصاصه المتنوعة كالمحاسبة، التحليل المالي، التسيير المالي، التحليل الاقتصادي..إلخ.

وفي هذا السياق فالطالب ليس مطالبا بإتقان كل مفاصل الخوارزميات كالمبرمج المحترف والمتخصص، بل يكفيه أن يكون على دراية كافية بالمبادئ العامة التي تمكنه من فهم ما يقوم به المبرمج المحترف، بحيث يستطيع أن يتعامل معه ويفهم ما يقوم به، فعلى سبيل المثال، المبرمج المحترف الذي يحاول تصميم برنامج خاص بالمحاسبة سيستعين لا محالة بخبير

في المحاسبة، بحكم إلمامه بكل خبايا هذا التخصص، أما المبرمج فهو تقني متخصص في البرمجة والمعلوماتية وقليل الإلمام بمبادئ المحاسبة، فيحدث تكامل بين المبرمج وخبير المحاسبة من أجل تصميم وإنجاز البرنامج السالف الذكر، ومن أجل إنجاح هذا التكامل لا بد لخبير المحاسبة أن يكون على علم بأبجديات الخوارزميات ليسهل له التعامل مع المبرمج ومد يد العون له وتوجيهه واقتراح حلول محاسبية قد تثري البرنامج، هذا الأخير سيوجه إلى المحاسبين وخبراء المحاسبة وليس للمبرمجين المحترفين وأصحاب اختصاصات المعلوماتية.

من خلال المثال السابق يظهر جليا الدور المنتظر من طلبتنا الأعزاء لأجل الاهتمام بهذا المقياس، والذي سيمكنهم من تولي مسؤوليات ومهام إدارية في المؤسسات التي سيشتغلون بها، إضافة إلى تفضيلهم على الموظفين الذين ليس لهم دراية باستعمال تطبيقات وبرامج المعلوماتية عند توفر مناصب عليا أو ترقيات.

2.I مفهوم البيئة المعلوماتية:

كثيرا ما نسمع بمصطلح البيئة المعلوماتية (Environnement Informatique)، لكن القليل منا من يعرف ماهيته، بالرغم من أن مفهومه جد بسيط، بحيث لا يتعارض والقواعد الكونية التي خلقها الله عزَّ وجلَّ، فبالتشابه مع البيئة في الطبيعة، فالسمكة التي تعيش في الماء لا يمكنها العيش خارجه، وبذلك يعتبر الماء بيئتها الطبيعية التي تعيش فيها ولو غيرنا هذه البيئة لماتت السمكة ولم تستطع التأقلم.

في المعلوماتية أو في الإعلام الآلي، البيئة بالنسبة لتطبيق معين أو برنامج ما (Application ou logiciel)، هي مجموعة الأجهزة والمعدات الإلكترونية وهو ما يصطلح على تسميته بـ (HardWare) وبرامج نظام التشغيل التي تسير وتشغل تلك الأجهزة وهو ما يعرف بأنظمة التشغيل أو أنظمة الاستغلال (Systèmes d'exploitation)، فمن أجل استغلال واستعمال أي تطبيق معلوماتي أو برنامج حاسوبي لا بد من توفر هذين العنصرين وهو ما يعرف بالثنائية (HardWare-SoftWare)، بحيث يمثل الـ HardWare الجزء المادي وهو الأجهزة والمعدات، بينما يمثل الـ SoftWare الجزء البرمجي والتطبيقي وهو كل البرامج والتطبيقات التي ستستعمل هذه الأجهزة. وتجدر الإشارة إلى أن جهاز الحاسوب أو ما اصطلح على تسميته بـ HardWare هو مكون من المعالج (microprocesseur CPU) الذي تسند له عملية المعالجة والتنفيذ، والذاكرة المركزية إضافة إلى الذاكرة الثانوية وأدوات الإدخال والإخراج.

وكمثال عن البيئة في المعلوماتية، تطبيق المجدول الغني عن التعريف إكسال (Tableur Microsoft Excel) يحتاج إلى بيئة معلوماتية مكونة من جهاز حاسوب شخصي (PC) أو حاسب موزع (Serveur) وهو ما يمثل الـ HardWare ونظام تشغيل من نوع MicroSoft Windows 8, Windows 10, Winows Server وهو ما يمثل الـ SoftWare، فلو نحاول تشغيل واستغلال برنامج إكسال على حواسيب من نوع ماكينتوش Mac لا يشتغل هذا البرنامج، وحتى وإن استعملنا حاسوب شخصي PC لا يستعمل نظام تشغيل MicroSoft Windows ويستعمل نظام تشغيل آخر كـ Linux لا يمكن تشغيل برنامج الإكسال عليه، لأنه وكما أسلفنا البيئة في المعلوماتية هي تكامل الثنائية (HardWare-SoftWare).

I.3 تعريف الخوارزمية:

هي المراحل أو الخطوات اللازم إتباعها من أجل حل مسألة معينة بعيدا عن أي لغة من لغات البرمجة، فهي إذا طريقة تفكير متسلسل وممنهج لحل مسألة ما من خلال تقسيمها إلى مراحل متتالية ومترابطة بكيفية منطقية. فالخوارزمية هي مجموعة من الخطوات تنفذ حسب **ترتيب معين** للحصول على النتيجة المرجوة، ويعتبر الترتيب المتبع في كتابة أي خوارزمية ذا أهمية بالغة بحيث عند تغيره يمكن للنتيجة المتحصل عيها أن تصبح غير منتظرة وعشوائية.

كما تعرف الخوارزمية أيضا على أنها مجموعة من الخطوات المنطقية المتسلسلة التي يتم تنفيذها لحل مسألة معينة أو مشكلة ما، وهي طريقة فعالة وموثوقة تصف بدقة ووضوح حل مسألة معينة دون التباس أو غموض.

كل خوارزمية يجب أن يتوفر فيها شرطان أساسيان هما:

- الطريقة، وهي كيفية التفكير لحل المسألة؛
- الترتيب، وهو تسلسل المراحل واحدة تلو الأخرى.

يوجد نوعان أساسيان من الخوارزميات: الخوارزميات الحسابية والخوارزميات غير الحسابية، وتعتبر هذه الأخيرة الأكثر استعمالا والأكثر حضورا، فعلى سبيل المثال معالجات النصوص والمدققات اللغوية وإدارة أنظمة المعلومات هي كلها أسست عن طريق خوارزميات غير حسابية، أما البرامج الحسابية كالمجدولات والبرامج الإحصائية أو برامج الهندسة المدنية والميكانيكية هي برامج أساسها خوارزميات حسابية تستعمل المعادلات الرياضية والصيغ الحسابية.

إتقان الخوارزميات يحتاج إلى صفتين أو ميزتين يجب أن يتصف بهما كل من يرغب في ذلك، تتمثل هاتان الصفتان في:

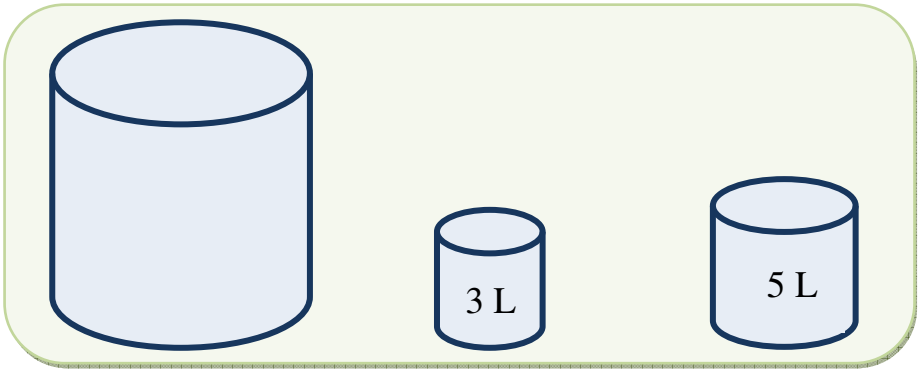
- الحدس (Intuition)، وهو سرعة البديهة، بحيث تسمح لكاتب الخوارزمية من أن تكون له معرفة مسبقة بالعمليات التي يجب استعمالها من أجل الحصول على النتائج المبتغاة أو المرجوة، فالحدس هو موهبة يمكن اكتسابها مع الوقت ومع كثرة الممارسة والخبرة، فمجموعة المسائل والعمليات التي قد تبدو في البداية معقدة وصعبة للغاية، تصبح مع مرور الوقت وكثرة المحاولات والحلول عفوية (spontané).
- التجريب، يعتبر تجريب الخوارزميات من صفات المبرمج الناجح والمحترف، إذ يحاول كاتب الخوارزمية تنفيذها خطوة بخطوة للتأكد من عدم احتواءها على أخطاء حسابية أو منطقية، وتتم عملية التجريب عادة بمراقبة الحالات الشاذة أو الحالات القليلة الحدوث، فعند تجريب أي خوارزمية يجب مقارنة النتائج المتحصل عليها مع النتائج المرجوة، فعند اختلاف النتائج يجب مراجعة كتابة الخوارزمية وتصحيحها.

يجب التنويه إلى وجود العديد من الخوارزميات الخاصة بالمسألة الواحدة، ولكن توجد الخوارزمية المثلى والفعالة، وتوجد الخوارزمية الأقل فعالية، وتكمن فعالية الخوارزمية ومثاليتها في الوقت المستغرق من خلال عدد العمليات، والذاكرة المستعملة، فالوقت يعتبر عاملا جد مؤثر على فاعلية الخوارزمية ومثاليتها فهو حسب المثل الأمريكي يساوي المال (Time is money)، أما الذاكرة فهي عنصر محدود حتى وإن توفرت في أيامنا هذه بسعات عالية في كل الحواسيب إلا أنها تبقى محدودة، من أجل هذا، فالخوارزمية المثلى هي التي تُنجَز في وقت قياسي مع استعمال أقل قدر من الذاكرة.

ولتبيان مسألة تعدد الخوارزميات لحل المسألة نفسها، نسوق المثال التالي والذي يوضح تنوع الخوارزمية واختلافها لحل المسألة نفسها.

نريد وضع 4 لتر من الماء في وعاء سعته 5 لتر ولا يحتوي على تدرجات تبين حجم الماء، وهذا باستعمال وعاء آخر سعته 3 لتر، كما هو موضح في الشكل التالي:

الشكل (1): مثال تطبيقي عن تعدد الحلول لمسألة واحدة



الحل الأول:

1. ملأ الدلو 3L؛
2. إفراغ الدلو 3L في الدلو 5L فيصبح الدلو 5L يحتوي على 3 لتر من الماء والدلو 3L فارغ؛
3. ملأ الدلو 3L من جديد؛
4. إفراغ الدلو 3L في الدلو 5L فيصبح الدلو 5L ممتلئ ويبقى في الدلو 3L لتر واحد؛
5. إفراغ محتوى الدلو 5L من الماء فيصبح فارغا (إتلاف الماء)؛
6. إفراغ محتوى الدلو 3L (1 لتر) في الدلو 5L؛
7. ملأ الدلو 3L من جديد؛
8. إفراغ محتوى الدلو 3L في الدلو 5L الذي كان فيه 1لتر فيصبح في الدلو 5L 4 لتر وبهذا وصلنا لنهاية الحل؛

الحل الثاني:

1. ملأ الدلو 5L؛
2. إفراغ الدلو 5L في الدلو 3L فيبقى في الدلو 5L 2 لتر بينما يمتلئ الدلو 3L؛
3. إفراغ محتوى الدلو 3L من الماء فيصبح فارغا (إتلاف الماء)؛
4. إفراغ محتوى الدلو 5L (2 لتر) في الدلو 3L فيصبح يحتوي على 2 لتر؛
5. ملأ الدلو 5L من جديد؛
6. إفراغ الدلو 5L في الدلو 3L والذي كان فيه 2 لتر من الماء فيمتلئ الدلو 3L (بحيث كان ينقصه 1 لتر) ويبقى في الدلو 5L 4 لتر وبهذا وصلنا لنهاية الحل.

الملاحظ في الحل الأول أنه تم بعد 8 خطوات وتضييع (إتلاف) 5 لتر من الماء، أما الحل الثاني فجاء بعد 6 خطوات فقط وتضييع (إتلاف) 3 لتر من الماء. نستطيع القول أن الحل الثاني هو الحل الأمثل بحيث تم بعد 6 خطوات وتضييع 3 لتر من الماء فقط، وبهذا يمكن تشبيه الماء بمساحة الذاكرة المستعملة وعدد الخطوات بالوقت، مع أن الحلين يؤديان لنفس النتيجة.

I.4 الشكل العام للخوارزمية:

تكتب الخوارزمية على الشكل التالي:

Algorithme Nom_Algorithme
<Partie_déclaration>
début
<Partie_traitement>
fin

الجزآن المهمان في كل خوارزمية هما:

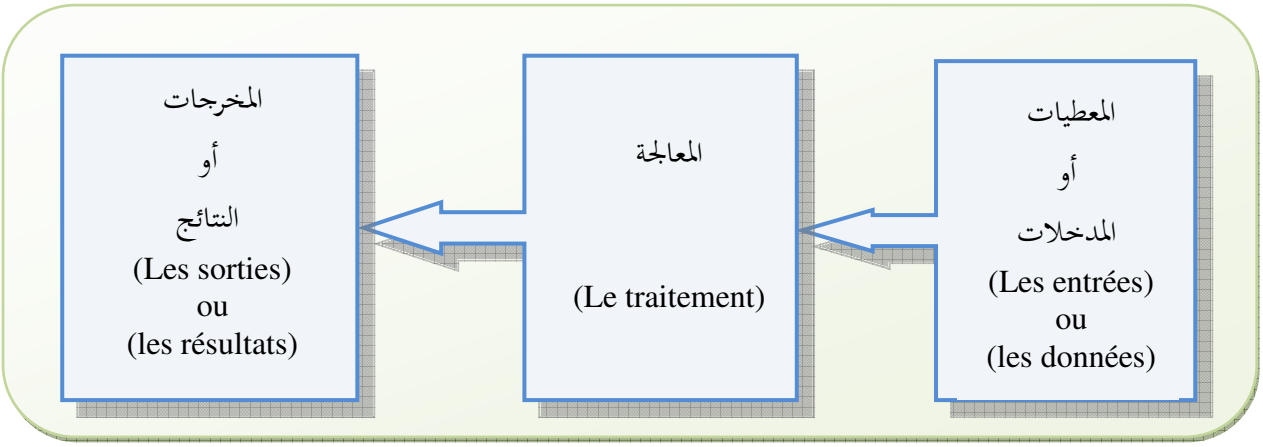
- جزء الإعلانات، أين يتم التصريح بكل المتغيرات أو الثوابت المستعملة في الخوارزمية؛
- جزء المعالجة، وهو يحتوي على العمليات الحسابية التي تتم على المتغيرات والثوابت المصرح بها، وكل المعالجة اللازمة لحل المسألة المطروحة.

تتميز الخوارزميات بالخصائص التالية والتي منها ما هو مشترك مع كل الخوارزميات الأخرى، ومنها ما هو خاص بمجموعة محددة، وتتلخص هذه الخصائص في:

- التخصيص أو إعطاء القيمة الأولية (أو قراءة-إدخال المعطيات)؛
- التسلسل في الإجراء؛
- الاختيار؛
- التكرار؛
- إخراج النتائج.

ويعتبر الاختيار والتكرار خصائص لا تتعلق بكل الخوارزميات ولكن تتعلق بمجموعة معينة من الخوارزميات والتي فيها الاختيار وتكرار العمليات، أما إدخال المعطيات والتسلسل في الإجراء وإخراج النتائج فهي خصائص مشتركة مع كل أنواع الخوارزميات، وعلى العموم يمكن أن تأخذ الخوارزميات الشكل التالي:

الشكل (2): مبدأ عمل الخوارزميات



5.I مجالات استعمال الخوارزميات:

نظرا لأهميتها ودورها الفعال في عالم المعلوماتية، فالخوارزميات هي أداة لا يمكن الاستغناء عنها، فهي أول ما يتطرق إليه كل مبرمج أو كل مهتم بميدان الإعلام الآلي، ومن هذا المنطلق لا تكاد تجد مطور برمجيات أو مصمم تطبيقات إعلام آلي ليس له الدراية الكافية والإلمام الوافي بمبادئها، لذا فهي تستعمل في المجالات التالية:

- قبل كل عملية برمجة، تستعمل الخوارزميات كمرحلة ابتدائية يتم فيها تصميم البرنامج، ويُرجَع إليها في حالات صيانة البرنامج أو تعديله، بحيث من الأفضل قراءة الخوارزمية وفهمها على قراءة البرنامج الذي يتسم بالتعقيد ووجود العديد من المحددات الخاصة بلغة البرمجة، أما الخوارزمية فتتصف بالبساطة وبلغتها الطبيعية الذي يستطيع أن يفهمها كل مبرمج حتى وإن كان لا يتقن لغة البرمجة التي كتب البرنامج محل الصيانة أو التعديل؛
- كما تستعمل الخوارزميات لتحديد الوقت التقريبي الذي يحتاجه البرنامج لكي ينفذ، وهذا عن طريق حساب درجة التعقيد الخاصة بالخوارزمية وعدد التعليمات التي تحتويها؛
- وتستعمل الخوارزميات أيضا لتحديد مساحة الذاكرة التي يحتاجها البرنامج من خلال حساب المساحة التقريبية لكل متغيراته ومخرجاته ومدخلاته؛
- كما تستعمل الخوارزميات لاختيار أفضل وأمثل البرامج من حيث الوقت وحجم الذاكرة المستعملة؛
- وأخيرا تستعمل الخوارزميات لاكتشاف الأخطاء، خاصة المنطقية منها، وتصحيحها قبل عملية البرمجة.

6.I تعريف البرمجة والبرنامج:

البرمجة هي طريقة تواصل مع الحاسوب من أجل طلب القيام بمهام معينة، والبرنامج هو مجموعة من المعطيات والأوامر والتعليمات مكتوبة بطريقة وبلغة معينة يزود بها الحاسوب لتنفيذ مهمة محددة ودقيقة.

عند كتابة برنامج ما، يجب مراعاة لغة البرمجة التي كتب بها هذا البرنامج وعدم نسيان أي علامة أو ملاحظة، فالبرنامج موجه لجهاز إلكتروني لا يستطيع أن يفكر أو يقرر لوحده، عكس الخوارزمية التي توجه للإنسان الذي يستطيع أن يتنبأ بما يريد صاحب الخوارزمية إبلاغه به، وحتى لو وجد خطأ في كتابة كلمة ما أو تعليمة ما، يستطيع قارئ

الخوارزمية استيعاب ذلك الخطأ وتصويبه، فمثلا في لغة البرمجة باسكال (Pascal) أو سي (C) الفاصلة المنقوطة "; " جد مهمة بحيث يُفْصَل بها بين التعليمات، لذا فبمجرد نسيان كتابتها، لا يتم تنفيذ البرنامج ويبلغك بوجود خطأ في الكتابة.

مثال عن برنامج بسيط مكتوب بلغة باسكال:

```
PROGRAM bonjour;
BEGIN
{Affiche Hello World à l'écran}
writeln (' Hello World ');
END.
```

يقوم هذا البرنامج البسيط بكتابة كلمة (Hello World) على الشاشة، أما الجملة الموجودة بين حاضنتين {Affiche Hello World à l'écran} فهي مجرد تعليق لا يأخذه البرنامج بعين الاعتبار وهو مدرج من أجل تسهيل قراءة البرنامج وفهمه من قبل المبرمج أو شخص آخر.

فالبرمجة هي إذًا الصيغة أو الكيفية التي تمكن للحاسوب أن يتعامل مع البيانات والمعطيات والأوامر المدخلة، من أجل القيام بمهمة معينة والحصول على النتائج المنتظرة أو المرجوة. فالأوامر هي مجموعة متسلسلة من الأفعال أو التعليمات (Instructions) (الجمع، الطرح، الضرب، الكتابة،....إلخ)، والمهمة هي الهدف من البرنامج والغاية منه.

على سبيل المثال، حساب معدل طالب في السنة الجامعية هي مهمة معينة وهي الهدف من البرنامج، أما المعطيات والبيانات، فهي النقاط المتحصل عليها في مختلف المقاييس وكذا معاملات المقاييس، وأما الأوامر والتعليمات فهي كل عمليات الجمع والضرب والقسمة التي تتم على البيانات المدخلة فمعدل الطالب يحسب بضرب كل مقياس في معامله ثم جمع النتائج وقسمتها على مجموع المعاملات، وتجدر الإشارة إلى أن الطرق المتبعة من أجل إنجاز وتنفيذ مهمة معينة تكون غالبا متعددة وغير وحيدة.

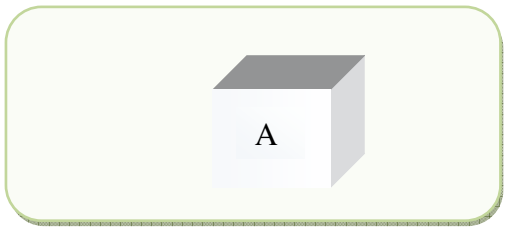
II. المتغيرات

تكتسي المتغيرات أهمية بالغة في الخوارزمية، بحيث لا تكاد تجد خوارزمية خالية منها، وعلى غرار المتغير في الرياضيات فالمتغير في الخوارزمية يغير من قيمه حسب تواجده داخل الخوارزمية وحسب نوعه وحسب التعليمة التي تتم عليه.

II.1 تعريف المتغير:

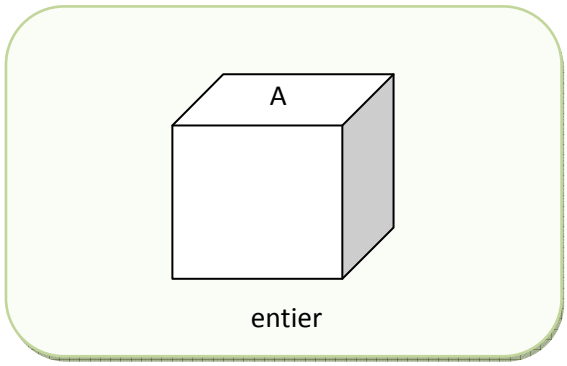
المتغير في الخوارزميات يمكن تمثيله بعلبة تتعرف عليها الخوارزمية (الحاسوب) بواسطة ملصقة أو اسم (Etiquette).

الشكل (3): مبدأ المتغير في الخوارزمية



لنتمكن من الوصول إلى محتوى هذه العلبة نستعمل الاسم (الملصقة)، والجدير بالذكر أن هذه العلبة تحتفظ باسمها ونوعها طوال تواجدها داخل الخوارزمية، لكن القيمة المحفوظة فيها تتغير بتوالي التعليمات لذا فهي تأخذ آخر قيمة كما هو مبين في الشكل التالي:

الشكل (4): طريقة حفظ المتغير في الخوارزمية



والمتغير عمليا هو منطقة في الذاكرة العشوائية للكمبيوتر (RAM Random Access Memory) ، لها اسم معطى ونوع معين. يسمح اسم المتغير بالوصول إلى محتويات منطقة الذاكرة، بينما يحدد النوع طبيعة ما يمكن تخزينه في منطقة الذاكرة (عدد صحيح، حقيقي، حرفي، منطقي،...إلخ). عادة ما تمثل المتغيرة بواسطة علبة، أعلاها نضع اسمها وهو يمثل اسم المتغيرة، وأدناها نضع نوعها، أما داخلها فنضع القيمة المخزنة داخل المتغيرة.

II.2.تعريف الثابت:

عكس المتغير الذي يأخذ مجموعة من القيم ويتغير بتغير مراحل الخوارزمية، يعتبر الثابت قيمة لا تتغير بتغير مراحل

الخوارزمية ولا يجوز تغييرها سواء عن طريق القراءة أو عن طريق الحساب، فهي قيمة تتعلق باسمها، بحيث عند بداية الخوارزمية يعطى الثابت اسما وقيمة يبقيان متلازمان طوال الخوارزمية، ويوجد نوعان من الثوابت هما:

- الثوابت العددية؛
- والثوابت الحرفية.

فالثوابت العددية هي قيمة عددية مرتبطة باسمها، بحيث تبقى ثابتة داخل الخوارزمية ولا نستطيع تغييرها، وحتى لو حاولنا تغييرها داخل أي لغة برمجة فسنصطدم بخطأ في التنفيذ يؤكد لنا استحالة تغير قيمة الثابت، فمثلا إذا استعملنا الثابت PI = 3.14 ستضل قيمة PI تكافئ 3.14 ولا تتغير ما دامت الخوارزمية لم تنته، وعند استعمال PI في أي عملية حسابية فإننا سنستعمل القيمة 3.14 التي هي ملازمة لها.

أما الثوابت الحرفية فلها نفس التوجه غير أنها تحتوي على قيمة حرفية ترتبط بالاسم ولا تتغير طوال الخوارزمية، فمثلا إذا استعملنا الثابت الحرفي universite = "Alger3"، تظل قيمتها ثابتة طيلة مراحل الخوارزمية، وأين نجد كلمة universite فإنها تعوض بـ"Alger3".

ملاحظة هامة:

عند اختيار اسم المتغير أو الثابت، يجب مراعاة القواعد التالية:

- يجب أن يبدأ الاسم وجوبا بحرف لاتيني، كما يجب اجتناب الأسماء التي تحتوي على حروف التشكيل كـ (é, è, ê, à, ä,….etc)؛
- يجب أن لا يحتوي الاسم على الحروف الخاصة أو العوامل الحسابية، كـ &، !، @، #، +، *، ....إلخ، ويسمح باستعمال الرمز "_" ؛
- لا يجب أن يتكون الاسم من كلمتين، كـ Nom Etudiant ويفضل في هذه الحالة كتابة NomEtudiant أو Nom_Etudiant
- لا يجب استعمال اسم متغير يتشابه مع أحد الأسماء التابعة للخوارزمية (mots réservés) كـ fin وsi وalors.... إلخ، كما لا يجب أن نستعمل اسم متغير يتشابه مع اسم الخوارزمية.

يجب التنويه بأن يعود الطالب نفسه على استعمال أسماء متغيرات لها علاقة بالموضوع المراد دراسته، بدل استعمال أسماء جوفاء أو صماء.

II.3.أنواع المتغيرات:

تحتوي الخوارزميات على عدة أنواع من المتغيرات منها ما هو بسيط ويشبه إلى حد كبير تلك الموجودة في الرياضيات ومنها ما هو مركب وخاص بالمعلوماتية، وتوجد أساسا أربعة أنواع من المتغيرات هي: النوع الصحيح، النوع الحقيقي، النوع المنطقي والنوع الحرفي.

II.1.3.النوع الصحيح:

وهو النوع الذي يستعمل عند الحاجة للتعامل مع أعداد صحيحة لا تحتوي على أجزاء أقل من الواحد وأكبر من الصفر سواء للأعداد الموجبة أو السالبة، وبهذا لا يمكن أن نستعمل النوع الصحيح للتعامل مع أعداد كـ 1.15 أو 0.07، فمثلا إذا كنا نستعمل نظام تشغيل يعمل على 16 بيت bits (2 أوكتي octet أو 2 بايت Byte)* فالنوع الصحيح يأخذ حيز 2 بايت أي 16 بيت وكل بيت يأخذ قيمتين فقط (0 أو 1) إذًا نستطيع كتابة: قيمة 65 536 = 2^{16}، فإذا كان النوع صحيح موجب فالأعداد الممكن كتابتها تبدأ من 0 وتنتهي بـ 65 535 ، بحيث من 0 إلى65 535 يوجد 65 536 عدد (من 0 إلى 9 يوجد 10 أعداد وليس 9) أما إذا النوع الصحيح يحتوي على أعداد موجبة وأخرى سالبة، فيقسَّم المجال من 0 إلى 65 535 على جزأين متساويين يحتوي كل جزء على 65 536/2 أي 32 768 عدد، جزء سالب من 32 768-إلى 1- وجزء موجب من 0 إلى 32 767.

أما في حالة نظام تشغيل يعمل على 32 بيت، فالنوع الصحيح يأخذ حيز 4 بايت أي 32 بيت وكل بيت يأخذ قيمتين فقط (0 أو 1) إذا نستطيع كتابة: قيمة 4 294 967 296 = 2^{32}، وهو عدد ضخم يمكن تقسيمه إلى جزأين في حالة الأعداد الموجبة والسالبة، أو تركه مجالا موجبا فقط من 0 إلى 4 294 967 295.

يكتب النوع الصحيح عند التصريح به داخل الخوارزمية entier أما في لغات البرمجة فكل لغة برمجة لها كتابتها الخاصة، ففي لغة الباسكال Pascal يكتب النوع الصحيح integer أما في لغة سي C فيكتب int.

العوامل (opérateurs) والدوال (fonctions) المستعملة على النوع الصحيح هي: كل العوامل والدوال المعروفة كالجمع والضرب والطرح والقسمة، إضافة إلى الجذر التربيعي والأس والقيمة المطلقة، كل هذه العوامل والدوال تستعمل أيضا على النوع الحقيقي، لكن النوع الصحيح يتميز بنوعين إضافيين من العوامل، هما: Div وMod، بحيث يمثل الـ Div حاصل القسمة الصحيحة لعدد صحيح على عدد صحيح آخر، بينما يمثل الـ mod باقي القسمة الصحيحة.

مثال:

13 Div 3 = 4

و

13 Mod 3 = 1

وبهذا نستطيع أن نكتب:

13 = (13 Div 3) * 3 + (13 Mod 3)

II.2.3.النوع الحقيقي:

يكتب هذا النوع داخل الخوارزمية réel وهو يشمل الأعداد الصحيحة والأعداد الحقيقة معا، فكل عدد صحيح هو عدد حقيقي، والعكس غير صحيح.

والجدير بالتنويه أن النوع الحقيقي يحتاج إلى 32 أوكتي (octet) لتمثيل الأعداد أي 256 بيت كأقصى حد، وتتراوح قيمة الأعداد من $3.402 * 10^{-38}$ إلى $3.402 * 10^{38}$ ومثلها في الأعداد السالبة. لذا فلا يجب استعمال النوع الحقيقي إلا

* يجب التفريق بين bit و byte لأن 1 byte = 1 octet = 8 bits

إذا دعت الحاجة لذلك، فهو نوع مستهلك للذاكرة وإذا تمت المبالغة في استعماله دون الحاجة فهو يؤدي إلى تشبع الذاكرة وتعطل البرنامج.

ويكتب النوع الحقيقي قي لغات البرمجة على أشكال متعددة، real أو float أو double.

II.3.3.النوع المنطقي:

يكتب هذا النوع داخل الخوارزمية boolean-booléen نسبة إلى عالم الرياضيات البريطاني (George Boole) الذي قام سنة 1854 بوضع أسس الحساب المنطقي الذي يعتمد على قيمتين فقط، صحيح – خطا، vrai – faux، true-false، 1-0.

ويستعمل النوع المنطقي مجموعة من العوامل المنطقية نذكر منها: الوصل (و, ET, AND)، الفصل (أو, OU, OR) والنفي (NON, NOT).

ولتوضيح كيفية عمل هذه العوامل ارتأينا إدراج الجدول التالي والذي يسمى بجدول الحقيقة (table de vérité)، بحيث نأخذ متغيرين منطقيين A وB ونبين كل الحالات الناتجة عن استعمال العوامل النفي، الوصل، الفصل، NON, ET, OU، مع التنبيه أنه في حالة وجود عدة عوامل منطقية في نفس العملية فالأولوية ترجع إلى النفي ثم الوصل وأخيرا الفصل.

الجدول (1): جدول الحقيقة

A	B	NON A	A ET B	A OU B
Vrai	Vrai	Faux	Vrai	Vrai
Vrai	Faux	Faux	Faux	Vrai
Faux	Vrai	Vrai	Faux	Vrai
Faux	Faux	Vrai	Faux	Faux

يرتبط النوع المنطقي عموما بالمقارنات والشروط والمراقبة، وكمثال عن ذلك تعتبر المقارنة التالية: $(x > y)$ صحيحة إذا كانت قيمة x أكبر من قيمة y، فإذا كان $x = 5$ و $y = 13$ فالمقارنة السابقة تصبح خاطئة، أما المقارنة التالية: $(x > y)$ ET $(y < z)$ لا تكون صحيحة إلا إذا كانت قيمة x أكبر من قيمة y وقيمة y أصغر من قيمة z.

يجب التنويه في هذا المقام إلى كيفية كتابة عوامل المقارنة في المعلوماتية وداخل الخوارزميات، فرمز أصغر أو يساوي والذي يكتب في الرياضيات $\leq$ يكتب في الخوارزميات =< ورمز أكبر أو يساوي يكتب في الخوارزميات =>، أما رمز لا يساوي $\neq$ فيكتب <>. كما يجب أن ننوه إلى أن نفي المقارنة السابقة أي $\overline{(x > y) \text{ ET } (y < z)}$ هو $(x <= y) \text{ OU } (y >= z)$ وهو بالتالي $(x > y) \text{ OU } (y < z)$ لأن نفي أكبر تماما هو أصغر أو يساوي ونفي أصغر تماما هو أكبر أو يساوي، والرمز $\overline{\quad}$ يمثل النفي.

يعتبر النوع المنطقي من الأنواع المحببة لدى كثير من المبرمجين، ويرجع هذا إلى الحجم الضئيل الذي يحتاجه هذا النوع في الذاكرة، بحيث يكفي بيت واحد لتمثيله، عندما يأخذ البيت 1 يكون المتغير صحيح vrai وعندما يأخذ البيت 0 يكون المتغير خاطئا faux، وكمثال عن استعمال المتغير المنطقي يمكن لخوارزمية تعطي نتائج الطلبة نهاية السنة الجامعية،

أن تستخدم متغير من النوع المنطقي تأخذ القيمة الصحيحة vrai عندما يكون الطالب ناجحا وتأخذ القيمة الخاطئة faux عندما يكون الطالب راسبا.

II.4.3.النوع الحرفي:

يستعمل هذا النوع لكتابة الحروف أو الكلمات ويكتب داخل الخوارزمية caractère أو car أو chaine de caractère (string) وهو نوع كثير الاستعمال في الخوارزميات بحيث نجد الكثير من المتغيرات من هذا النوع كاسم الطالب أو لقبه، اسم المقياس، اسم الوحدة، اسم التخصص،....إلخ. يكتب محتوى هذا النوع من المتغيرات بين مزدوجتين للتفريق بينه وبين أنواع المتغيرات الأخرى، فمثلا نكتب "a"، "b"، "toto"، "21".

يجب أن نفرق بين "21" و21 فالأول هو مجموعة الحروف المكونة من "1" و"2" والثاني هو العدد 21 (واحد وعشرون)، تجدر الإشارة إلى أن عملية الإلصاق بين الحروف يرمز لها بـ "&" أو "+" بحيث:

"21" = "1" + "2" ، وكذلك "21" = "1" & "2"

II.4.كيفية التصريح بالمتغيرات والثوابت داخل الخوارزميات (Déclaration des variables et des constantes):

يتم التصريح بكل الثوابت والمتغيرات التي سوف تستعمل داخل الخوارزمية، فنبدأ دائما بالتصريح بالثوابت أولا ثم نصرح بالمتغيرات، فلا يجوز استعمال ثابت أو متغير داخل الخوارزمية دون التصريح به في جزء الإعلانات، كذلك لا يجوز التصريح بمتغير أو ثابت ثم لا يتم استعماله داخل الخوارزمية، مع أن أغلب لغات البرمجة تصدر خطأ (error) في الحالة الأولى وتحذير (warning) في الحالة الثانية.

يتم اختيار أنواع المتغيرات حسب المسألة التي نود حلها، ونختار أسماء المتغيرات بعناية بحيث ترمز لمعطيات وحلول المسألة، ولا تكون جوفاء بدون علاقة مع موضوع المسألة، فمثلا المتغير rayon, surface والثابت PI وكذلك العملية:

surface = PI * rayon * rayon

أفضل من المتغير y ,x والثابت z والعملية السابقة أفضل من هذه العملية:

$y = z * x * x$

ملاحظة:

اسم المتغير أو الثابت يبدأ لزوما بحرف أبجدي، فلا يجوز تسمية متغيرة بـ 3Rayon أو &Rayon ولكن يجوز تسمية متغيرة Rayon3 أو Rayon&3.

نعلن عن المتغيرات والثوابت بالشكل العام التالي:

الثابت (const)

const nom_constante = valeur_constante

المتغير (var)

var nom_variable : type_variable

مثال:

```
const   PI = 3.14
var  A : entier
     surface, rayon : réel
```

أو

```
const   PI = 3.14
var  A : entier
     surface : réel
     rayon : réel
```

II .5. كيفية تحليل الصيغ الرياضية في الخوارزمية:

تختلف كتابة الصيغ الرياضية داخل الخوارزميات وداخل جميع لغات البرمجة عن كتابتها الرياضية الكلاسيكية، وهذا لاعتبارات عدة، منها ما هو متعلق بسهولة الكتابة ومنها ما هو متعلق بعدم الالتباس والوضوح، وفيما يلي توضيح لكتابة العوامل والدوال الواسعة الاستعمال داخل المعلوماتية عموما وداخل الخوارزميات على وجه الخصوص:

- الضرب يرمز له بـ *، فلا نكتب $a \times b$ ولا نكتب $a \cdot b$ ولا نكتب $a\,b$ بل نكتب $a * b$.
- القسمة يرمز لها بـ /، فلا نكتب $\frac{a}{b}$ ولا نكتب $a \div b$ ولكن نكتب a/b.
- الأس نرمز له بـ ^ أو ** فلا نكتب a^b ولكن نكتب $a \wedge b$ أو $a ** b$.
- الجذر التربيعي يرمز له بـ sqrt (square root) فنكتب sqrt(x) ولا نكتب $\sqrt{x}$.
- القيمة المطلقة يرمز لها بـ abs ونكتب abs(x) ولا نكتب $|x|$.

II .6. استعمال الأولويات لحساب الصيغ الرياضية:

عند تنفيذ أي عملية حسابية سواء في الخوارزميات أو في الرياضيات، يوجد أولويات للعوامل المستعملة تضمن عدم الالتباس وأحادية النتيجة، فعلى سبيل المثال العملية التالية يمكن أن تعطي نتيجتين مختلفتين وخاطئتين إذا لم تحترم الأولويات:

15 + 5 * 3 + 2 = 62 (1)
15 + 5 * 3 + 2 = 100 (2)

أما النتيجة الصحيحة فهي:

15 + 5 * 3 + 2 = 32 (3)

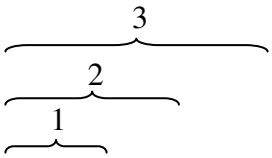
وهذا راجع لأولوية عملية الضرب على عملية الجمع، وفيما يلي ترتيب الأولويات:

1. الأس؛
2. الضرب والقسمة؛
3. الجمع والطرح.

- في حالة وجود عملية حسابية بها عوامل متساوية الأولويات يتم التنفيذ والحساب من اليسار إلى اليمين؛

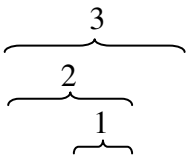
- في حالة وجود عملية تحتوي على أقواس، ترجع الأولوية للعمليات التي تقع داخل الأقواس، مع إعطاء الأولوية المطلقة للأقواس الداخلية.

لذا وبالرجوع إلى المثال السابق، الحساب الأول تم حسب ما يلي:



((15 + 5) * 3) +(2)= 62

وهو بذلك أعطى الأولوية لعملية الجمع على عملية الضرب، ونفس الشيء حدث في العملية الثانية، والحساب الصحيح هو الذي تم في العملية الثالثة، بحيث احترم مبدأ الأولويات فقد بدأ بحساب عملية الضرب ثم وبوجود عمليتي جمع بدأ باليسار ثم جمع الناتج مع العدد الذي جاء في اليمين على النحو التالي:



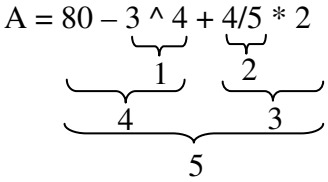
15 + 5 * 3 + 2 = 32

تطبيق حسابي:

نفذ كلا من العمليتين التاليتين باحترام مبدأ الأولويات:

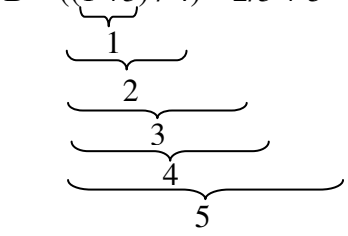
A = 80 – 3 ^ 4 + 4/5 * 2
B = ((1 +3) / 4) * 2/3 + 5

الحل:



إذا الحساب يتم كالآتي:

A = 80 – 81 + 4/5 * 2 (1)
A = 80 – 81 + 0.8 * 2 (2)
A = 80 – 81 + 1.6 (3)
A = -1 + 1.6 (4)
A = 0.6 (5)



والحساب يتم كالآتي:

B = (4 / 4) * 2/3 + 5 (1)

B = 1 * 2/3 + 5 (2)
B = 2/3 + 5 (3)
B = 0.66 + 5 (4)
B = 5.66 (5)

III.تمارين مقترحة:

التمرين الأول:

1) أكتب الصيغ الرياضية التالية كتابة معلوماتية:

$$L = \frac{3y^2x^3 + 5y^3zx}{5x - z^2}\ ;\ P = z^2x^2 - 5x^3zy\ ;$$

$$K = \frac{2bd^3 + bd}{db + ba}\ ;\ M = ((d+b)^a)^c\ ;$$

$$N = \frac{\frac{2}{a-b}}{\frac{(c+d)(b+a)}{(d+b)^2-(a-c)^3}}\ ;\ S = \frac{\frac{b+\sqrt{c^2-4ad}}{a}}{|c+a|}$$

$$R = \frac{(1+d)c + (b+a)d}{(a+c)^2} - \sqrt{\frac{(a+b)^2}{(d-c)^2 + cd + ab}} + (b+a)c$$

2) قم بتنفيذ الصيغ R , S , M بالقيم التالية: a=2, b=3, c=-3, d=-5 مع تبيان كل مراحل الحساب.

التمرين الثاني:

1) أكتب الصيغة الرياضية التالية كتابة معلوماتية

$$y = z^{b-3} + 2\pi\sqrt{\frac{|c+3a|}{b-a^2}}$$

2) قم بتنفيذ الصيغة y بالقيم التالية: $a = 1, b = 5, c = -19, z = -5$ مع تبيان كل مراحل الحساب.

IV. حلول التمارين

حل التمرين الأول:

1)

- $L = \frac{3y^2x^3 + 5y^3zx}{5x - z^2}$

$L = (3 * y \wedge 2 * x \wedge 3 + 5 * y \wedge 3 * z * x)/(5 * x - z \wedge 2)$

- $P = z^2x^2 - 5x^3zy$

$P = z \wedge 2 * x \wedge 2 - 5 * x \wedge 3 * z * y$

- $K = \frac{2bd^3+bd}{db+ba}$

$K = (2 * b * d \wedge 3 + b * d)/(d * b + b * a)$

- $M = ((d+b)^a)^c$

$M = ((d + b) \wedge a) \wedge c$

- $N = \frac{\frac{2}{a-b}}{\frac{(c+d)(b+a)}{(d+b)^2-(a-c)^3}}$

$N = (2/(a - b))/(((c + d) * (b + a)\ )/(((d + b) \wedge 2)-((a - c) \wedge 3)))$

- $S = \frac{\frac{b+\sqrt{c^2-4ad}}{a}}{|c+a|}$

$S = (b + sqrt(c \wedge 2 -4 * a * d))/a/abs(c + a)$

- $R = \frac{(1+d)c+(b+a)d}{(a+c)^2} - \sqrt{\frac{(a+b)^2}{(d-c)^2+cd+ab}} + (b+a)c$

$R = ((1 + d) * c + (b + a) * d)/(a + c) \wedge 2 - sqrt((a + b) \wedge 2/((d - c) \wedge 2 + c * d + a * b)) + (b + a) * c$

a=2, b=3, c=-3, d=-5 (2

- $M = ((-5 + 3) \wedge 2) \wedge -3$

$M = ((-2) \wedge 2) \wedge -3$ (1)

$M = (4) \wedge -3$ (2)

$M = 0.015625$ (3)

- $S = (b + sqrt(c \wedge 2 -4 * a * d))/a/abs(c + a)$

$S = (3 + sqrt(-3 \wedge 2 - 4 * 2 * -5))/2/abs(-3 + 2)$ (1)

$S = (3 + sqrt(9 - 4 * 2 * -5))/2/abs(-3 + 2)$ (2)

$S = (3 + sqrt(9 - 8 * -5))/2/abs(-3 + 2)$ (3)

$S = (3 + sqrt(9 - - 40))/2/abs(-3 + 2)$ (4)

$S = (3 + sqrt(49))/2/abs(-3 + 2)$ (5)

$S = (3 + 7)/2/abs(-3 + 2)$ (6)

$S = (10)/2/abs(-3 + 2)$ (7)

$S = 5/abs(-3 + 2)$ (8)

$S = 5/abs(-1)$ (9)

$S = 5/1$ (10)

$S = 5$ (11)

- $R = ((1 + -5) * -3 + (3 + 2) * -5)/(2 + -3) \wedge 2 - sqrt((2 + 3) \wedge 2/((-5 - -3) \wedge 2 + -3 * -5 + 2 * 3)) + (3 + 2) * -3$

R = $(-4 * -3 + (3 + 2) * -5)/(2 + -3) \wedge 2 - sqrt((2 + 3) \wedge 2/((-5 - -3) \wedge 2 + -3 * -5 + 2 * 3)) + (3 + 2) * -3$ (1)

R = $(-4 * -3 + 5 * -5)/(2 + -3) \wedge 2 - sqrt((2 + 3) \wedge 2/((-5 - -3) \wedge 2 + -3 * -5 + 2 * 3)) + (3 + 2) * -3$ (2)

R = (12+ 5*-5)/(2+ -3)^2 – *sqrt* ((2+3) ^2/((-5 - -3) ^ 2 + -3 * -5 + 2 * 3)) + (3 + 2) * -3 (3)

R = (12+ -25)/(2 + -3) ^2 – *sqrt* ((2 + 3) ^2/((-5 - -3) ^ 2 + -3 * -5 + 2 * 3)) + (3 + 2) * -3 (4)

R = -13/(2 + -3) ^2 – *sqrt* ((2 + 3) ^2/((-5 - -3) ^ 2 + -3 * -5 + 2 * 3)) + (3 + 2) * -3 (5)

R = -13/(-1) ^2 – *sqrt* ((2 + 3) ^2/((-5 - -3) ^ 2 + -3 * -5 + 2 * 3)) + (3 + 2) * -3 (6)

R = -13/1– *sqrt* ((2 + 3) ^2/((-5 - -3) ^ 2 + -3 * -5 + 2 * 3)) + (3 + 2) * -3 (7)

R = -13 – *sqrt* ((2 + 3) ^2/((-5 - -3) ^ 2 + -3 * -5 + 2 * 3)) + (3 + 2) * -3 (8)

R = -13 – *sqrt* (5 ^2/((-5 - -3) ^ 2 + -3 * -5 + 2 * 3)) + (3 + 2) * -3 (9)

R = -13 – *sqrt* (25/((-5 - -3) ^ 2 + -3 * -5 + 2 * 3)) + (3 + 2) * -3 (10)

R = -13 – *sqrt* (25/(-2^ 2 + -3 * -5 + 2 * 3)) + (3 + 2) * -3 (11)

R = -13 – *sqrt* (25/(4 + -3 * -5 + 2 * 3)) + (3 + 2) * -3 (12)

R = -13 – *sqrt* (25/(4 + 15 + 2 * 3)) + (3 + 2) * -3 (13)

R = -13 – *sqrt* (25/4 + 15 + 6) + (3 + 2) * -3 (14)

R = -13 – *sqrt* (25/25) + (3 + 2) * -3 (15)

R = -13 – *sqrt* (1) + (3 + 2) * -3 (16)

R = -13 – 1 + (3 + 2) * -3 (17)

R = -13 – 1 + 5 * -3 (18)

R = -13 – 1 + -15 (19)

R = -14 + -15 (20)

R = -29 (21)

حل التمرين الثاني:

1)

- $y = z^{b-3} + 2\pi\sqrt{\frac{|c+3a|}{b-a^2}}$

$y = z \text{ ^ } (b - 3) + 2 * PI * sqrt((abs(c + 3 * a))/(b - a \text{ ^2}))$

2) $a = 1, b = 5, c = -19, z = -5, PI = 3.14$

$y = -5 \text{ ^ } (5 - 3) + 2 * 3.14 * sqrt((abs(-19 + 3 * 1))/(5 - 1 \text{ ^2}))$ (1)

$y = -5 \text{ ^ } 2 + 2 * 3.14 * sqrt((abs(-19 + 3 * 1))/(5 - 1 \text{ ^2}))$ (2)

$y = 25 + 2 * 3.14 * sqrt((abs(-19 + 3))/(5 - 1 \text{ ^2}))$ (3)

$y = 25 + 2 * 3.14 * sqrt((abs(-16))/(5 - 1 \text{ ^2}))$ (4)

$y = 25 + 2 * 3.14 * sqrt(16/(5 - 1 \text{ ^2}))$ (5)

$y = 25 + 2 * 3.14 * sqrt(16/(5 - 1))$ (6)

$y = 25 + 2 * 3.14 * sqrt(16/4)$ (7)

$y = 25 + 2 * 3.14 * sqrt(4)$ (8)

$y = 25 + 2 * 3.14 * 2$ (9)

$y = 25 + 6.28 * 2$ (10)

$y = 25 + 12.56$ (11)

$y = 37.56$ (12)

V. التعليمات الأساسية البسيطة في الخوارزميات–تعليمة التخصيص، القراءة والكتابة

تعتبر التعليمات (instructions) أهم جزء في الخوارزمية، بحيث يتم من خلالها إنجاز المهام المطلوب إنجازها من طرف الخوارزمية، لذا سنتناول في هذا الفصل أهم هذه التعليمات انطلاقا من تعليمة التخصيص (affectation) وانتهاء بالتعليمات التكرارية، مع تناول التعليمات الشرطية وتعليمات الإدخال والإخراج.

1.V تعريف التعليمة:

تعرف التعليمة على أنها الأمر الذي يعطى للخوارزمية من أجل تنفيذه، بحيث تتغير غالبا قيم المعطيات والبيانات بعد تنفيذ هذه التعليمة. وتنقسم التعليمات إلى نوعين أساسين هما:

- تعليمات داخلية؛ تغير من محتوى المتغيرات والبيانات من داخل الخوارزمية، بحيث لا يمكن لمستعمل الخوارزمية التدخل في تغيير تلك البيانات والمتغيرات؛
- تعليمات خارجية؛ تغير من محتوى المتغيرات والبيانات من خارج الخوارزمية عن طريق مستعمل الخوارزمية دون أن يكون له علم بكاتبة الخوارزمية وبطريقة عملها.

وسنستعرض فيما يلي مجموعة من التعليمات الأساسية المستعملة في الخوارزميات.

2.V تعليمة التخصيص أو إعطاء القيمة (Affectation):

تعتبر هذه التعليمة من التعليمات الداخلية في الخوارزمية، وتسمح بتغير محتوى متغير بقيمة ثابتة بسيطة أو قيمة ثابتة مركبة أو جملة رياضية مكونة من مزيج من المتغيرات والثوابت، ويرمز لها اصطلاحا في لغة الخوارزميات بسهم متجه نحو اليسار (←) وهي تختلف كل الاختلاف عن رمز المساواة (=)، وكمثال عن تعليمة تخصيص بسيطة، الكتابة التالية:

A ← 5

وتترجم هذه الكتابة بـ: المتغير A يأخذ القيمة 5 (A reçoit 5) .

وتكتب تعليمة التخصيص في البرمجة حسب لغة البرمجة المستعملة، ففي لغة الباسكال تكتب تعليمة التخصيص كالآتي:

A : = 5

بينما تكتب المقارنة كالآتي:

A = 5

بينما تكتب تعليمة التخصيص في لغة الـ C أو C++ كالآتي:

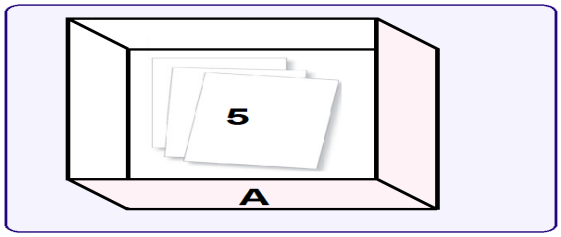
A = 5

بينما تكتب المقارنة كالآتي:

A = = 5

وتجدر بنا الإشارة إلى أن القيمة التي كانت محتواة داخل المتغير A قبل تنفيذ عملية التخصيص، ستفقد إلى الأبد ولن يتم استرجاعها، بحيث بعد عملية التخصيص تمحى القيمة السابقة وتستبدل بالقيمة الجديدة، ولتوضيح هذه العملية أدرجنا الشكل التالي الذي يبين المتغير A على شكل علبة مفتوحة ملصق عليها من وجهها الخارجي اسمها (A) أما داخلها مجموعة من الأوراق بحيث لا يرى إلا آخر ورقة وهي الورقة الأخيرة التي وضعت بداخلها، وبهذا نقول أن العلبة A بها الورقة 5 أما الورقة التي كانت من قبل لا يمكن رؤيتها.

الشكل (5): شكل توضيحي لعملية التخصيص



وكما أسلفنا، يمكن لعملية التخصيص أن تكون بقيمة ثابتة كالمثال السابق بحيث يأخذ المتغير قيمة الثابت مباشرة، أو قيمة مركبة كـ: A ← 5 + 2 * 4، وفي هذه الحالة تحسب نتيجة العمليات الحسابية ثم توضع داخل المتغير.

وقد تكون عملية التخصيص بقيمة متغيرة كـ: A ← B، وفي هذه الحالة يأخذ محتوى المتغير الذي على يمين السهم، ويوضع داخل المتغير الذي على يساره، وبهذا يصبح محتوى المتغيرين متساويا، أما آخر حالة لعملية التخصيص فتكون بجملة رياضية مكونة من مجموعة من المتغيرات كـ: A ← B * C + D ^ 2، في هذه الحالة يتم حساب الجملة الرياضية التي على يمين السهم، باستعمال مبدأ الأولويات كما رأينا آنفا، ثم يوضع ناتج الحساب داخل المتغير الذي على يسار السهم.

يجب التنويه إلى أن الجملة الرياضية التي على يمين عملية التخصيص يمكن أن تحتوي نفس المتغيرة كـ: A ← A + 1، أو A ← 2 * A + B، وفي كلتا الحالتين يحسب ناتج العملية الحسابية باستعمال قيمة المتغير A القديمة أي التي كانت قبل عملية التخصيص ثم يوضع هذا الناتج في المنتغير A فتتغير قيمته، وكمثال تطبيقي نأخذ المتغير A والمتغير B داخل الخوارزمية التالية ثم نتتبع قيمهما بحيث بعد كل عملية تخصيص نلاحظ تغيرهما:

```
Algorithme variables
var   A, B : entier
début
A ← 10

B ← 8

A ← B * 2

A ← A + B

fin
```

في هذه المرحلة وبعد تنفيذ عملية التخصيص يأخذ المتغير A القيمة 10

في هذه المرحلة وبعد تنفيذ عملية التخصيص يأخذ المتغير B القيمة 8

في هذه المرحلة وبعد تنفيذ عملية التخصيص يأخذ المتغير A القيمة 16 = 2 * 8

في هذه المرحلة وبعد تنفيذ عملية التخصيص يأخذ المتغير A القيمة القديمة لـ A أي 16 ثم نضيف لها قيمة B أي 8 فيصبح المجموع 24 = 8 + 16 وبهذا تصبح القيمة الجديدة والنهائية لـ A 24

تسمى عملية متابعة تغير قيم المتغيرات داخل الخوارزمية بعملية محاكاة التنفيذ أو عملية متابعة التنفيذ أو عملية التظاهر باليد (déroulement d'exécution)، كما تسمى عملية إضهار النتائج، بعملية عرض النتائج على الشاشة أو عملية الإخراج (Affichage Ecran)، وهو ما سنتطرق إليه فيما يلي:

V.1.2 عملية محاكاة التنفيذ أو متابعة التنفيذ (Déroulement de l'exécution):

تتم هذه العملية من خلال جدول يحتوي على مجموعة من الأعمدة ومجموعة أخرى من الأسطر، بحيث يمثل كل عمود متغير معين داخل الخوارزمية، بينما تمثل الأسطر تغيرات قيم هذا المتغير، مع تفادي تمثيل المتغيرات التي لا تغير من قيمها داخل الخوارزمية، وكمثال تطبيقي لعملية محاكاة التنفيذ نأخذ الخوارزمية السابقة ونحاول محاكاة تنفيذها:

```
Algorithme variables
1) var   A, B : entier
début
2) A ← 10
3) B ← 8
4) A ← B * 2
5) A ← A + B
fin
```

	A	B
1	?	?
2	10	?
3	10	8
4	16	8
5	24	8

لتسهيل عملية المحاكاة من الأفضل ترقيم التعليمات بحيث يعطى لكل تعليمة رقما تسلسليا، ويتم وضع قيم المغيرات مقابل رقم التعليمة، كما يتم إعطاء جزء الإعلانات الرقم (1)، بحيث وبمجرد التصريح بالمتغيرات يأخذ المتغير قيمة عشوائية يرمز لها بـ ? كما هو موضح في الجدول السابق. يمكن أن يبدأ الترقيم بعد كلمة début مع اعتبار قيم المتغيرات المصرح بها قيم عشوائية ?.

تأخذ المتغيرات قيمتها النهائية بعد آخر تعليمة في الخوارزمية (في حالة الخوارزمية السابقة التعليمة رقم (5) هي آخر تعليمة) وتعتبر القيم 24 و8 القيم النهائية للمتغيرتين A و B على التوالي.

V.2.2 عملية العرض على الشاشة أو إظهار النتائج (Affichage Ecran):

تتعلق هذه العملية بتعليمة الإخراج أو تعليمة الكتابة وكذا تعليمة الإدخال أو الكتابة، وهو ما سيتم التطرق إليه فيما يلي.

V.3. تعليمات القراءة والكتابة أو الإدخال والإخراج (Lecture/Ecriture ou Entrée/Sortie):

تعتبر هاتان التعليمتان من التعليمات ذات العلاقة بأجهزة الإدخال والإخراج أو ما يرمز له بـ E/S أو I/O (Input/Output)، ولأهمية هاتين التعليمتين في المعلوماتية عموما وفي الخوارزميات على وجه الخصوص، يعتبر أول ما ينفذه جهاز الحاسوب عند محاولة تشغيله هو نظام الإدخال والإخراج الأساسي أو ما يعرف اختصارا (BIOS) (Basic Input Output System) وهو النظام المسؤول عن مجموعة من الوظائف مدرجة داخل الذاكرة الميتة للحاسوب

أو ما يعرف بـ ROM (Read Only Memory) ويسمح هذا النظام من مراقبة كل أجهزة الإدخال والإخراج الموصولة مع الجهاز، بحيث في حالة تعطل أحد هذه الأجهزة يمكن أن يحدث خلل في نظام التشغيل.

V.1.3. تعليمة القراءة أو الإدخال (Instruction de lecture / d'entrée):

تعتبر هذه التعليمة من التعليمات الخارجية، بحيث تتيح للمستخدم من إدخال قيم المتغيرات أو البيانات من خارج الخوارزمية، دون معرفة محتوى الخوارزمية أو كيفية كتابتها، بحيث توجه القيم المدخلة من طرف المستخدم مباشرة إلى المتغيرات المعنية، وتكتب تعليمة القراءة في الخوارزمية بالشكل التالي:

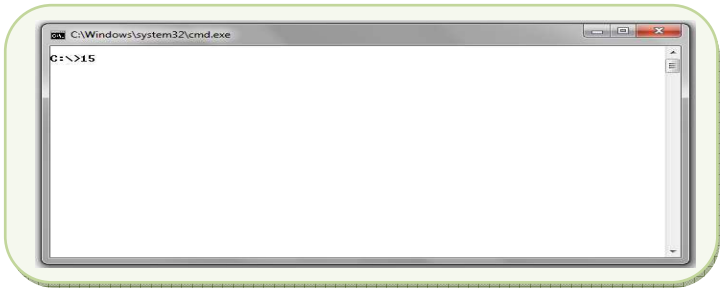
```
lire (Nom_variable1)
lire (Nom_variable2)
```

أو

```
lire( Nom_variable1 , Nom_variable2 )
```

عند تنفيذ عملية القراءة فإن جهاز الحاسوب يعرض على الشاشة نافذة يتم من خلالها إدخال القيمة المنتظرة، ويتم تأكيد هذه العملية من خلال الظغط على زر الدخول (entrée ou enter) عبر لوحة المفاتيح، كما هو موضح من خلال الشكل التالي الذي يوضح عملية قراءة للقيمة 15 من خلال تعليمة lire(a):

الشكل (6): شكل واجهة عملية إدخال المعطيات من خلال تعليمة القراءة عبر عارض cmd لنظام windows



بمجرد الظغط على زر الإدخال تأخذ المتغيرة a القيمة 15.

مثال تطبيقي:

```
Algorithme lecture
var   x, y : réel
début
lire (x)
y ← x * x
fin
```

الخوارزمية السابقة تطلب من المستخدم إدخال قيمة معينة من خلال التعليمة lire(x) ثم تقوم بحساب مربع هذه القيمة وتضعه في المتغير y.

V.2.3. تعليمة الكتابة أو الإخراج (Instruction d'écriture / de sortie):

تسمح هذه التعليمة، كما يدل عليها اسمها، بكتابة النتائج وإخراجها للمستخدم، وتكون عملية الإخراج أو الكتابة

غالبا على شاشة الحاسوب، وتمكن هذه التعليمة من عرض ونشر النتائج حتى يتم الإطلاع عليها، وتكتب هذه التعليمة في الخوارزمية على عدة أشكال حسب الحاجة، فقد تكون عملية الكتابة بسيطة بحيث تعرض على الشاشة محتوى المتغيرة كما في المثال التالي:

écrire (Nom_variable1)
écrire (Nom_variable2)

في هذه الحالة يعرض محتوى المتغيرتين Nom_variable1 وNom_variable2 على الشاشة، الواحدة بجانب الأخرى، كما يمكن الجمع بين التعليمتين السابقتين في تعليمة واحدة على الشكل التالي:

écrire (Nom_variable1, Nom_variable2)

يمكن استعمال تعليمة الكتابة لعرض تعاليق على الشاشة وكتابتها بالشكل الذي يرتضيه كاتب الخوارزمية، فيتم عرض التعليق على الشاشة بكل دقة وأمانة، بحيث أي حرف أو شكل داخل التعليق سوف يعرض على الشاشة بحذافيره، وكمثال عن ذلك التعليق التالي: écrire ("Cet algorithme calcule le carré d'un nombre entier !")

بعد تنفيذ هذه التعليمة سيظهر على الشاشة هذا التعليق كما هو مكتوب بين المزدوجتين.

ويمكن استعمال تعليمة كتابة التعاليق لإرشاد المستخدم حول ما هو مطلوب منه، وما هو دور الخوارزمية، كما هو موضح في المثال التالي:

```
Algorithme carré_nombre
var   a, b : entier
début
    écrire ("Cet algorithme calcul le carré d'un nombre entier !" )
    écrie ("Veuillez SVP entrer un nombre entier")
    lire (a)
    b ← a * a
    écrire ("Le carré du nombre saisi est :")
    écrire (b)
fin
```

الخوارزمية السابقة تحسب مربع عدد صحيح وتعرضه على الشاشة، في بادئ الأمر تعرض الخوارزمية تعليق على الشاشة تبين فيه الدور المنوط بها، ثم تعرض تعليقا آخر تطلب من خلاله من المستخدم إدخال عدد صحيح ليتم حساب مربعه، ثم بعد عملية الحساب تعرض تعليقا آخر تبين فيه أن مربع العدد الذي تم إدخاله هو: وبعد هذا التعليق تقوم بعرض النتيجة من خلال التعليمة écrire(b) والتي تعرض محتوى المتغيرة b على الشاشة.

ويمكن المزاوجة بين تعليمة كتابة النتائج وتعليمة كتابة التعاليق في تعليمة واحدة بشرط احترام القواعد التالية:

- يجب كتابة التعاليق بين مزدوجتين؛
- في حالة كتابة تعليق مع كتابة نتيجة متغيرة معينة، يجب وضع فاصلة بينهما كما في المثال التالي:

écrire ("Le carré du nombre saisi est : ", b)

أو

écrire ("Le carré de ", a," est : ", b)

في حالة ما إذا كان العدد المدخل يساوي 4 على سبيل المثال فإن الشاشة تُظهر في الحالى الأولى:

Le carré du nombre saisi est : 16

أما في الحالة الثانية فتُظهر

Le carré de 4 est :16

VI تمارين مقترحة:

التمرين الأول:

قم بمحاكاة التنفيذ (Déroulement) للخوارزميات التالية :

1)

```
Algorithme CALCUL
  var   a, b, c: entier
        z : réel
début
        a←3
        b←2
        c←5
        a←c div b
        b←c mod  a
        a← (a+4)*3+b*a
        z←c/a+(c-b)*2*a
        z←z * b - a
        Ecrire(a ," ", b, " ",c)
        Ecrire(" ",z)
fin
```

2)

```
Algorithme CALCUL
  var      x, y, z: réel
début
        x ← -3,14
        y ← 5,85
        z ←x^3*y+2*x*y
        y←abs(z/3-y^2)
        x ← sqrt((y – z * x)^2 + y^2)
        z←x-y/z
        Ecrire(x," ", y," ", z)
 fin
```

التمرين الثاني:

قم بمحاكاة التنفيذ (Déroulement) للخوارزميات التالية :

1)

```
Algorithme Mots
    var   a, b, c, d, e,  Mat, Nom_etud, res : caractère
début
        a←"3104545"
        b←"20"
        c ← "le matricule"
        d ← " de "
        e ← " est :  "
        Mat← b + a
        Nom_Etudiant ← "Mohamed"
        res ← c + d + Nom_Etudiant + e + Mat
        Ecrire(res)
fin
```

2)

```
Algorithme Mots
 var   a, b, c, d,  Nom_etud, res : caractère
début
        a←"17"
        d ← ",75"
        a ← a + d
        b ← "la moyenne"
        c ← " de "
        e ← " est :  "
        Nom_Etudiant ← "Mohamed"
        res ← b + c + Nom_Etudiant + e + a
        Ecrire(res)
fin
```

التمرين الثالث:

أكتب خوارزمية تحسب معدل (Moy) ثلاثة نقاط: Note1, Note2, Note3 موزونة بمعاملاتها: coef1, coef2, coef3 ثم تعرضه على الشاشة.

ملاحظة: Moy = (Note1*coef1 +Note2*coef2+Note3*coef3)/ (coef1+ coef2+ coef3)

التمرين الرابع:

أكتب خوارزمية تطلب من المستخدم إدخال قيمة ثمن منتوج من دون الضريبة PHT ثم تطلب منه إدخال نسبة الرسم على القيمة المضافة (taux TVA) TTVA ثم تحسب قيمة الرسم على القيمة المضافة PTVA وتعرضه على الشاشة وثمن المنتوج بكل الرسوم PTTC وتعرضه على الشاشة.

ملاحظة:

.PTVA = PHT * TTVA/100
.PTTC = PHT *(1 + TTVA/100)

التمرين الخامس:

أكتب خوارزمية تطلب من المستخدم إدخال ثمن منتوج من دون الضريبة PHT ثم تطلب نسبة الرسم على القيمة المضافة TTVA ثم تطلب كمية المنتوج (Qte) ثم تحسب قيمة الرسم على القيمة المضافة PTVAG لكل البضاعة (الكمية الكلية للمنتوج) وتعرضه على الشاشة وتحسب القيمة الكلية للبضاعة بكل الرسوم PTTCG وتعرضها على الشاشة.

ملاحظة:

PTVAG = PHT *TTVA/100 * Qte
PTTCG = PHT *(1+TTVA/100) * Qte

VII.حلول التمارين

حل التمرين الأول:

1)
Algorithme CALCUL
1) var a, b, c: entier
z : réel
début
2) a←3
3) b←2
4) c←5
5) a←c div b
6) b←c mod a
7) a← (a+4)*3+b*a
8) z←c/a+(c-b)*2*a
9) z←z * b - a
Ecrire(a ," ",b," ", c)
Ecrire(" ",z)
fin

المحاكاة

N°	a	b	c	z
1	?	?	?	?
2	3	?	?	?
3	3	2	?	?
4	3	2	5	?
5	2	2	5	?
6	2	1	5	?
7	20	1	5	?
8	20	1	5	160.25
9	20	1	5	140.25

الشاشة

20 1 5 140.25

مناقشة الحل:

لكي نتأكد من صحة المحاكاة وعدم وجود أخطاء حسابية وتسلسلية، يجب مراعاة ما يلي:

عند تنفيذ تعليمة معينة، على سبيل المثال التعليمة رقم (4) نلاحظ أن المتغيرة المعنية بعملية التخصيص هي c، بحيث تقع على يسار السهم، فإذًا بعد تنفيذ هذه التعليمة المتغيرة c فقط تغير من قيمتها بينما تبقى كل المتغيرات الأخرى على حالتها الأولية؛

عند تنفيذ تعليمة بحيث نجد نفس المتغيرة على يسار السهم وعلى يمينه، نستعمل القيمة السابقة للمتغيرة لحساب الناتج الذي نضعه في المتغيرة، وهو ما حدث في التعليمة رقم (7) بالنسبة للمتغيرة a والتعليمة رقم (9) بالنسبة للمتغيرة z.

```
2)
Algorithme CALCUL
  1) var      x, y, z: réel
début
   2)  x ← -3,14
   3)  y ← 5,85
   4)  z ←x^3*y+2*x*y
   5)  y←abs(z/3-y^2)
   6)   x ← sqrt((y – z * x)^2 + y^2)
   7)   z←x-y/z
    Ecrire(x, " ",y," ", z)
fin
```

المحاكاة

N°	x	y	z
1	?	?	?
2	-3.14	?	?
3	-3.14	5.85	?
4	-3.14	5.85	-217.85
5	-3.14	-106.84	-217.85
6	103.07	-106.84	-217.85
7	103.07	-106.84	102.58

الشاشة

103.07	-106.84	102.58

تمت عملية الحساب باستعمال مبدأ الأولويات مع تقريب النتائج إلى 100/1.

حل التمرين الثاني:

```
 1)
 Algorithme Mots
 1)  var   a, b, c, d, e,  Mat,
 Nom_etud, res : caractère
début
2)  a←‘‘3104545’’
3)  b←‘‘20’’
4)  c ← ‘‘le matricule’’
 5) d ← ‘‘ de ’’
 6) e ← ‘‘ est :  ’’
 7) Mat← b + a
 8) Nom_Etud ← ‘‘Mohamed’’
 9)  res ← c + d + Nom_Etudiant + e + Mat
   Ecrire(res)
fin
```

المحاكاة

N°	a	b	c	d	e	Mat	Nom_etud	res
1	?	?	?	?	?	?	?	?
2	3104545	?	?	?	?	?	?	?
3	3104545	20	?	?	?	?	?	?
4	3104545	20	le matricule	?	?	?	?	?
5	3104545	20	le matricule	de	?	?	?	?

6	3104545	20	le matricule	de	est :	?	?	?
7	3104545	20	le matricule	de	est :	203104545	?	?
8	3104545	20	le matricule	de	est :	203104545	Mohamed	?
9	3104545	20	le matricule	de	est :	203104545	Mohamed	le matricule de Mohamed est : 203104545

بعد تنفيذ تعليمة الكتابة écrire(res) سيظهر على الشاشة الجملة المحتواة داخل المتغيرة res أي سيظهر:

le matricule de Mohamed est : 203104545

كما أسلفنا عند تناولنا للنوع الحرفي، فالرمز "+" يعني إلصاق محتوى المتغيرين (concaténation) وهذا ما حدث عند تنفيذ التعليمة رقم (7) بحيث قمنا بإلصاق "20" مع "3104545" فكانت النتيجة "203104545".

2)

Algorithme Mots

1) var a, b, c, d, e, Nom_etud, res : caractère

début

2) a←''17''

3) d ← '',75''

4) a ← a + d

5) b ← ''la moyenne''

6) c ← '' de ''

7) e ← '' est : ''

8) Nom_Etudiant ← ''Mohamed''

9) res ← b + c + Nom_Etudiant + e + a

Ecrire(res)

fin

المحاكاة

N°	a	b	c	d	e	Nom_etud	res
1	?	?	?	?	?	?	?
2	17	?	?	?	?	?	?
3	17	?	?	,75	?	?	?
4	17,75	?	?	,75	?	?	?
5	17,75	La moyenne	?	,75	?	?	?
6	17,75	La moyenne	de	,75	?	?	?
7	17,75	La moyenne	de	,75	est	?	?
8	17,75	La moyenne	de	,75	est	Mohamed	?
9	17,75	La moyenne	de	,75	est	Mohamed	La moyenne de Mohamed est 17,75

الملاحظ في النتيجة التي ستظهر على الشاشة أنها تحتوي على معدل الطالب، إلا أن هذا المعدل، حتى وإن يظهر للمشاهد أنه عدد، فهو في واقع الأمر وداخل الخوارزمية مجموعة حروف تم التعامل معها من خلال عملية إلصاق لمجموعتين من الحروف، ",75" و "17" وهو ما يعطي "17,75".

حل التمرين الثالث:

```
Algorithme Moyenne
  var   coef1 coef2, coef3 : entier
        Note1, Note2, Note3, Moy : réel
début
   écrire ("Donnez les 3 notes ")
   lire (Note1, Note2, Note3)
   écrire ("Donnez les 3 coefficients ")
    lire (coef1, coef2, coef3)
   Moy ← Note1*coef1 + Note2*coef2 + Note3*coef3
   Moy ← Moy/(coef1 + coef2 + coef3)
   écrire ("La moyenne des notes est :", Moy)
fin
```

مناقشة الحل:

- فيما يخص جزء الإعلانات كنا نستطيع الإعلان عن كل متغير في سطر لوحده، وهذا صحيح، لكن جرت العادة، وهي من العادات الحسنة في كتابة الخوارزميات، أن يتم جمع المتغيرات التي من نفس النوع في سطر واحد كما هو في الحل المقترح؛
- حساب المعدل تم على مرحلتين، ولكن كنا نستطيع حساب المعدل في تعليمة واحدة كما يلي:

```
Moy ← (Note1*coef1 + Note2*coef2 + Note3*coef3) /(coef1 + coef2 + coef3)
```

ولكن يفضل الكثير من المبرمجين المحترفين الطريقة الأولى لاعتبارات سوف نوضحها لاحقا.

- عرض النتيجة على الشاشة جاء في تعليمة واحدة وكان يمكن تقسيمه إلى تعليمتين، أولاهما تعرض التعليق والأخرى تعرض النتيجة، ويرجع الخيار لكاتب الخوارزمية، بحيث إذا كانت الخوارزمية قصيرة يمكن تقسيم التعليمة، أما إذا كانت طويلة وكثير الأسطر من الأفضل أن تختصر قدر الإمكان، للتقليل من عدد الأسطر وكذا عدد التعليمات.

حل التمرين الرابع:

```
 Algorithme PTTC
  var   PHT, TTVA, PTVA, PTTC : réel
début
   écrire ("Donnez le prix du produit ")
   lire (PHT)
   écrire ("Donnez le taux de la TVA ")
    lire (TTVA)
   PTVA ← PHT * TTVA/100
   PTTC ← PHT *(1 + TTVA/100)
   écrire ("La valeur de la TVA est :", PTVA)
   écrire ("Le prix du produit TTC est :", PTTC)
fin
```

مناقشة الحل:

- يمكن حساب ثمن المنتوج بكامل الرسوم بطريقتين مختلفتين، الطريقة الأولى هي التي أدرجت في الحل السابق،

وتقضي بأن يحسب الثمن مباشرة وبتعليمة واحدة، أما الطريقة الثانية، فهي تتمثل في حساب ثمن المنتوج من خلال جمع قيمة المنتوج من دون رسوم PHT مع قيمة الرسم على القيمة المضافة PTVA، وهذا للاستفادة من الحساب المسبق لـ PTVA، ويفضل هذا الحل على الحل المقترح لاعتبارات حسابية تؤثر في زمن تنفيذ الخوارزمية أو البرنامج، بحيث في الطريق الأولى يحسب الـ PTTC عن طريق عملية ضرب، أما في الطريقة الثانية فالـ PTTC يحسب عن طريق عملية جمع، ومعروف في عالم البرمجة والخوارزميات أن عملية الجمع أبسط وأسهل من عملية الضرب بحيث أن درجة تعقيد عملية الضرب أكبر بكثير من درجة تعقيد عملية الجمع وتأخذ وقت أكبر منها*، من أجل هذا، في حالة الخيار بين استعمال عملية ضرب وعملية جمع، تختار عملية جمع.

- عرض النتائج على الشاشة جاء في تعليمتين مختلفتين وكان يمكن كتابتها في تعليمة واحدة كما يلي:

écrire ("La valeur de la TVA est :", PTVA, "Le prix du produit TTC est :", PTTC)

حل التمرين الخامس:

```
Algorithme PTTCG
  var   Qte : entier
        PHT, TTVA, PTVAG,  PTTCG : réel
début
   écrire ("Donnez le prix du produit ")
   lire (PHT)
  écrire ("Donnez la quantité du produit ")
   lire(Qte)
   écrire ("Donnez le taux de la TVA ")
    lire (TTVA)
   PTVAG ← PHT  *  TTVA/100 * Qte
   PTTCG ← PTVAG + PHT * Qte
   écrire ("La valeur globale de la TVA de l'ensemble de la marchandise est :", PTVAG)
   écrire ("Le montant total TTC de la marchandise est :", PTTCG)
fin
```

مناقشة الحل:

الملاحظ في هذه الخوارزمية أنها تشبه إلى حد كبير الخوارزمية التي قبلها، والاختلاف الوحيد يكمن في إضافة كمية المنتوج، لذا فكل ملاحظات هذه الخوارزمية تتطابق مع التي قبلها، مع أننا في هذه الخوارزمية وعند حساب المبلغ الكلي بكامل الرسوم استفدنا من حساب المبلغ الكلي لـ PTVAG وجمعناه مع PHT * Qte فتحصلنا على PTTCG.

* في حقيقة الأمر عملية الضرب هي مجموعة من عمليات الجمع، فمثلا 4*3 هي 4 + 4 +4.

الفصل الثاني

الفصل الثاني: التعليمات الأساسية في الخوارزميات– التعليمات الشرطية

تمهيد:

تُعَرَف التعليمات الشرطية على أنها كل تعليمة تغير مسار المعالجة أو التنفيذ حسب نتيجة صحة الشرط، بحيث يعتبر الشرط محددا لكيفية تنفيذ التعليمة، لذا سمي هذا النوع من التعليمات بالتعليمات الشرطية (Instructions conditionnelles)، وتنقسم هذه الأخيرة إلى أربعة أنواع هي:

- التعليمة الشرطية البسيطة (Instruction conditionnelle simple)؛
- التعليمة الشرطية المتناوبة (Instruction conditionnelle alternative)؛
- التعليمة الشرطية المركبة أو المتداخلة (Instruction conditionnelle imbriquée)؛
- التعليمة (selon).

سيتم تناول هذه التعليمات بشيء من التفصيل والإيضاح من خلال ما يلي:

I. التعليمة الشرطية البسيطة:

تعتبر التعليمة الشرطية البسيطة أسهل وأبسط تعليمة شرطية، لذا سميت بالبسيطة، لكن في المقابل تعتبر الأقل استعمالا في الخوارزميات، وهذا راجع إلى الحالات المحدودة للمسائل التي تستوجب استعمال هذه التعليمة.

1.I تعريف التعليمة الشرطية البسيطة:

تعتمد هذه التعليمة على شرط واحد وتقوم بعمل واحد فقط في حالة تحقق ذلك الشرط، أما حالة عدم تحقق الشرط لا تُأخذ التعليمة الشرطية بعين الاعتبار ويتم تجاهل الدخول إلى متنها.

تكتب هذه التعليمة في الخوارزمية على الشكل التالي:

```
si (condition)
| alors
|    Instruction 1
|    Instruction 2
|       :
|    Instruction n
fsi
```

2.I كيفية عمل التعليمة:

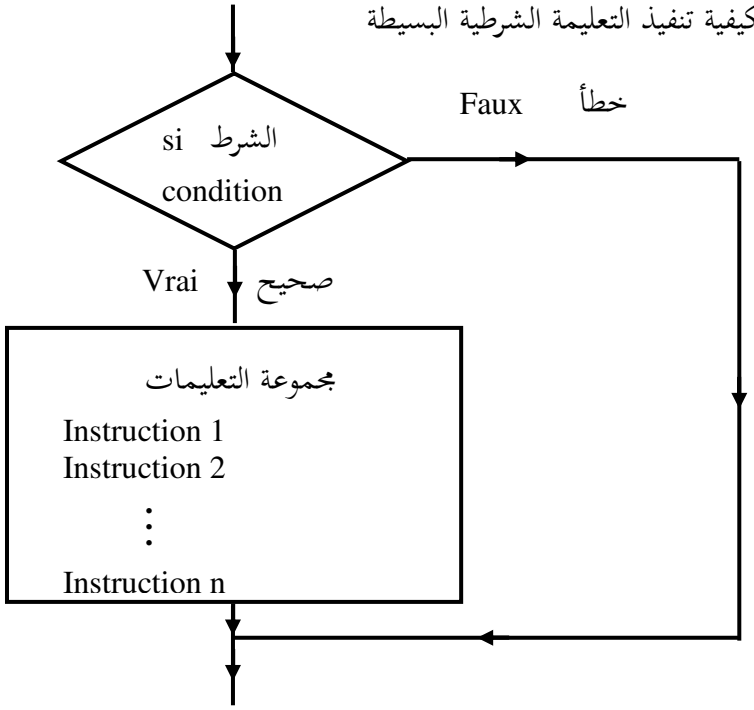
لتنفيذ هذه التعليمة في الخوارزمية نتبع الخطوات التالية:

نتحقق من الشرط، أي نحاول التأكد من صحته، فإذا كان الشرط صحيحا نقوم بتنفيذ مجموعة التعليمات التي تقع

داخل التعليمة الشرطية بشكل تسلسلي، الواحدة تلو الأخرى، حتى ننتهي منها ونصل إلى نهاية التعليمة الشرطية، ويفسر هذا على النحو التالي، نتحقق من الشرط (condition) في حالة تحققه أي (condition vraie) فمجموعة التعليمات التي تقع بين alors وfsi أي (Instruction 1, Instruction 2,…., Instruction n) سوف تنفذ حسب ترتيبها، أما في حالة عدم تحقق الشرط فهذه التعليمات لا تنفذ ولا يتم الدخول إلى داخل التعليمة الشرطية وتكمل الخوارزمية مسارها انطلاقا من كلمة fsi.

ولتوضيح عمل هذه التعليمة بشكل بياني مبسط ارتأينا إدراج الشكل التالي:

الشكل (3): شكل توضيحي لكيفية تنفيذ التعليمة الشرطية البسيطة



مثال تطبيقي:

الخوارزمية التالية تطلب من المستخدم إدخال عدد صحيح ثم تعلمه إن كان هذا العدد موجب.

```
Algorithme Test
var a : entier
début
    écrire ("Veuillez saisir un nombre entier SVP…")
    lire (a)
    si a  >= 0
    |  alors
    |écrire ("Le nombre saisi est positif !")
   fsi
   écrire ("Le nombre saisi est :", a)
fin
```

في المثال السابق نلاحظ أنه في حالة ما إذا كان الشرط صحيحا أي العدد المدخل موجب (a >= 0) فإن الخوارزمية ستعلمنا بأنه موجب، ثم تكمل مسارها بعد fsi بإعلامنا بقيمة العدد المدخل، أما في حالة ما إذا كان العدد المدخل غير ذلك فإن الخوارزمية لا تعطينا أي جواب وتكمل مسارها، كما في الحالة الأولى، بإعلامنا بقيمة العدد المدخل، لهذا فإن استعمال هذا النوع من التعليمات يعتبر نادرا بحيث قل ما نجد حالات تستدعي استعمال التعليمة الشرطية البسيطة،

فأغلب المسائل والمشاكل التي نواجهها تستوجب وجود حالتين على الأقل (ناجح-راسب، مقبول-مرفوض، صغير-كبير، صحيح-خطأ،...إلخ)، لهذا فقد دعت الحاجة إلى استخدام نوع آخر من التعليمات الشرطية التي تأخذ بعين الاعتبار تناوب حالتين، هذه التعليمة تعرف بالتعليمة المتناوبة.

II. التعليمة الشرطية المتناوبة:

على عكس سابقتها، فالتعليمة الشرطية المتناوبة تسمح بدراسة مسائل ومشاكل تستوجب وجود حالتين متعاكستين، بحيث يتناوب الحل بينهما، وهذا سبب تسميتها بالمتناوبة.

II.1 تعريف التعليمة الشرطية المتناوبة:

تعتمد هذه التعليمة في تنفيذها على التحقق من الشرط والذي يحتمل حالتين (صحيح أو خطأ)، ففي حال كان الشرط صحيحا تنفذ التعليمة العمل المقابل لهذه الحالة، أما في حال كان الشرط خاطئا، تنفذ التعليمة عملا آخر يتوافق مع هذه الحالة.

تكتب هذه التعليمة في الخوارزمية على الشكل التالي:

```
si (condition)
| alors
|    | Instruction 11
|    | Instruction 12
|    |      :
|    | Instruction 1n
| sinon
|    Instruction 21
|    Instruction 22
|         :
|    Instruction 2m
|
fsi
```

II.2. كيفية عمل التعليمة:

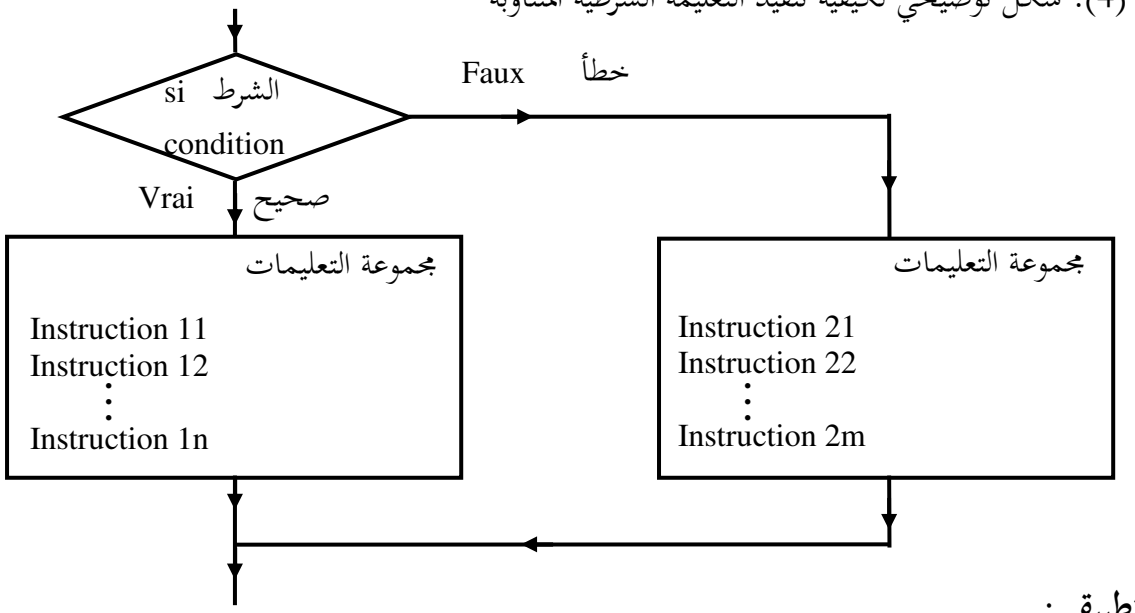
لتنفيذ هذه التعليمة في الخوارزمية نتبع الخطوات التالية:

نتحقق من الشرط، فنحاول التأكد من صحته، فإذا كان الشرط صحيحا نقوم بتنفيذ مجموعة التعليمات التي تقع داخل الجزء الأول من التعليمة الشرطية المتناوبة بشكل تسلسلي، الواحدة تلو الأخرى، حتى ننتهي منها ونصل إلى كلمة sinon والتي معناها انتهاء الجزء المتعلق بحالة تحقق الشرط، أما في حالة عدم تحقق الشرط وكان الشرط خاطئا، نقوم بتنفيذ الجزء الثاني من التعليمة الشرطية والمتعلق بعدم صحة الشرط، فنقوم بتنفيذ التعليمات الموجودة داخله، ويمكن تفسير عمل هذه التعليمة على الشكل التالي، نتحقق من الشرط (condition) في حالة تحققه أي (condition vraie) فمجموعة التعليمات التي تقع بين alors وsinon أي (Instruction 11, Instruction 12,…., Instruction 1n) سوف تنفذ

حسب ترتيبها، أما في حالة عدم تحقق الشرط فالمجموعة الثانية من التعليمات والتي تقع بين sinon وfsi أي (Instruction 21, Instruction 22,…., Instruction 2m) هي التي ستنفذ حسب ترتيبها داخل الخوارزمية، لذا فإن عملية التنفيذ ستمر لا محالة عبر متن التعليمة الشرطية سواء تحقق الشرط أم لم يتحقق، ولكن سيتم تنفيذ جزء واحد منها (الجزء المتعلق بالشرط).

ولتوضيح عمل هذه التعليمة بشكل بياني مبسط ارتأينا إدراج الشكل التالي:

الشكل (4): شكل توضيحي لكيفية تنفيذ التعليمة الشرطية المتناوبة



مثال تطبيقي:

الخوارزمية التالية تشبه نوعا ما الخوارزمية السابقة، لكنها تحتوي على اختلاف جوهري يتمثل في إمكانية إعلام المستخدم بنوع العدد المدخل سواء كان موجبا أو سالبا، وهي في هذه الحالة ستنفذ بطريقة مختلفة تماما على الخوارزمية السابقة، بحيث تطلب من المستخدم إدخال عدد صحيح ثم تعلمه إن كان هذا العدد موجبا أو سالبا.

```
Algorithme Test
var a : entier
début
    écrire ("Veuillez saisir un nombre entier SVP…")
    lire (a)
    si a  >= 0
    |  alors
    |    |  écrire ("Le nombre saisi est positif !")
    |    sinon
    |       écrire ("Le nombre saisi est négatif !")
    fsi
    écrire ("Le nombre saisi est :", a)
fin
```

في المثال السابق نلاحظ أنه في حالة ما إذا كان الشرط صحيحا أي العدد المدخل موجب (a >= 0) فإن الخوارزمية ستعلمنا بأنه موجب، أما في حالة ما إذا كان الشرط غير ذلك وهو في هذا الحالة نفي الشرط أي (a < 0) فإن

الخوارزمية سنفذ الجزء المتعلق بنفي الشرط والذي يقع بعد كلمة sinon وتعلمنا بأن العدد المدخل هو عدد سالب، ثم تكمل الخوارزمية مسارها بإعلامنا بقيمة العدد المدخل.

مع أن هذا النوع من التعليمات الشرطية كثير الاستعمال، إلا أنه عادة ما نواجه مسائل ومشاكل تستدعي دراسة حالات متعددة لا تدخل في دائرة التناوب، كـ (صغير-كبير-متوسط، موافق-معارض-محايد، موجب-سالب-معدوم، ...إلخ) لهذا فإن استعمال هذا النوع من التعليمات لا يمكن أن يفي بهذا الغرض، ويأتي نوع آخر من التعليمات الشرطية، والذي يسمح بدراسة حالات متعددة من خلال عملية التداخل والتركيب، أي تداخل تعليمة شرطية داخل أخرى وهو ما يعرف بالتعليمة الشرطية المتداخلة أو المركبة.

III. التعليمة الشرطية المتداخلة:

تعتبر التعليمة الشرطية المتداخلة من أكثر التعليمات الشرطية استعمالا، وفي المقابل أكثرها تعقيدا، بحيث يجب على مستعملها أن يتقن جيدا أساسيات المنطق الرياضي، بحيث عند استعمال مجوعة شروط يجب مراعاة إمكانية تحقق تلك الشروط ولا يضع شروطا لا يمكن تحققها أو في حالة نفي شرط يجب الإلمام بعملية النفي جيدا، ومثالا على ذلك الشرط المركب التالي، والذي لا يمكن تحققه على الإطلاق: (a >= 20 ET a <= 0)

هذا الشرط لا يمكن تحققه على الإطلاق بحيث لا يمكن لأي عدد أن يكون أصغر أو يساوي الصفر ويكون في نفس الوقت أكبر أو يساوي عشرون، أما نفي هذا الشرط فهو كالتالي: (a < 20 OU a > 0)

وهو شرط محقق على الدوام، بحيث أي عدد سيكون إما أكبر تماما من الصفر أو أصغر أو يساوي الصفر، ومجوعة الأعداد الأصغر تماما من عشرين تحتوي مجموعة الأعداد الأصغر أو يساوي الصفر.

III.1 تعريف التعليمة الشرطية المتداخلة:

على عكس التعليمة الشرطية البسيطة والتعليمة الشرطية المتناوبة، فإن التعليمة الشرطية المتداخلة يمكن أن تحتوي على عدة شروط، تجزئها إلى عدد من التعليمات الشرطية المركبة الواحدة داخل الأخرى بشكل متسلسل ومنطقي، وأبسط التعليمات الشرطية المتداخلة هي التي تحتوي على ثلاثة مجموعات من التعليمات، بحيث تعتمد في تنفيذها على التحقق من الشرط الأول الذي إن كان صحيحا يؤدي إلى تنفيذ المجموعة الأولى، أما إذا كان خاطئا فسيتم التحقق من الشرط الثاني الذي يتم التحقق منه بدوره فإذا كان صحيحا يؤدي إلى تنفيذ المجموعة الثانية أما إذا كان خاطئا، فسيؤدي إلى تنفيذ المجموعة الثالثة.

تكتب هذه التعليمة في الخوارزمية على الشكل التالي:

```
si (condition1)
  alors
    Instruction 11
    Instruction 12
        ⋮
    Instruction 1n
  sinon si (condition 2)
          alors
            Instruction 21
            Instruction 22
                ⋮
            Instruction 2m
          sinon
            Instruction 31
            Instruction 32
                ⋮
            Instruction 3p
        fsi
fsi
```

III.2 كيفية عمل التعليمة:

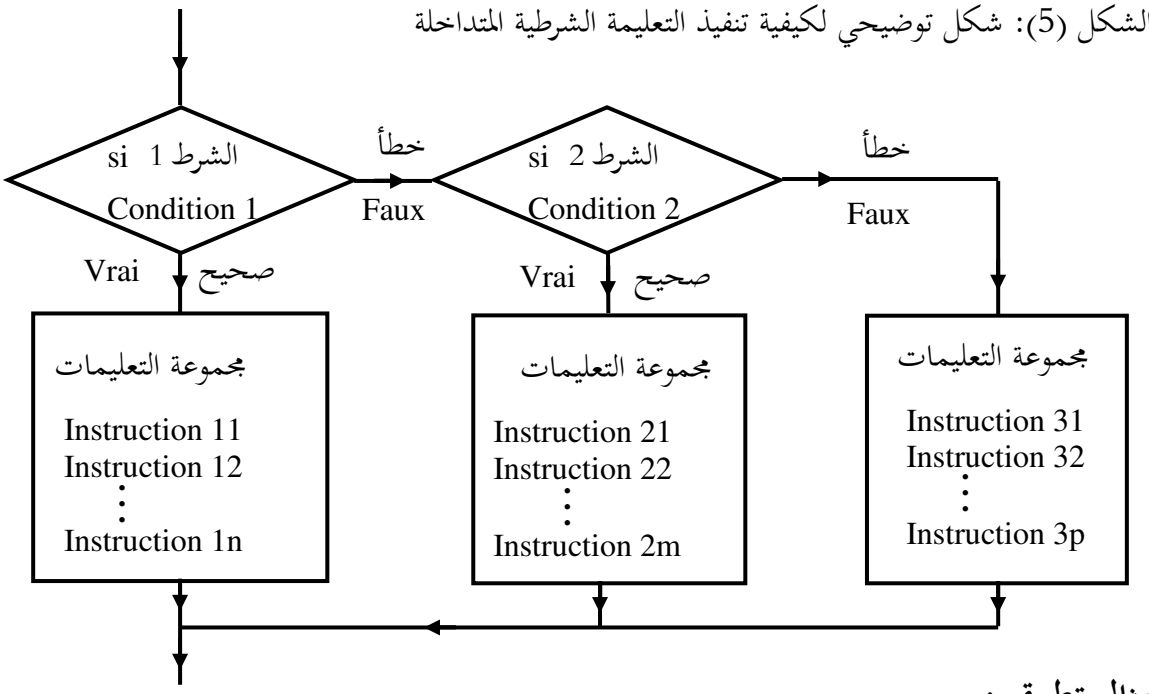
لتنفيذ هذه التعليمة في الخوارزمية نتبع الخطوات التالية:

نحاول التحقق من الشرط الأول، فإذا كان هذا الشرط صحيحا نقوم بتنفيذ مجموعة التعليمات التي تقع داخل الجزء الأول من التعليمة الشرطية المتناوبة بشكل تسلسلي، الواحدة تلو الأخرى، حتى ننتهي منها ونصل إلى كلمة sinon والتي معناها انتهاء الجزء المتعلق بحالة تحقق الشرط الأول، أما في حالة عدم التحقق الشرط والأول وكان هذا الشرط خاطئا، نقوم بتنفيذ الجزء الثاني من التعليمة الشرطية والمتعلق بعدم صحة الشرط الأول، فنقوم بالتحقق من الشرط الثاني والذي في حالة صحته نقوم بتنفيذ الجزء الثاني من التعليمة أما في حالة عدم تحققه فسنقوم بتنفيذ الجزء الثالث.

ويمكن تفسير عمل هذه التعليمة على الشكل التالي، نتحقق من صحة الشرط الأول (condition 1) في حالة تحققه أي (condition1 vraie) فمجموعة التعليمات التي تقع بين alors و sinon si (Instruction 11, Instruction 12,…., Instruction 1n) سوف تنفذ حسب ترتيبها، أما في حالة عدم تحقق الشرط الأول (condition 1 fausse) فيتم الانتقال إلى الجزء الثاني من التعليمة الشرطية ويتم التحقق من الشرط الثاني (condition 2) والذي في حالة تحققه أي (condition 2 vraie) فمجموعة التعليمات التي تقع بين sinon وalors (Instruction 21, Instruction 22,…., Instruction 2m) سوف تنفذ حسب ترتيبها، أما في حالة عدم تحقق الشرط االثاني (condition 2 fausse) فيتم الانتقال إلى الجزء الثالث من التعليمة الشرطية ويتم تنفيذ التعليمات التي تقع بين sinon و fsi (Instruction 31, Instruction 32,…., Instruction 3p).

ولتوضيح عمل هذه التعليمة بشكل بياني مبسط ارتأينا إدراج الشكل التالي:

الشكل (5): شكل توضيحي لكيفية تنفيذ التعليمة الشرطية المتداخلة



مثال تطبيقي:

الخوارزمية التالية تطلب من المستخدم إدخال عدد صحيح ثم تعلمه إن كان هذا العدد موجبا تماما أو سالبا أو معدوما، وهي بذلك تدرس الحالات الثلاثة للعدد الصحيح، بحيث تم فصل حالة العدد الموجب أو المعدوم إلى حالتين منفصلتين موجب تماما ومعدوم، ومن المعلوم أن الصفر هو عدد موجب.

```
Algorithme Test
var a : entier
début
    écrire ("Veuillez saisir un nombre entier SVP…")
    lire (a)
    si a  > 0
    |  alors
    |    |    écrire ("Le nombre saisi est strictement positif !")
    |  sinon si a < 0
    |              |    alors
    |              |      |    écrire ("Le nombre saisi est négatif !")
    |              |    sinon
    |              |          écrire ("Le nombre saisi est nul !")
    |          fsi
    fsi
    écrire ("Le nombre saisi est :", a)
fin
```

في المثال السابق نلاحظ وجود شرطين يتمثلان في: (a > 0) و(a < 0) وأنه في حالة ما إذا كان الشرط الأول صحيحا أي العدد المدخل موجبا تماما (a > 0) فإن الخوارزمية ستعلمنا بأنه موجب تماما، أما في حالة ما إذا كان الشرط الأول غير صحيح (وهو في هذه الحالة نفي الشرط الأول أي (a <= 0) فإن الخوارزمية ستقسم هذه الحالة إلى قسمين هما: a (0 > و(a = 0)، ثم نفذ الجزء المتعلق بحالة (a < 0) والذي يقع بعد كلمتي sinon si وتعلمنا بأن العدد المدخل هو عدد سالب، ثم في حالة عدم تحق هذا الشرط (a < 0) فإن الحالة الأخيرة هي a = 0 بحيث العدد الذي لا يكون

موجبا تمام ولا يكون سالبا فهو بدون أدنى شك عدد معدوم، وبذلك ستعلمنا أن العدد المدخل هو عدد معدوم، ومن ثم ستكمل الخوارزمية مسارها وتعلمنا بقيمة العدد المدخل.

الجدير بالذكر أن التعليمة الشرطية المتداخلة قد تمكن من حل المسائل ذات عدة حالات واحتمالات إلا أنه في حالة ما إذا تعدى عدد الحالات حد معين يصبح التعامل مع هذه التعليمة صعبا للغاية ويحتمل الوقوع في الأخطاء المنطقية وأخطاء الكتابة، وكمثال تطبيقي، نأخذ الخوارزمية التي تعلمنا بتقدير طالب في شهادة البكالوريا، بحيث إذا تحصل على معدل أكبر أو يساوي 18 وأصغر أو يساوي 20 فإن الطالب يمنح له تقدير ممتاز، أما إذا كان معدله أصغر تماما من 18 وأكبر أو يساوي 16 فإن تقديره هو جيد جدا، وفي حالة ما إذا كان معدله أصغر تمام من 16 وأكبر أو يساوي 14 فأن الطالب يمنح له تقدير جيد، وإذا كان المعدل المتحصل عليه أصغر تماما من 14 وأكبر أو يساوي 12 يمنح للطالب تقدير قريب من الجيد، وفي حالة ما إذا كان معدل الطالب أصغر تماما من 12 وأكبر أو يساوي 10 فإن الطالب يمنح له تقدير مقبول، وفي حالة ما إذا كان معدل الطالب أصغر تماما من 10 وأكبر أو يساوي 0 فإن الطالب يعتبر راسبا، ما عدا ذلك يعتبر المعدل المدخل للخوارزمية خاطئا.

انطلاقا من هذه المعطيات سنحاول كتابة نسخة أولية لخوارزمية تترجم المطلوب منها حسب الحالات المذكورة سابقا وباستعمال التعليمة الشرطية المتداخلة والشروط التي جاءت في متن المسألة، بحيث كل شرط مكون من مقارنتين (20=>معدل => 18 و18>معدل => 16 ....وهكذا):

```
Algorithme Mention
var moyenne : réel
début
    écrire ("Veuillez saisir la moyenne de l'élève SVP…")
    lire (moyenne)
    si (moyenne  >= 18) ET (moyenne  <= 20)
    |  alors
    |  |    écrire ("Mention Excellent !")
    |  sinon si (moyenne  >= 16) ET (moyenne  < 18)
    |  |        alors
    |  |        |   écrire ("Mention Très bien !")
    |  |        sinon  si (moyenne  >= 14) ET (moyenne  < 16)
    |  |        |        alors
    |  |        |        |   écrire ("Mention Bien !")
    |  |        |        sinon  si (moyenne  >= 12) ET (moyenne  < 14)
    |  |        |        |        alors
    |  |        |        |        |   écrire ("Mention Assez bien !")
    |  |        |        |        sinon  si (moyenne  >= 10) ET (moyenne  < 12)
    |  |        |        |        |        alors
    |  |        |        |        |        |   écrire ("Mention Passable !")
    |  |        |        |        |        sinon  si (moyenne  >= 0) ET (moyenne  < 10)
    |  |        |        |        |        |        alors
    |  |        |        |        |        |        |   écrire ("Ajourné !")
    |  |        |        |        |        |        sinon
    |  |        |        |        |        |        |   écrire ("Saisie erronée !")
    |  |        |        |        |        |        fsi
    |  |        |        |        |        fsi
    |  |        |        |        fsi
    |  |        |        fsi
    |  |        fsi
    |  fsi
fin
```

الملاحظ في هذه الخوارزمية العدد الكبير من التعليمات الشرطية المتداخلة الواحدة داخل الأخرى وكذلك الشروط المركبة بحيث يصعب على كاتب الخوارزمية مراقبة تسلسل التنفيذ، كما يصعب عليه مراقبة عمل الخوارزمية، فإذا حدث خطأ في كتابة أحد التعليمات فإن الخطأ سيكون من الصعب اكتشافه، كما أن الهدف الأساسي من كتابة الخوارزمية هو كتابة البرنامج والذي بدوره يعطي الحلول للمسائل المطروحة له، وفي حالة الخوارزمية السابقة سيكون من الصعب صيانة البرنامج أو تعديله دون الوقوع في أخطاء منطقية وكتابية، من أجل هذا، سنحاول كتابة نسخة محسنة للخوارزمية السابقة نستعمل فيها شروط غير مركبة وبسيطة وسهلة المراقبة والمتابعة.

```
Algorithme Mention
var moyenne : réel
début
    écrire ("Veuillez saisir la moyenne de l' élève SVP…")
    lire (moyenne)
    si (moyenne  > 20) OU (moyenne  < 0)
    | alors
    |    | écrire "("Saisie erronée !")
    | sinon si (moyenne  >= 18)
    |          | alors
    |          |    | écrire ("Mention Excellent !")
    |          | sinon  si (moyenne  >= 16)
    |          |          | alors
    |          |          |    | écrire ("Mention Très bien !")
    |          |          | sinon  si (moyenne  >= 14)
    |          |          |          | alors
    |          |          |          |    | écrire ("Mention bien !")
    |          |          |          | sinon  si (moyenne  >= 12)
    |          |          |          |          | alors
    |          |          |          |          |    | écrire ("Mention Assez bien !")
    |          |          |          |          | sinon  si (moyenne  >= 10)
    |          |          |          |          |          | alors
    |          |          |          |          |          |    | écrire ("Mention Passable ! ")
    |          |          |          |          |          | sinon
    |          |          |          |          |          |       écrire ("Ajourné !")
    |          |          |          |          |          fsi
    |          |          |          |          fsi
    |          |          |          fsi
    |          |          fsi
    |          fsi
    fsi
fin
```

الملاحظ في هذه الخوارزمية أنها احتفظت كسابقتها بنفس عدد التعليمات الشرطية المتداخلة بحيث احتوت على ستة تعليمات شرطية متناوبة ومتداخلة الواحدة داخل الأخرى، لكنها في المقابل خففت من عدد المقارنات داخل الشروط، فقد اشتملت على خمسة مقارنات بسيطة وواحدة فقط مركبة، بينما احتوت النسخة الأولى من الخوارزمية على ستة مقارنات كلها مركبة وهو ما يزيد من درجة تعقيدها ويزيد من الوقت اللازم لتنفيذها.

ومحاولة منا شرح طريقة عمل الخوارزمية وكيف أنها خففت من عدد المقارنات بينما أدت إلى نفس نتيجة الخوارزمية السابقة، سنبدأ بأول مقارنة وهي مقارنة مركبة جاءت على النحو التالي:

(moyenne > 20) OU (moyenne < 0)

يفسر هذا الشرط المركب على عدم قبول معدل الطالب المدخل من قبل المستخدم، بحيث لا يمكن لمعدل طالب أن يكون أكبر من 20 فهي أقصى علامة يمكن التحصل عليها من 20 كما لا يمكن للطالب التحصل على علامة سالبة وأصغر من الصفر، لذا فقد جاء جواب هذا الشرط أن رصد العلامة خاطئ، أما نفي هذا الشرط والذي جاء مباشرة بعد كلمة sinon الأولى فمعناه أن المعدل المرصود هو داخل المجال [0 , 20] أي أن: (moyenne >=0) ET (moyenne <=20)، ثم إذا وضعنا بعده شرط (moyenne >= 18) فهو بالضرورة سيكون (moyenne <=20) لذلك فلا داعي لإضافته في الشرط، ونفس الملاحظة تنطبق على باقي المقارنات فنفي (moyenne >= 18) سيكون (moyenne < 18) ثم نضيف له (moyenne >= 16) وهكذا حتى نهاية الخوارزمية.

ومع أن هذه الخوارزمية تعتبر نسخة محسنة عن سابقتها إلا أنه ما يعاب عليها كثرة التعليمات الشرطية المتداخلة وصعوبة التحكم في كتابتها وكذلك إن صادفنا مسألة تستدعي مثلا 20 تعليمات شرطية متداخلة أو أكثر فسيكون من الصعب كتابتها ومراقبتها، لذا فقد جاءت التعليمة selon لتحل هذا الإشكال، بحيث بفضل كتابتها البسيطة والمرتبة أمكن كتابة تعليمات شرطية تحتوي على عدد كبير من الحالات دون الوقوع في مشكلة الكتابة المعقدة والطويلة للخوارزميات.

IV. التعليمة الشرطية selon:

تعتبر التعليمة selon نسخة محسنة من التعليمة الشرطية المتداخلة، فهي تؤدي دورها لكن بطريقة أكثر تنظيما وبساطة.

1.IV تعريف التعليمة الشرطية selon:

تستعمل هذه التعليمة عادة عندما يكون عدد الحالات التي تستدعي الدراسة كبير، بحيث يصعب على التعليمة الشرطية المتداخلة التعامل معه بسهولة، وتكتب هذه التعليمة في الخوارزمية على الشكل التالي:

```
Selon
   Cas (condition 1) :
       Instruction 11
       Instruction 12
          .
          .
       Instruction 1n
   Cas (condition 2) :
       Instruction 21
       Instruction 22
          .
          .
       Instruction 2m
Cas (condition 3) :
       Instruction 31
       Instruction 32
          .
          .
```

```
        Instruction 3p
    Cas (condition k) :
        Instruction k1
        Instruction k2
            :
        Instruction ki
  Autrement :
        Instruction a1
        Instruction a2
            :
        Instruction aj
fin selon
```

إذا فالتعليمة selon يمكن أن تحتوي على عدة أجزاء من تعليمات شرطية مرتبة ومنظمة بحيث يسهل كتابتها وفهمها وتنفيذها دون التباس أو خلط، وهي في نهاية الأمر تعمل عمل التعليمة الشرطية المتداخلة.

2.IV كيفية عمل التعليمة:

لتنفيذ هذه التعليمة في الخوارزمية نتبع الخطوات التالية:

نتحقق من الشرط الأول (condition 1)، فإذا كان هذا الشرط صحيحا نقوم بتنفيذ مجموعة التعليمات التي تقع داخل الجزء الأول من هذه التعليمة بشكل تسلسلي، حتى ننتهي منها ثم نخرج من التعليمة selon ونكمل سير الخوارزمية بعد كلمة fin selon، أما في حالة عدم تحقق الشرط الأول، نقوم بتنفيذ الجزء الثاني من التعليمة الشرطية والمتعلق بالشرط الثاني والذي في حالة صحته نقوم بتنفيذ الجزء الثاني من التعليمة أما في حالة عدم تحققه فسنقوم بتنفيذ الجزء الثالث، وهكذا حتى استنفاذ كل الشروط وكل أجزاء التعليمة والوصول إلى كلمة Autrement والتي معناها متممة الشروط السابقة أو الشروط غير المذكورة من قبل، ثم بعد تنفيذ كل التعليمات الخاصة بالجزء الذي يأتي بعد كلمة Autrement نخرج من التعليمة selon ونكمل مسار الخوارزمية بعد كلمة fin selon.

مثال تطبيقي:

في هذا المثال سنحاول كتابة الخوارزمية التي تعلمنا بتقدير طالب في شهادة البكالوريا باستعمال التعليمة selon، بعدما كتبنها سابقا باستعمال التعليمة الشرطية المتداخلة.

```
Algorithme Mention
var moyenne : réel
début
    écrire ("Veuillez saisir la moyenne de l'élève SVP…")
    lire (moyenne)
   cas moyenne <= 20 ET moyenne >= 18 :
       écrire ("Mention Excellent !")
   cas moyenne  <  18 ET moyenne >= 16 :
       écrire ("Mention Très bien !")
   cas moyenne <  16 ET moyenne >= 14 :
       écrire ("Mention bien !")
   cas moyenne < 14 ET moyenne >= 12 :
```

```
        écrire ("Mention Assez bien !")
    cas moyenne < 12 ET moyenne >= 10 :
        écrire ("Mention Passable !")
    cas moyenne < 10 ET moyenne >= 0 :
        écrire ("Ajourné !")
    Autrement :
        écrire "("Saisie erronée !")
   fin selon
fin
```

الملاحظ في الخوارزمية أعلاه أنها جاءت بسيطة وسهلة القراءة والتفسير وكذا التنفيذ، وهو ما يجعل هذا النوع من التعليمات الشرطية كثير الاستعمال عند تعدد الحالات المرتبطة بالشروط، والجدير بالذكر كذلك أن هذا النوع من التعليمات قد تختلف كتابته من لغة برمجة إلى أخرى، ففي لغة الباسكال نجده يكتب (case of) أما في لغة C أو C++ فيكتب (switch).

V. تمارين مقترحة:

التمرين الأول:

أكتب خوارزمية تطلب من المستخدم إدخال ثلاثة نقاط Note3 ، Note2 ، Note1 ثم إدخال معاملاتها coef3، coef2 ، coef1. وتقوم بعد ذلك بحساب المعدل وعرضه على الشاشة. كما تقوم الخوارزمية بعرض كلمة "Admis (e)" في حالة معدل أكبر أو يساوي 10 كما تقوم بعرض كلمة "Ajourné (e)" في حالة ما إذا كان المعدل أصغر تماما من 10، قم بعملية المحاكاة (déroulement) من أجل القيم التالية:

coef1 = 3 ، coef2 = 1، coef3 = 2 و Note1 = 12، Note2 = 14، Note3 = 13

التمرين الثاني:

أكتب خوارزمية تسمح لنا بتبديل محتوى متغيرتين A و B.

- **الحالة الأولى**: باستعمال متغيرة مؤقتة
- **الحالة الثانية**: بدون استعمال المتغيرة المؤقتة

التمرين الثالث:

أكتب خوارزمية تقوم بحساب وعرض $|x|$ على الشاشة بدون استعمال الدالة ABS( ) وباستعمال التعليمة SI البسيطة دون استعمال Sinon.

التمرين الرابع:

أكتب خوارزمية تقوم بحل معادلة من الدرجة الأولى من الشكل $ax + b = 0$ قم بعملية المحاكاة (déroulement) لكل

الحالات الممكنة.

التمرين الخامس:

أكتب خوارزمية تقوم بحل معادلة من الدرجة الثانية من الشكل $ax^2 + bx + c = 0$ قم بعملية المحاكاة (déroulement) لكل الحالات الممكنة.

التمرين السادس:

أكتب خوارزمية تطلب من المستخدم إدخال عدد طبيعي ثم تكتب على الشاشة''nombre paire'' إذا كان زوجيا أو ''nombre impaire'' إذا كان فرديا.

التمرين السابع:

أكتب خوارزمية تقوم بالبحث عن أصغر قيمة بين ثلاثة أعداد صحيحة A ، B و C معطاة من طرف المستخدم وتعرض النتيجة على الشاشة.

- باستعمال متغير اسمه Min.
- بدون استعمال المتغير.
- ماذا نغير في الخوارزمية إذا أردنا البحث عن أكبر قيمة.

التمرين الثامن:

أكتب خوارزمية تطلب من المستخدم إدخال إيرادات و نفقات مؤسسة معينة ثم تقوم بحساب الفرق بين الإيرادات والنفقات انطلاقا من المعادلة التالية:

Ecart = Recettes – Dépenses

ثم تعلم المستخدم إن كانت المؤسسة رابحة أم خاسرة أم متوازنة حسب قيمة الفرق (Ecart):

مؤسسة رابحة (Entreprise bénéficiaire) إذا كان الفرق موجبا.

مؤسسة خاسرة (Entreprise déficitaire) إذا كان الفرق سالبا.

مؤسسة متوازنة (Entreprise équilibrée) إذا كان الفرق معدوما.

التمرين التاسع:

أكتب خوارزمية تطلب من المستخدم إدخال ثلاثة أعداد صحيحة ثم تقوم بترتيبها ترتيبا تصاعديا وتعرضها على الشاشة حسب هذا الترتيب.

التمرين العاشر:

يقوم محاسب شركة Easysoft المختصة في البرمجيات بحساب المنحة العائلية لكل موظفيه حسب الطريقة التالية:

- إذا كان عدد الأولاد أصغر أو يساوي 3 تمنح له منحة قدرها 600,00 دج عن كل ولد.
- وإذا كان أكبر تماما من 3 وأصغر أو يساوي 6 تمنح له منحة قدرها 500,00 دج عن كل ولد.
- أما إذا كان أكبر تماما من 6 تمنح له منحة قدرها 450,00 دج عن كل ولد على أن لا تتعدي قيمة المنحة 3 600,00 دج.

أكتب خوارزمية تطلب من هذا المحاسب إدخال عدد أولاد الموظف، ثم تحسب وتعرض على الشاشة قيمة منحته العائلية.

التمرين الحادي عشر:

أكتب خوارزمية تحسب و تعرض على الشاشة المبلغ الصافي الواجب دفعه من طرف الزبون (Montant net à payer) بعد إدخال المعلومات التالية من طرف المستخدم:

نسبة الرسم على القيمة المضافة (Taux TVA) تقدر بـ 19% .

نفقات النقل تقدر بـ 5 000,00 دج لكل 100 كم، يجب على المستخدم إدخال المسافة التي قطعتها البضاعة.

يطبق التخفيض (Remise) على المبلغ غير الخاضع للضريبة (Montant HT) حسب المعطيات التالية:

إذا كان: Montant HT > 200 000,00 DA فإن مبلغ التخفيض يقدر بـ 10%

أما إذا كان: 100 000,00 DA ≤ Montant HT ≤ 200 000,00 DA فإن مبلغ التخفيض يقدر بـ 5%

وأما إذا كان: 50 000,00 DA ≤ Montant HT ≤ 100 000,00 DA فإن مبلغ التخفيض يقدر بـ 2%

وفي حالة ما إذا كان المبلغ غير الخاضع للضريبة أصغر تماما من 50 000,00 DA فلا يوجد تخفيض.

يجب على المستخدم إدخال الكمية المطلوبة والسعر الوحدوي: Qte, PU

المبلغ غير الخاضع للضريبة = الكمية × السعر الوحدوي: Montant HT = Quantité × PU

الرسم على القيمة المضافة = (المبلغ الغير خاضع للضريبة – مبلغ التخفيض + نفقات النقل)× 19% .

TVA = (Montant HT – Montant Remise + Frais de transport) × 19%

نفقات النقل = $\frac{\text{المسافة} \times 5\,000}{100}$ Frais de transport = $\frac{\text{Distance} \times 5\,000}{100}$

المبلغ الصافي الواجب دفعه = المبلغ غير خاضع للضريبة – مبلغ التخفيض + نفقات النقل + الرسم على القيمة المضافة.

Montant net à payer = Montant HT – Montant Remise + Frais de transport + TVA.

VI. حلول التمارين:

حل التمرين الأول:

```
Algorithme Moyenne
1) var  Note1, Note2, Note3, Moy : réel
        coef1, coef2, coef3 : entier
début
2)      écrire ("Veuillez saisir les trois notes SVP…")
3)      lire (Note1, Note2, Note3)
4)     écrire ("Veuillez saisir leurs coefficients respectifs SVP…")
5)    lire (coef1, coef2, coef3)
6)  Moy ← Note1 * coef1 + Note2 * coef2 + Note3 * coef3
7)  Moy ← Moy/(coef1 + coef2 + coef3)
8)   écrire ("La moyenne de cet étudiant (e) est :", Moy)
9)  si Moy  >=  10
    |  alors
10) |    |        écrire ("Cet étudiant (e) est : Admis( e) ")
    |  sinon
11) |            écrire ("Cet étudiant (e) est : Ajourné( e) « )
    fsi
fin
```

عملية المحاكاة (déroulement):

سيتم إجراء عملية المحاكاة على المتغيرة التي تتغير قيمها داخل الخوارزمية، وهذا من خلال التعليمات المختلفة التي تستخدمها وتؤثر في محتواها، وهي في هذه الحالة المتغيرة Moy، لأن المتغيرات الأخرى (Note1, Note2, Note3, coef1, coef2, coef3) يتم إدخالها من طرف المستخدم ولا تتغير داخل الخوارزمية.

Note1 = 12، Note2 = 14، Note3 = 13 و coef1 = 3 ، coef2 = 1، coef3 = 2

تأخذ هذه المتغيرات القيم أعلاه بعد عمليتي القراءة مباشرة، التعليمة (3) والتعليمة (5)، أما المتغيرة Moy فتبدأ بالتغير انطلاقا من التعليمة رقم (6).

العرض على الشاشة

```
Veuillez saisir les trois notes SVP…
12
14
13
Veuillez saisir leurs coefficients respectifs SVP…
3
1
2
La moyenne de cet étudiant (e) est :  12.67
Cet étudiant (e) est : Admis( e)
```

محاكاة التنفيذ (في الذاكرة)

N°	Moy
6	76
7	12.67

مناقشة الحل:

الجدير بالذكر أنه في عملية المحاكاة، تعليمتي القراءة والكتابة لا يمكن إظهارها في جدول تغيرات القيم والذي يتم ربطه بالذاكرة، لأن تعليمة القراءة والكتابة هي تعليمات إدخال وإخراج لها علاقة بأجهزة الإدخال والإخراج والتي تكون غالبا الشاشة ولوحة المفاتيح، لذا ففي حالة التمرين السابق لا يظهر أثر هاتين التعليمتين إلا من خلال الشاشة.

حل التمرين الثاني:

```
Algorithme Exchange_1
var  A, B, Inter : entier
début
  Inter  ← A
  A ← B
  B ← Inter
Fin

Algorithme Exchange_2
var  A, B : entier
début
  A  ← A + B
  B ← A - B
  A ← A - B
fin
```

مناقشة الحل:

في الخوارزمية الأولى لتغيير محتوى متغيرتين فيما بينهما استعملنا متغيرة مؤقتة، دورها حفظ محتوى المتغيرة A قبل أن يتم فقدانه بمحتوى المتغيرة B، ثم نضع محتوى المتغيرة المؤقتة في المتغيرة B وبهذا نكون قد وضعنا محتوى المتغيرة A في المتغيرة B ومحتوى المتغيرة B في المتغيرة A، وحتى نتأكد من عمل هذه الخوارزمية سنقوم بعملية محاكاة للمتغيرتين A = 3 وB=5 كما يلي:

A	B	Inter
3	5	?
3	5	3
5	5	3
5	3	3

الملاحظ أنه بعد نهاية الخوارزمية أصبح A=5 وB=3 وهذا هو المطلوب.

في الخوارزمية الثانية لتغيير محتوى المتغيرتين لم نستعمل متغيرة مؤقتة ولكن للحفاظ على قيمة A قبل أن يتم فقدانها استعملنا عملية الجمع والتي تحتفظ بقيمة A داخل الحاصل ثم يتم استرجاعه من خلال عملية طرح، ولنتأكد من صحة الخوارزمية ومن أنها تقوم بالعمل المطلوب، سنقوم بعملية محاكاة باستعمال نفس القيم السابقة كما يلي:

A	B
3	5
8	5
8	3
5	3

الملاحظ أنه بعد نهاية الخوارزمية أصبح A=5 وB=3 وهذا هو المطلوب.

حل التمرين الثالث:

```
Algorithme val_abs
var  x, abs : entier
début
  écrire ("Veuillez saisir un entier SVP")
  lire (x)
  abs ← x
  si x < 0
  |  alors
  |   abs ← -x
  fsi
 écrire ("La valeur absolue de :", x, "est :", abs)
fin
```

مناقشة الحل:

بعد قراءة العدد المدخل من طرف المستخدم تقوم الخوارزمية بحفظ العدد داخل المتغيرة abs ثم إذا كان هذا العدد سالبا يُضرب في الناقص لتغيير إشارته إلى الموجب، أما إذا كان غير ذلك فالخوارزمية لا تغير من قيمته، وبذلك عند عرض النتيجة نستعمل قيمة العدد (x) والتي لم تتغير بينما abs تتغير حسب إشارة x.

حل التمرين الرابع:

كتابة خوارزمية تقوم بحل معادلة من الدرجة الأولى من الشكل $ax + b = 0$.

في الرياضيات المعادلة من الدرجة الأول تحل بشكل بسيط كالآتي:

$ax + b = 0$، إذن $x = -b/a$ بشرط أن تكون a وb معرفتين، مثلا $2x + 3 = 0 \Rightarrow x = -3/2$ ، لكن في الخوارزميات الغاية من كتابة هذه الخوارزمية هو اقتراح حلول لكل أشكال هذه المعادلة، بحيث يقوم المستخدم بإدخال قيم a وb وتقوم الخوارزمية بإعطائه حلول المعادلة، بما في ذلك الحالات الخاصة.

ولدراسة كل الحالات الممكنة المتعلقة بقيم a وb يجب أن نفرق بين حالة القيم المعدومة وغير المعدومة وهي كالآتي:

- **الحالة الأولى:** a <> 0 وb <> 0، في هذه الحالة مجموعة الحلول هي: {-b/a } بحيث المعادلة تقبل حلا واحدا ووحيدا؛
- **الحالة الثانية:** a <> 0 وb = 0، في هذه الحالة مجموعة الحلول هي: {-0/a = 0 } بحيث المعادلة تقبل حلا واحدا ووحيدا؛
- **الحالة الثالثة:** a = 0 وb <> 0، في هذه الحالة المعادلة لا تقبل حلولا ومجموعة الحلول هي مجموعة خالية: ∅.

- **الحالة الرابعة:** a = 0 و b = 0، في هذه الحالة كل الأعداد الحقيقية تحقق المعادلة ومجموعة الحلول هي]-∞, +∞[والمعادلة تقبل مجموعة لا منتهية من الحلول؛

يمكن اختصار عدد الحالات لتصبح ثلاث حالات عوض أربعة، بحيث يتم دمج الحالة الأولى والثانية لتصبح حالة واحدة كالآتي: في حالة a <> 0 فإن المعادلة تقبل حلا واحدا ووحيدا هو x = -b/a، بحيث إذا كان b = 0 يصبح الحل 0 وهو -0/a.

انطلاقا من هذه الحالات الثلاثة يمكن أن نكتب خوارزمية باستعمال تعليمة شرطية متداخلة تكون كالآتي:

```
si a <> 0
  alors
     x ← -b/a
sinon si b <> 0
          alors
             المعادلة لا تقبل حلولا
          sinon
             المعادلة تقبل مجموعة غير منتهية من الحلول
       fsi
fsi
```

انطلاقا من المعطيات السابقة يمكن كتابة الخوارزمية على الشكل النهائي التالي:

```
Algorithme eq_prem_deg
var  a, b, x : réel
début
  écrire ("Résolution d'une équation du premier degré du type ax+b = 0 ")
  écrire ("Veuillez entrer les valeur de a et b ")
  lire (a, b)
   si a <> 0
     alors
1)       x ← -b/a
2)       écrire ("L'équation admet une solution unique égale à :  ", x)
     sinon si b <> 0
             alors
3)              écrire ("L'équation n'admet aucune solution ")
             sinon
4)              écrire ("L'équation admet une infinité de solutions ")
          fsi
  fsi
fin
```

مناقشة الحل:

إذا كانت المتغيرة a <> 0 في هذه الحالة لا داعي لدراسة حالة المتغيرة b بحيث الحل هو مباشرة –b/a بما في ذلك الحالة b = 0.

أما إذا كانت المتغيرة a = 0 وهو يمثل نفي a <> 0 أي ما يأتي بعد كلمة (sinon)، في هذه الحالة يجب دراسة حالات المتغيرة b، b <> 0 و b = 0.

عملية المحاكاة لكل الحالات الممكنة:

في حالة a = 2 و b= -4 المعادلة هي من الشكل $2x - 4 = 0$ التعليمة رقم (1) هي التي ستنفذ لأنها تدخل ضمن حالات a <> 0 وبذلك $x \leftarrow -4/2$ أي $x \leftarrow 2$ ثم يتم تنفيذ التعليمة رقم (2) وهي التي ستعرض على الشاشة التعليق والنتيجة.

في حالة a = 0 و b= 5 التعليمة رقم (3) هي التي ستنفذ لأنها تدخل ضمن حالات a = 0 و b <> 0 وبذلك يتم عرض التعليق الذي يعلمنا بعدم وجود حلول.

في حالة a = 0 و b= 0 التعليمة رقم (4) هي التي ستنفذ لأنها تدخل ضمن حالات a = 0 و b = 0 وبذلك يتم عرض التعليق الذي يعلمنا بوجود مجموعة لا منتهية من حلول.

وكما أسلفنا في مستهل هذه المطبوعة، فإن حلول مسألة معينة في الخوارزميات قد تكون متعددة وينتج عنها مجموعة كبيرة من الخوارزميات تؤدي نفس الدور وتعطي نفس النتائج، فالمسألة السابقة يمكن أن تحل بطريقة أخرى، انطلاقا من الحالات الثلاثة السالفة الذكر، بحيث نبدأ بحالة a = 0 ثم نختم بحالة a <> 0، ويمكن أن نكتب على هذا الأساس خوارزمية باستعمال تعليمة شرطية متداخلة تكون كالآتي:

```
si a = 0
  alors  si b = 0
           alors
                                المعادلة تقبل مجموعة غير منتهية من الحلول
           sinon
                                المعادلة لا تقبل حلولا
         fsi
  sinon
         x ← -b/a
fsi
```

انطلاقا من المعطيات السابقة يمكن كتابة الخوارزمية على الشكل النهائي التالي:

```
Algorithme eq_prem_deg
var  a, b, x : réel
début
  écrire ("Résolution d'une équation du premier degré du type ax+b = 0 ")
  écrire ("Veuillez entrer les valeur de a et b ")
  lire (a, b)
  si a = 0
    alors  si b = 0
             alors
                 écrire ("L'équation admet une infinité de solutions ")
             sinon
                 écrire ("L'équation n'admet aucune solution ")
           fsi
    sinon
           x ← -b/a
           écrire ("L'équation admet une solution unique égale à :  ", x)
  fsi
fin
```

حل التمرين الخامس:

لكتابة خوارزمية تقوم بحل معادلة من الدرجة الثانية من الشكل $ax^2 + bx + c = 0$ نتبع نفس الإستراتيجية التي اتبعناها في كتابة الخوارزمية التي تقوم بحل معادلة من الدرجة الأولى، لذا سنقوم أولا بتعداد الحالات المتعلقة بالمتغيرات a, b, c بحيث كل متغيرة تأخذ حالتين (تساوي الصفر وتختلف عن الصفر) وبما أن عدد المتغيرات هو ثلاثة إذن فعدد الحالات هو 2^3 أي ثمان حالات:

- a = 0, b = 0, c = 0؛
- a = 0, b = 0, c <> 0؛
- a = 0, b <> 0, c = 0؛
- a = 0, b <> 0, c <> 0؛
- a <> 0, b = 0, c = 0؛
- a <> 0, b = 0, c <> 0؛
- a <> 0, b <> 0, c = 0؛
- a <> 0, b <> 0, c <> 0؛

الملاحظ في هذه الحالات الثمانية أنه يمكن تقسيمها إلى قسمين رئيسيين، قسم يتعلق بـ a = 0 وقسم آخر يتعلق بـ a <> 0، بحيث القسم الذي يتعلق بـ a = 0 تصبح فيه المعادلة $ax^2 + bx + c = 0$ $bx + c = 0$ وهي معادلة من الدرجة الأولى سبق وأن كتبنا الخوارزمية الخاصة بها، أما القسم الذي يتعلق بـ a <> 0 فهو متعلق بمعادلة من الدرجة الثانية مهما تكن قيم b وc، والمعادلة من الدرجة الثانية يتم حلها عن طريق حساب المميز والذي يضفي إلى ثلاثة حالات حسب إشارة المميز: موجب، سالب ومعدوم، وبهذا تصبح الخوارزمية تحتوي على ستة حالات، ثلاثة حالات خاصة بالمعادلة من الدرجة الأولى أي لما يكون a = 0 وثلاثة حالات خاصة بإشارة المميز عندما يكون a <> 0.

انطلاقا من المعطيات السابقة يمكن كتابة الخوارزمية على الشكل النهائي التالي:

```
Algorithme eq_prem_deg
var  a, b, c,  x1, x2, Delta : réel
début
   écrire ("Résolution d'une équation du premier degré du type ax² + bx + c = 0 ")
   écrire ("Veuillez entrer les valeurs de a et b et c ")
   lire (a, b, c)
    si a = 0
        alors
1)          écrire ("L'équation devient une équation du premier degré du type bx + c = 0 ")
            si b <> 0
                alors
2)                  x1 ← -c/b
3)                  écrire ("L'équation admet une solution double x1 = x2 =  ", x1)
                sinon  si c <> 0
                            alors
4)                              écrire ("L'équation n'admet aucune solution ")
                            sinon
5)                              écrire ("L'équation admet une infinité de solutions ")
                        fsi
            fsi
```

```
        sinon
6)          delta ← b * b - 4 * a *c
            si delta  <  0
                alors
7)                écrire ("L'équation n'admet aucune solution ")
                sinon si delta = 0
                        alors
8)                        x1 ← -b/(2*a)
9)                        écrire ("L'équation admet une solution double x1 = x2 =  ", x1)
                        sinon
10)                     x1 ← (-b –sqrt(delta))/(2*a)
11)                     x2 ← (-b +sqrt(delta))/(2*a)
12)                     écrire ("L'équation admet deux solutions x1 = ", x1, "et x2 =  ", x2)
                      fsi
            fsi
    fsi
fin
```

مناقشة الحل:

إذا كانت المتغيرة a = 0 في هذه الحالة يمكن وضع الخوارزمية السابقة (الخوارزمية الخاصة بحلول معادلة من الدرجة الأولى) في المكان الموافق لهاته الحالة مع إبدال المتغيرات a وb بالمتغيرات b وc، بحيث تصبح الحلول الثلاثة: c/b– ، المجموعة الخالية والمجموعة اللامنتهية.

أما إذا كانت المتغيرة a <> 0 نمر مباشرة لحساب delta، ثم ندرس الحالات الثلاثة المعروفة: delta < 0، delta = 0 و delta > 0.

عملية المحاكاة لكل الحالات الممكنة:

في حالة a = 0 b = 2 وc= -4 المعادلة هي من الشكل $2x - 4 = 0$ التعليمة رقم (1) ستنفذ وتعرض على الشاشة أن المعادلة أصبحت من الدرجة الأولى، ثم تنفذ التعليمة (2) لأنها تدخل ضمن حالات b <> 0 وبذلك $x1 \leftarrow$ -4/2 - أي $x1 \leftarrow 2$ ثم يتم تنفيذ التعليمة رقم (3) وهي التي ستعرض على الشاشة التعليق والنتيجة، أما بقية الحالات المتعلقة بـ a = 0 هي نفسها كالخوارزمية السابقة التي تناولت حلول معادلة من الدرجة الأولى.

في حالة a = 1 وb=2 وc = 4 نقوم بحساب delta = 2 *2 - 4 * 1 * 4 = -12 من خلال تنفيذ التعليمة رقم (6) وبهذا فإن المميز جاء سالبا فنقوم بتنفيذ التعليمة رقم (7) والتي تعلمنا بعدم وجود حلول.

في حالة a = 1 وb=2 وc = 1 في هذه الحالة نقوم بحساب delta = 2 *2 - 4 * 1 * 1 = 0 من خلال تنفيذ التعليمة رقم (6) وبهذا فإن المميز جاء معدوما فنقوم بتنفيذ التعليمة رقم (8) والتي تقوم بحساب الحل المضاعف والذي يساوي في حالتنا هذه -1 = 1*2/2-، بعدها مباشرة ننفذ التعليمة رقم (9) والتي تقوم بعرض الحل والتعليق الخاص به.

في حالة a = 1 و b= 0 و c = -4 في هذه الحالة نقوم بحساب delta = 0 *0 - 4 * 1 * -4 = 16 من خلال تنفيذ التعليمة رقم (6) وبهذا فإن المميز جاء موجبا فنقوم بتنفيذ التعليمة رقم (10) والتعليمة رقم (11) والتي تقوم بحساب x1 وx2 بحيث x1 = -2 و x2 = 2، بعدها مباشرة ننفذ التعليمة رقم (12) والتي تقوم بعرض الحل والتعليق الخاص به.

نفس الملاحظة التي سقناها عندما تناولنا الخوارزمية الخاصة بحل معادلة من الدرجة الأولى تنطبق على حلول المعادلة من الدرجة الثانية، إذ يمكن أن نعطي على الأقل أربعة خوارزميات، ونكتفي في هذا المقام بإعطاء خوارزمية أخرى تعطي نفس النتائج وتختلف فقط في الكتابة وطريقة التفكير، بحيث نبدأ بدراسة حالة a <> 0 ثم نمر إلى دراسة حالة a = 0، وتتفرع حالة a <> 0 إلى ثلاث حالات جزئية حسب إشارة المميز (delta)، بينما تفرع حالة a = 0 إلى ثلاث حالات كما تم الإشارة إليه في حلول معادلة من الدرجة الأولى.

انطلاقا من المعطيات السابقة يمكن كتابة الخوارزمية الثانية على الشكل التالي:

```
Algorithme eq_prem_deg
var  a, b, c,  x1, x2, Delta : réel
début
   écrire ("Résolution d'une équation du premier degré du type ax² + bx + c = 0 ")
   écrire ("Veuillez entrer les valeur de a et b et c ")
   lire (a, b, c)
   si a <> 0
            delta ← b * b - 4 * a *c
            si delta  <  0
               alors
                  écrire ("L'équation n'admet aucune solution ")
               sinon si delta = 0
                        alors
                           x1 ← -b/(2*a)
                           écrire ("L'équation admet une solution double x1 = x2 =  ", x1)
                        sinon
                         x1 ← (-b –sqrt(delta))/(2*a)
                         x2 ← (-b +sqrt(delta))/(2*a)
                         écrire ("L'équation admet deux solutions x1 = ", x1, "et x2 =  ", x2)
                      fsi
            fsi
      sinon
            écrire ("L'équation devient une équation du premier degré du type bx + c = 0 ")
            si b <> 0
               alors
                  x1 ← -c/b
                  écrire ("L'équation admet une solution double x1 = x2 =  ", x1)
               sinon   si c <> 0
                           alors
                              écrire ("L'équation n'admet aucune solution ")
                           sinon
                              écrire ("L'équation admet une infinité de solutions ")
                        fsi
            fsi
   fsi
fin
```

حل التمرين السادس:

```
Algorithme paire_impaire
var  nbr : entier
début
    écrire ("Veuillez saisir un entier SVP")
     lire (nbr)
    si nbr mod 2 = 0
      alors
        | écrire ("Le nombre saisi est pair")
      sinon
         écrire ("Le nombre saisi est impair")
    fsi
fin
```

مناقشة الحل:

اعتمدنا في هذه الخوارزمية على العامل mod والذي يحسب باقي القسمة الصحيحة لعدد صحيح على عدد صحيح آخر، ففي حال كان الباقي معدوما فإن العدد الأول هو من مضاعفات العدد الثاني، وفي حالتنا هذه العدد الزوجي هو كل عدد مضاعف لـ 2، لذلك فإن أي عدد باقي قسمته على 2 يساوي 0 هو مضاعف لـ 2 وهو بالتالي زوجي، أما العدد الذي باقي قسمته على 2 لا يساوي 0 هو عدد فردي، وهو ما تم كتابته في الخوارزمية السابقة.

حل التمرين السابع:

```
Algorithme Min_1
var  a, b, c, Min : entier
début
    écrire ("Veuillez saisir trois entiers SVP")
     lire (a, b, c)
    Min ← a
    si Min > b
      alors
          Min ← b
    fsi
    si Min > c
       alors
          Min ← c
    fsi
    écrire ("Le minimum de ces trois entiers est :", Min)
fin

Algorithme Min_2
var  a, b, c : entier
début
    écrire ("Veuillez saisir trois entiers SVP")
     lire (a, b, c)
    si a < b ET a < c
      alors
```

```
        écrire ("Le minimum de ces trois entiers est :", a)
    fsi
     si b < a ET b < c
       alors
         écrire ("Le minimum de ces trois entiers est :", b)
    fsi
    si c < a ET c < b
      alors
        écrire ("Le minimum de ces trois entiers est :", c)
    fsi
fin
```

مناقشة الحل:

في الخوارزمية الأولى استعملنا متغيرة إضافية Min بحيث نعتبر a هو أصغر قيمة ونضعه في المتغيرة Min، وفي حال وجود قيمة أخرى أصغر من المتغير Min نقوم بتغيير Min ونضع فيه تلك القيمة، أما في الخوارزمية الثانية وفي غياب المتغيرة الإضافية Min قمنا مباشرة بعملية مقارنة بين المتغيرات الثلاثة.

لكتابة خوارزمية تبحث عن أكبر قيمة يكفي تغيير الإشارة، في الخوارزمية الأولى تغيير رمز أكبر برمز أصغر، وفي الخوارزمية الثانية تغيير إشارة أصغر بإشارة أكبر.

حل التمرين الثامن:

```
Algorithme Ecart
var  Ecart, Dep, Rec : réel
début
      écrire ("Veuillez saisir les recettes de l'entreprise")
       lire (Rec)
      écrire ("Veuillez saisir les dépenses de l'entreprise")
       lire (Dep)
       Ecart ← Rec – Dep
       si Ecart > 0
         alors
                  écrire ("Entreprise bénéficiaire")
         sinon si Ecart < 0
                  alors
                     écrire ("Entreprise déficitaire ")
                  sinon
                     écrire ("Entreprise équilibrée ")
               fsi
      fsi
fin
```

مناقشة الحل:

في هذه الخوارزمية استعملنا التعليمة الشرطية المتداخلة في أبسط أشكالها، فهي تشبه كثيرا المثال التطبيقي الذي رأيناه عندما تطرقنا لهذه التعليمة لأول مرة، والملاحظ في هذه الخوارزمية أننا نستطيع التخلي عن المتغيرة Ecart ونستعمل

المقارنة المباشرة بين المتغيرين Dep وRec بحيث عندما يكون Rec > Dep فإن المؤسسة رابحة وعندما Rec < Dep فالمؤسسة خاسرة وعندما Rec = Dep فالمؤسسة متوازنة.

حل التمرين التاسع:

```
Algorithme Tri
var  a, b, c : entier
début
      écrire ("Veuillez saisir trois entiers")
       lire (a, b, c)
      si a < = b
        alors
            si b < = c
              alors
                écrire (a, " ", b, " ", c)
              sinon si a < =c
                        alors
                          écrire (a, " ", c, " ", b)
                       sinon
                          écrire (c, " ", a, " ", b)
                    fsi
          fsi
        sinon si a <= c
                alors
                   écrire (b, " ", a, " ", c)
                sinon si b <= c
                          alors
                              écrire (b, " ", c, " ", a)
                           sinon
                              écrire (c, " ", b, " ", a)
                        fsi
               fsi
  fsi
fin
```

مناقشة الحل:

لترتيب ثلاثة أعداد صحيحة a, b, c ترتيبا تصاعديا لدينا الحالات الستة التالية:

1) a <= b <= c فإن الترتيب يكون a, b, c
2) a <= c <= b فإن الترتيب يكون a, c, b
3) b <= a <= c فإن الترتيب يكون b, a, c
4) b <= c <= a فإن الترتيب يكون b, c, a
5) c <= a <= b فإن الترتيب يكون c, a, b
6) c <= b <= a فإن الترتيب يكون c, b, a

وهي نفسها الحالات التي تم التوصل إليها من خلال الخوارزمية السابقة والتي جاءت حسب ترتيب الحالات التالية: (1 ثم 2) ثم (5 ثم (3 ثم (4 ثم (6.

حل التمرين العاشر:

```
Algorithme prime
var  Nb-enf, Prime : entier
début
      écrire ("Veuillez donner le nombre d'enfant")
       lire (Nb_enf)
      si Nb_enf  <  0
        alors
            écrire( " Saisie erronée")
        sinon si Nb_enf <= 3
                  alors
                        Prime ← Nb_enf * 600
                  sinon si Nb_enf <= 6
                            alors
                                Prime ← Nb_enf * 500
                            sinon
                                 Prime ← Nb_enf * 450
                        fsi
              fsi
    fsi
    si Prime > 3600
       alors
              Prime ← 3600
   fsi
   écrire (" Le montant de la prime est :", Prime)
fin
```

مناقشة الحل:

في هذه الخوارزمية استعملنا أربعة تعليمات شرطية، ثلاثة منها متداخلة وواحدة بسيطة، بحيث كان الشرط الأول من التعليمة الشرطية المتداخلة يتعلق بإدخال عدد أولاد Nb_enf موجب أو معدوم وفي حالة ما إذا كان غير ذلك أي Nb_enf < 0 فإن الخوارزمية تعرض على الشاشة تعليق يبين للمستخدم (وهو في هذه الحالة المحاسب) أن القيمة المدخلة خاطئة، أما في حالة نفي الشرط السابق ومعناه Nb_enf >= 0 فأن القيمة المدخلة مقبولة، ولكن يجب دراسة الحالات الثلاثة كما هو مبين في المطلوب:

- Nb_enf >= 0 و Nb_enf <= 3 ؛
- Nb_enf > 3 و Nb_enf <= 6 ؛
- Nb_enf > 6؛

الملاحظ في الخوارزمية السابقة وعند كتابة الشروط الثلاثة، كنا نكتب الجزء الثاني فقط من الشرط لأن الجزء الأول يدخل آليا، بحيث عندما نكتب عبارة sinon للشرط Nb_enf < 0 فإن هذا يعني Nb_enf >= 0 لذا فلا داعي لإعادة كتابته بل يكفي كتابة Nb_enf <= 3 ، وعند كتابة عبارة sinon المتعلقة بالشرط Nb_enf <= 3 لا داعي لكتابة Nb_enf > 3 بل يكفي كتابة Nb_enf <= 6، أما عبارة sinon الأخيرة فيدخل في مجالها Nb_enf > 6.

التعليمة الشرطية البسيطة الأخيرة تمت كتابتها لنضمن أن لا تتعدى قيمة المنحة 3600 دج بحيث إذا كان عدد الأولاد 7 أولاد فإن الموظف يتقاضى منحة قدرها 3150 دج أما إذا كان عدد أولاد 8 أولاد فإن الموظف يتقاضى منحة قدرها

3600 دج وهي أقصى قيمة يتقاضاها، بحيث مهما كان عدد الأولاد أكبر من 8 فإن الموظف يتقاضى منحة قدرها 3600 دج.

حل التمرين الحادي عشر:

```
Algorithme Net_payer
    const TTVA = 19
    var Fr_trans, MTVA, MHT, Net_P, M_rem, Qte, PU : réel
        distance : entier
début
    écrire ("Veuillez donner le prix unitaire du produit")
    lire (PU)
    écrire ("Veuillez donner la quantité du produit")
    lire (Qte)
    écrire ("Veuillez donner la distance parcourue par le produit")
    lire (distance)
    MHT ← Qte * PU
    Fr_trans ← distance * 5000/100
    si MHT > 200000
        alors
            M_rem ← MHT * 0.1
        sinon si MHT > = 100000
                alors
                    M_rem ← MHT * 0.05
                sinon si MHT > = 50000
                        alors
                            M_rem ← MHT * 0.02
                        fsi
            fsi
    fsi
    MTVA ← (MHT – M_rem + Fr_trans) * TTVA/100
    Net_P ← MHT – M_rem + Fr_trans + MTVA
    écrire ("Le montant Net de la marchandise est :", Net_P)
fin
```

مناقشة الحل:

استعملنا في هذه الخوارزمية نفس الطريقة المستعملة في الخوارزمية السابقة، وهذا عند حساب قيمة التخفيض لكي لا يتم كتابة الشروط الداخلة آليا، كما يمكن أن نلاحظ أن قيمة البضاعة بعد التخفيض يمكن حسابها مباشرة دون حساب قيمة التخفيض، بحيث عوض أن نكتب:

```
M_rem ← MHT * 0.1
MHT ← MHT – M_rem
```

نكتب مباشرة: M_HT ← MHT * 0.9

وهذا يكافئ:

```
M_rem ← MHT * 0.1
MHT ← MHT – M_rem
```

الفصل الثالث

الفصل الثالث: التعليمات الأساسية في الخوارزميات– التعليمات التكرارية

تمهيد:

نواجه في حياتنا اليومية العديد من المسائل التي تحل عن طريق تكرار نفس العمليات وتختلف فقط في المعطيات، فعند حساب معدلات طلبة كليتنا سنكون مدعوين لاستعمال نفس العمليات الحسابية، لكن نقاط الطلبة تختلف من طالب لآخر، وبالتالي فمعدلاتهم ستكون حتما مختلفة.

وعلى سبيل المثال مجموع الأعداد من 1 إلى 100، أو بصفة عامة من 1 إلى N بحيث N عدد طبيعي، يمكن كتابته في الخوارزميات على الشكل التالي:

$S \leftarrow 0$
$S \leftarrow S + 1$
$S \leftarrow S + 2$
$S \leftarrow S + 3$
$\vdots$
$S \leftarrow S + N$

أو على الشكل التالي: $S \leftarrow 1 + 2 + 3 + 4 + \ldots + N$

إذا كان N = 10 000 يجب كتابة ($S \leftarrow S + i$) 10 000 مرة بحيث في كل مرة نغير من قيمة i من 1 إلى 10 000.

في الكتابة السابقة ($S \leftarrow S + i$) نلاحظ أن i يتغير في كل مرة ولكن صيغة العملية الحسابية لا تتغير، وبهذا نقوم بتكرار نفس العملية الحسابية ونغير فقط في المعطيات، ومن هنا ظهرت الحاجة الماسة إلى استخدام نوع جديد من التعليمات الواسعة الانتشار في عالم الخوارزميات والبرمجة، ألا وهي التعليمات التكرارية.

وتنقسم التعليمات التكرارية إلى قسمين رئيسين هما:

- التعليمات التكرارية بعداد؛
- والتعليمات التكرارية بشرط الخروج.

وهما القسمان اللذان سنتطرق إليهما بشيء من التفصيل فيما يلي:

I– التعليمة التكرارية بعداد أو الحلقة بعداد (التعليمة pour / for)

تستعمل هذه التعليمة أو الحلقة (boucle) في الخوارزميات وفي البرمجة عادة، عندما يكون عدد التكرار معروف مبدئيا، ولا يتم الخروج من الحلقة حتى الوصول إلى نهاية التكرار،* بحيث نستطيع معرفة المرات التي ستنفذ فيها هاته التعليمة التكرارية.

*هناك بعض لغات البرمجة (مثال عن ذلك لغة C++) من يسمح بالخروج من الحلقة pour قبل نهاية التكرار باستعمال تعليمة goto أو تعليمة break.

1.I طريقة كتابة الحلقة pour:

تكتب الحلقة pour على الشكل العام التالي:

```
pour cmpt ← Val_initiale à Val_finale pas Val_pas
faire
   Instruction 1
   Instruction 2
        :
     Instruction n
fin pour
```

بحيث cmpt يعبر عن العداد وهو بطبيعة الحال يجب أن يكون عددا طبيعيا، Val_initiale تمثل القيمة الابتدائية للعداد وهي نقطة انطلاقة الحلقة أما Val_finale فتمثل القيمة النهائية للعداد وهي نقطة الوصول، أما Val_pas فتمثل قيمة خطوة الانتقال وهو المقدار الذي تنتقل فيه الحلقة من قيمة إلى أخرى.

2.I طريقة عمل الحلقة pour:

تقوم هذه الحلقة بتنفيذ كل التعليمات التي تقع داخلها، أي بين الكلمتين faire وfin pour، أي (Instruction 1, Instruction 2,…., Instruction n) بشرط أن يكون العداد داخل المجال [Val_initiale , Val_finale]، عند وصول الحلقة إلى كلمة fin pour تقوم بزيادة قيمة العداد بمقدار خطوة (Val_pas) ثم قبل معاودة الدخول إلى الحلقة تقوم بالتأكد من أن قيمة العداد الجديدة لا زالت داخل المجال السابق، وفي حالة خروج قيمة العداد من المجال فإن الخوارزمية تكون قد انتهت من التعليمة التكرارية pour وتمر لتنفيذ التعليمات الأخرى التي تلي تعليمة Pour.

ملاحظات هامة:

- عندما تكون قيمة الخطوة تساوي واحد لا داعي لكتابتها، بحيث في حالة عدم كتابة الخطوة فإن التعليمة pour ستعتبرها آليا مساوية لواحد.
- عند استعمال عداد الحلقة pour لا ينبغي تغيير قيمته داخل الحلقة، على سبيل المثال لا نستطيع كتابة الخوارزمية التالية:

```
Algorithme erreur
var cmpt, somme : entier
début
   somme ← 0
  pour cmpt ← 1 à 100
      faire
        somme ← somme + cmpt
        cmpt ← cmpt -1
     fin pour
   écrire (somme)
fin
```

الملاحظ في الخوارزمية السابقة أن قيمة العداد cmpt تم تغييرها داخل الحلقة وهو ما يؤثر على السير الحسن لهذه الحلقة، بحيث قيمة العداد لا تتقدم وبالتالي الحلقة تصبح حلقة لانهائية، والبرنامج الناتج عن هذه الخوارزمية لا يمكن أن ينفذ، ويؤدي إلى توقف نظام التشغيل بحيث يصبح لزاما على مستخدم هذا البرنامج أن يوقفه عنوة عبر الكبس المتلازم على المفاتيح (Ctrl + Alt + Suppr, puis forcer l'arrêt).

- عند استعمال الحلقة pour لا نستطيع الخروج منها إلا إذا أكملنا كل التعداد، أي لا يمكن الخروج منها حتى يصل العداد إلى القيمة النهائية الخاصة به (Val_finale).

مثال تطبيقي:

أكتب خوارزمية تقوم بجمع الأعداد الطبيعية من 1 إلى 100.

```
Algorithme Somme
var cmpt, somme : entier
début
   somme ← 0
  pour cmpt ← 1 à 100
      faire
         somme ← somme + cmpt
     fin pour
   écrire (somme)
fin
```

مناقشة الحل:

تقوم الخوارزمية السابقة بجمع الأعداد من 1 إلى 100 أي: somme = 1 + 2 + 3 + 4 + ……+ 100

والتي يمكن حسابها عبر 100 خطوة بالطريقة التالية:

أولا يجب إعطاء القيمة الابتدائية للمتغيرة somme بحيث نعطيها القيمة 0 وهذا باعتبار 0 العنصر الحيادي لعملية الجمع، وكذلك لمسح أي قيمة تكون في المتغيرة somme، بحيث عند التصريح بأي متغيرة جديدة، تأخذ هذه الأخيرة قيمة عشوائية من خلال موقعها في الذاكرة المركزية.

الخطوة الأولى:

somme ← somme + 1

وبهذا تصبح المتغيرة somme تحتوي على القيمة 1.

الخطوة الثانية:

somme ← somme + 2

وبهذا تصبح المتغيرة somme تحتوي على القيمة 3 = 2 + 1 أي القيمة القديمة لـ somme + 2.

الخطوة الثالثة:

somme ← somme + 3

وبهذا تصبح المتغيرة somme تحتوي على القيمة 6 = 3 + 3 أي القيمة القديمة لـ somme + 3.

وهكذا حتى نصل إلى الخطوة 100 والتي تكون على الشكل التالي:

somme ← somme + 100

وبهذا تصبح المتغيرة somme تحتوي على القيمة 5050 = 100 + 4950 أي القيمة القديمة لـ somme + 100.

وبهذا نحصل على مجموع الأعداد من 1 إلى 100، ويمكن توضيح خطوات هذا الحساب عبر الكتابة التالية:

somme = 1 + 2 + 3 + ……+ 100

الخطوة الأولى

الخطوة الثانية

الخطوة الثالثة

الخطوة 100

الملاحظ في الخوارزمية السابقة أنها لم تستعمل الخطوة (val_pas) في التعليمة pour، وبالتالي فالخطوة تعتبر آليا 1، كما نستطيع جمع الأعداد من 1 إلى 100 من خلال عملية جمع من 100 إلى 1 بما أن الجمع تبديلي، وبذلك يمكن كتابة:

somme = 100 + 99 + ……+ 2 + 1

والخوارزمية التي تقوم بهذا الحساب ستكون كالآتي:

```
Algorithme Somme
var cmpt, somme : entier
début
   somme ← 0
  pour cmpt ← 100  à 1 pas -1
      faire
        somme ← somme + cmpt
     fin pour
   écrire (somme)
fin
```

الملاحظ في الخوارزمية السابقة أنها استعملت القيمة -1 للخطوة (val_pas)، وبالتالي بعد كل تنفيذ للتعليمة pour نقوم بتخفيض قيمة العداد بـ 1.

وكمثال ثاني عن استعمال قيمة الخطوة، نريد كتابة خوارزمية تقوم بحساب مجموع الأعداد الزوجية من 2 إلى 100، والتي تكتب على الشكل التالي:

```
Algorithme Somme_paire
var cmpt, somme_p : entier
début
   somme _p← 0
  pour cmpt ← 2  à 100  pas 2
      faire
        somme_p ← somme_p + cmpt
     fin pour
   écrire (somme_p)
fin
```

الخوارزمية السابقة تحسب المجموع التالي: somme_p = 2 + 4 + ….+ 100

II- التعليمة التكرارية بشرط الخروج

قد نصادف في العديد من الحالات، مسائل تستوجب التكرار، لكن لا يمكن معرفة عدد هذا التكرار مبدئيا، وبالتالي لا نستطيع استعمال التعليمة Pour، والتي يرتبط استعمالها مع الحالات التي يكون عدد التكرار فيها معروف، من هنا جاءت الحاجة الماسة إلى نوع آخر من الحلقات أو التعليمات التكرارية والتي تستعمل في الحالات التي يكون عدد التكرار غير معروف مبدئيا، مع إمكانية إضافة شرط للخروج من الحلقة.

1.II التعليمة التكرارية (الحلقة) tant que:

تستعمل هذه التعليمة (الحلقة) عندما يكون عدد التكرار غير معروف مبدئيا، كما يمكن أن لا ينفذ التكرار ولا مرة، ويرتبط استعمال هذه الحلقة مع شرط الخروج والذي قد يكون شرطا باستعمال متغير منطقي أو مقارنة.

1.1.II طريقة كتابة الحلقة tant que:

تكتب الحلقة tant que على الشكل العام التالي:

```
tant que  condition
faire
   Instruction 1
   Instruction 2
        ⋮
   Instruction n
fin tant que
```

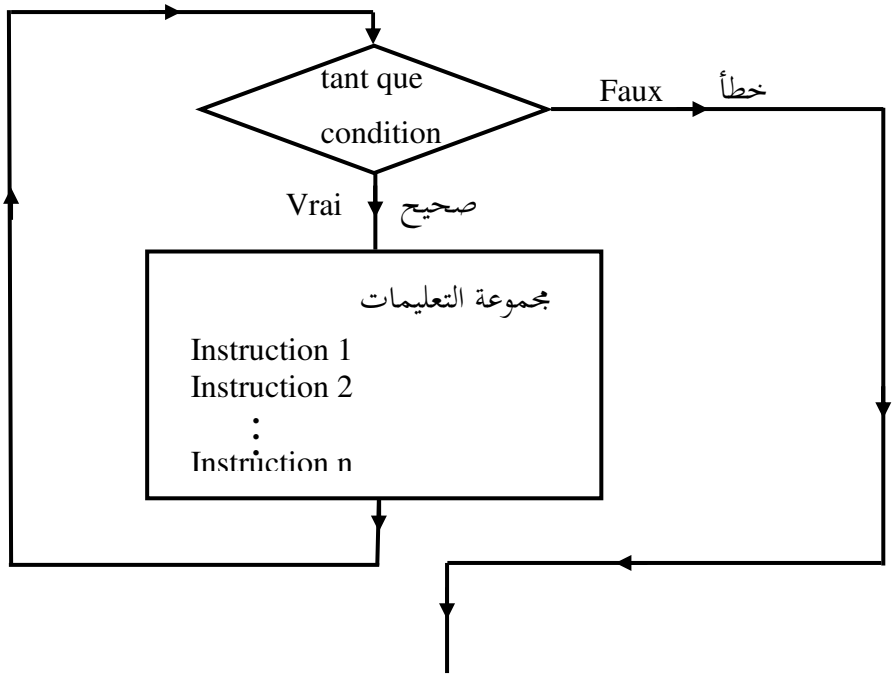
بحيث يعتبر الشرط condition والذي يأتي مباشرة بعد كلمة tant que هو مفتاح الدخول إلى الحلقة وفي حالة تحققه يتم الدخول إلى الحلقة، أما في حالة العكس لا يتم الدخول إلى الحلقة.

2.1.II طريقة عمل الحلقة tant que:

يشبه عمل هذه الحلقة كثيرا عمل التعليمة الشرطية البسيطة، ويكمن الاختلاف في إمكانية تكرار تنفيذ هذه التعليمة (الحلقة)، بحيث عند تحقق الشرط (condition) فإنه يتم الدخول إلى متن الحلقة ويتم تنفيذ كل التعليمات التي تقع بين faire وfin tant que، أما في حالة عدم تحقق الشرط، يتم الخروج مباشرة من الحلقة ومواصلة الخوارزمية بعد كلمة fin tant que.

ولتوضيح عمل التعليمة التكرارية tant que بشكل بياني مبسط ارتأينا إدراج الشكل التالي:

الشكل (1): شكل توضيحي لطريقة عمل التعليمة التكرارية tant que



من خلال الشكل أعلاه يمكن أن نلاحظ أن عند وصول الخوارزمية إلى نهاية الحلقة أي عند الانتهاء من آخر تعليمة داخل الحلقة (Instruction n) وبمجرد الوصول إلى كلمة fin tant que فإن الخوارزمية تعاود مراقبة الشرط أي تقوم بالتأكد من استمرارية تحقق الشرط، وبذلك يكون لنا احتمالين إما الدخول مجددا إلى الحلقة أو إنهاء الحلقة ومتابعة سير الخوارزمية بعد نهاية الحلقة (fin tant que).

مثال توضيحي:

نريد أن نكتب خوارزمية تعلم المستخدم عن نتيجة الطالب (ناجح أو راسب) وهذا بعد إدخال معدله، يجب على الخوارزمية إرغام المستخدم على إدخال معدل صحيح ومقبول (معدل يكون ضمن المجال [20 , 0])، في حالة إدخال معدل غير مقبول يطلب من المستخدم إدخال معدل آخر صحيح.

```
Algorithme Resultat_Etudiant
var moyenne : réel
début
  écrire ("Donnez la moyenne de l'étudiant(e) cette moyenne doit être comprise entre 0 et 20")
1)   lire(moyenne)
   tant que moyenne < 0 OU moyenne > 20
     faire
        écrire ("La moyenne saisie est erronée, donnez une moyenne comprise entre 0 et 20")
2)      lire(moyenne)
     fin tant que
  si moyenne < 10
     alors
        écrire (" L'étudiant(e)  est ajourné(e)")
     sinon
        écrire (" L'étudiant(e)  est admis(e)")
fin
```

مناقشة الحل:

تم استعمال الحلقة tant que في الخوارزمية السابقة لإرغام المستخدم على إدخال معدل صحيح ومقبول، بحيث: حتى يكون المعدل مقبولا يجب أن لا يقل عن الصفر ولا يزيد عن العشرين، والملاحظ عند استعمال الحلقة tant que أنها يمكن أن لا تنفذ ولا مرة، ففي المثال السابق عند إدخال المعدل في التعليمة رقم (1 والتي جاءت قبل الحلقة يمكن لهذا المعدل أن يكون مقبولا، أي أصغر أو يساوي 20 وأكبر أو يساوي 0 (0 =< moyenne ET moyenne <=20)، وبهذا لن تنفذ التعليمة tant que إطلاقا، أما في حالة إدخال معدل خاطئ وهو يتمثل في نفي القضية (moyenne >=0 ET moyenne <= 20)، أي (moyenne < 0 OU moyenne > 20) ففي هذه الحالة يتم الدخول إلى الحلقة وتنفيذ التعليمة (2 ولن يتم الخروج منها إلا عبر إدخال معدل صحيح يجعل من الشرط (moyenne < 0 OU moyenne > 20) غير محقق.

ملاحظات هامة:

- يمكن تحويل أي حلقة pour إلى الحلقة tant que لكن العكس غير صحيح، وكمثال عن ذلك، نأخذ الخوارزمية التي تقوم بحساب مجموع الأعداد الزوجية من 2 إلى 100.

```
Algorithme somme_p
    var cmpt, somme_p : entier
    début
      somme_p ← 0
      cmpt ← 2
     tant que cmpt  <= 100
        faire
           somme_p ← somme_p + cmpt
           cmpt ← cmpt + 2
     fin tant que
fin
```

الملاحظ في الخوارزمية أعلاه أنها تؤدي نفس دور الخوارزمية التي استعملت الحلقة pour، إلا أن الفرق بينهما يكمن في قيام الحلقة pour بإعطاء القيمة الابتدائية (initialisation) أو قيمة بداية مجال الحساب، مراقبة قيمة العداد (contrôle) بحيث لا يجب أن يخرج عن مجال الحساب، وزيادة قيمة العداد (incrémentation) بطريقة آلية وضمنية (implicite)، ولكن عند استعمال الحلقة tant que لا بد من القيام بكتابة هذه المهام الثلاثة، والحرص على أن تكتب بطريقة صحيحة، بحيث عند نسيان واحدة منها أو كتابتها بطريقة خاطئة، يمكن أن يحدث خلل في عمل الخوارزمية، لذا فمن الأحسن في حالة المفاضلة بين الحلقة pour والحلقة tant que أن نختار استعمال الحلقة pour لسهولة كتابتها وبالتالي التقليل من احتمال الخطأ في كتابة المهام الثلاثة السالفة الذكر، ففي الخوارزمية أعلاه التعليمات بالخط العريض المسطر تمثل المهام الثلاثة التي يجب العناية بكتابتها كتابة منطقية صحيحة.

- عند كتابة الحلقة tant que يجب أن يتضمن متنها على تعليمة تمكن من تغير حالة الشرط الذي في بدايتها، وهو ما يمكننا من الخروج من هذه الحلقة، ومن هنا جاءت تسمية التعليمات التكرارية بشرط الخروج، بحيث في حالة عدم وجود تعليمة تغير من حالة الشرط تصبح الحلقة tant que حلقة لا نهائية، فعلى سبيل المثال الخوارزمية التالية تحتوي حلقة tant que لا نهائية.

```
Algorithme boucle_infinie
var i, j,  somme : entier
début
  somme ← 0
   i ← 0
   j ← 1
 tant que j  <= 5
    faire
       somme ← somme  +  i
       i ← i + 2
       j ← j - 1
    fin tant que
fin
```

الملاحظ في الخوارزمية السابقة أنها استعملت حلقة tant que ولكن لم تحسن استعمالها، بحيث أخلت بأحد المهام الثلاثة التي ذكرناها آنفا، ألا وهي زيادة قيمة العداد (incrémentation)، مما أدى إلا حدوث خلل في عمل هذه الحلقة وأصبحت حلقة لا نهائية.

- يمكن استعمال الحلقة tant que مع شرط متعلق بمتغير منطقي (باستعمال متغير من النوع المنطقي) عوض الشروط التي تستعمل المقارنات، وفي هذه الحالة يجب أن يحتوي متن الحلقة على تعليمة تغير من قيمة المتغير المنطقي، وكمثال عن هذا الخوارزمية التي تستعمل في إدارة كلمات المرور لمختلف التطبيقات.

```
Algorithme pass_word
var i : entier
    code, code_saisi : caractère
    trouve : booléen
début
   i ← 1
   trouve ← faux
   code ← "pass_w"
   tant que i  <= 3 ET trouve = faux
    faire
     écrire ("Veuillez saisir votre mot de passe SVP !")
      lire (code_saisi)
       si code_saisi = code
         alors
               écrire ("mot de passe juste")
               trouve ← vrai
        sinon
              écrire ("mot de passe erronée, il vous reste," 3-i, "chances ")
      fsi
        i ← i + 1
    fin tant que
fin
```

الملاحظ في الخوارزمية أعلاه أنها استعملت حلقة tant que مع شرط الخروج والمتمثل في مقارنة عددين طبيعيين (i <=3) ومقارنة متغير منطقي (trouve = faux)، بحيث للخروج من الحلقة يجب تحقق نفي الشرط (i <= 3 ET trouve = faux) أي (i > 3 OU trouve = vrai)، وبهذا يتم الخروج من الحلقة بأحد الحالات الثلاثة التالية:

- إما يتم إدخال كلمة المرور الصحيحة وعدد المحاولات أصغر تماما من ثلاثة، وبهذا يصبح المتغير المنطقي يحتوي على القيمة vrai والمتغير الطبيعي يحتوي على قيمة أصغر تماما من 3.
- إما لا يتم إدخال كلمة المرور الصحيحة بحيث المتغير المنطقي يظل يحتوي على القيمة faux، ولكن عدد المحاولات يتعدى 3، أي i >3.
- وإما يتم إدخال كلمة المرور الصحيحة وعدد المحاولات يساوي ثلاثة، وبهذا يصبح المتغير المنطقي يحتوي على القيمة vrai والمتغير الطبيعي يحتوي على قيمة أكبر تماما من 3.

II.2 التعليمة التكرارية (الحلقة) Répéter jusqu'à:

تستعمل هذه الحلقة عندما يكون عدد التكرار غير معروف مبدئيا، ولكن يجب أن ينفذ على الأقل مرة واحدة، ويرتبط استعمال هذه الحلقة كسابقتها مع شرط الخروج والذي قد يكون شرطا منطقيا أو مقارنة.

II.2.1 طريقة كتابة الحلقة Répéter jusqu'à:

تكتب الحلقة Répéter jusqu'à على الشكل العام التالي:

```
Répéter
   Instruction 1
   Instruction 2
        ⋮
   Instruction n
Jusqu' à condition
```

بحيث يعتبر الشرط condition والذي يأتي مباشرة بعد كلمة Jusqu' à هو مخرج الحلقة بحيث في حالة تحققه يتم الخروج من الحلقة، على عكس الحلقة tant que، أما في حالة عدم تحقق الشرط نعاود الدخول إلى الحلقة.

II.2.2 طريقة عمل الحلقة Répéter jusqu'à:

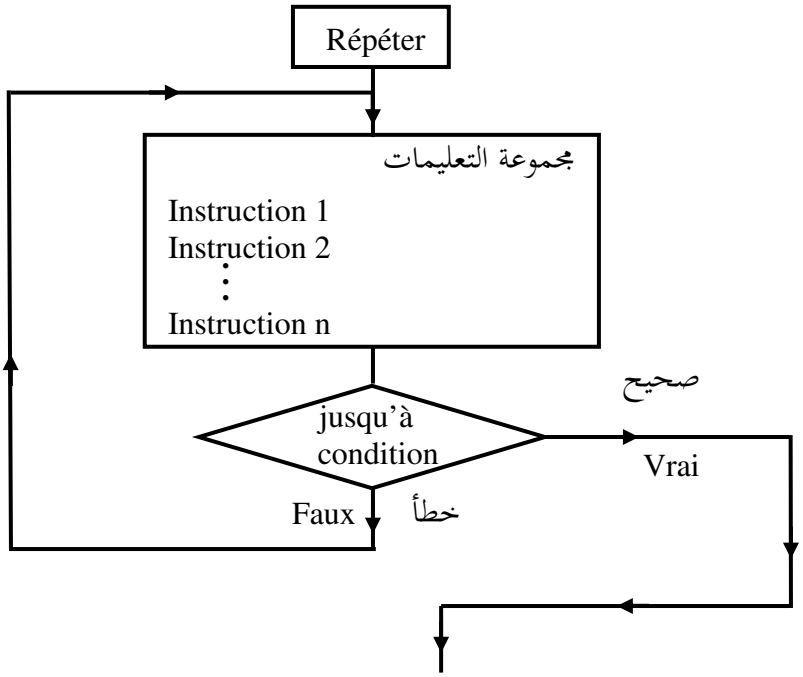
لا يختلف عمل هذه الحلقة كثيرا عن سابقتها الحلقة tant que والاختلافان الوحيدان بينهما هو وجود الشرط في بداية الحلقة tant que بينما يأتي الشرط في نهاية الحلقة Répéter jusqu'à، كما يشترط تحقق الشرط للدخول إلى الحلقة

tant que، بينما يشترط عدم تحقق الشرط للمواصلة داخل الحلقة Répéter jusqu'à وفي حالة تحققه يتم مغادرة الحلقة والخروج منها.

تجدر الإشارة إلى أنه في حالة عدم تحقق الشرط كل التعليمات التي تقع داخل الحلقة، أي بين كلمة Répéter وjusqu'à (Instruction 1, Instruction 2, …., Instruction n) سيتم تنفيذها الواحدة تلو الأخرى.

ولتوضيح عمل التعليمة التكرارية أو الحلقة Répéter jusqu'à بشكل بياني مبسط ارتأينا إدراج الشكل التالي:

الشكل (2): شكل توضيحي لطريقة عمل التعليمة التكرارية Répéter jusqu'à



من خلال الشكل أعلاه يمكن أن نلاحظ أن الحلقة répéter jusqu'à تقوم بتنفيذ التعليمات (Instruction 1, Instruction 2, …., Instruction n) قبل مراقبة الشرط، ولهذا فهي تنفذ على الأقل مرة واحدة. عند وصول الخوارزمية إلى كلمة jusqu'à فإن الخوارزمية تتأكد من صحة الشرط فإن كان صحيحا يتم الخروج من الحلقة.

مثال توضيحي:

نريد أن نكتب نفس الخوارزمية التي تعلم المستخدم عن نتيجة الطالب (ناجح أو راسب)، وهذا باستعمال الحلقة répéter jusqu'à عوضا عن التعليمة tant que، يجب على المستخدم إدخال معدل الطالب، وتتكفل الخوارزمية بإرغامه على إدخال معدل صحيح ومقبول (معدل يكون ضمن المجال [0 , 20])، وفي حالة إدخال معدل غير مقبول يطلب من المستخدم إعادة إدخال معدل آخر صحيح.

```
Algorithme Resultat_Etudiant
var moyenne : réel
début
  Répéter
  écrire ("Donnez la moyenne de l'étudiant(e) cette moyenne doit être comprise entre 0 et 20")
```

```
  lire(moyenne)
   jusqu'à  moyenne >= 0 ET  moyenne <= 20
  si moyenne < 10
    alors
       écrire (" L'étudiant(e)  est ajourné(e)")
    sinon
       écrire (" L'étudiant(e)  est admis(e)")
fin
```

الملاحظ في الخوارزمية أعلاه أنها جاءت أكثر بساطة وأقل تعقيدا من الخوارزمية التي استعملت التعليمة tant que بحيث احتوت هذه الخوارزمية على 13 سطرا بينما احتوت الخوارزمية الأخرى على 16 سطرا، ويرجع هذا الاختلاف في عدد الأسطر، إلى عدد التعليمات الإضافية التي حوتها التعليمة tant que مقابل عدد التعليمات في الحلقة répéter jusqu'à، بحيث عند استعمال التعليمة tant que قمنا بقراءة معدل الطالب قبل الدخول في التعليمة ثم قمنا بقراءته مرة أخرى داخل التعليمة، بينما لا يتم قراءة معدل الطالب إلى مرة واحدة داخل التعليمة répéter jusqu'à، كما جاء الشرط في التعليمة tant que عكس المطلوب، أي المعدلات التي لا تقبل فيطلب إعادة إدخال معدلات صحيحة (moyenne < 0 OU moyenne > 20)، أما الشرط في التعليمة répéter jusqu'à فجاء مطابقا للمطلوب بحيث يكون المعدل مقبولا إذا كان (moyenne >= 0 ET moyenne <= 20)، وهذا ما تم وضعه في شرط الخروج بحيث يطلب من المستخدم إعادة إدخال معدل يكون حسب الشرط السابق.

ملاحظات هامة:

- يمكن تحويل أي حلقة pour إلى الحلقة Répéter jusqu'à لكن العكس غير صحيح، وكمثال عن ذلك، نأخذ الخوارزمية التي تقوم بحساب مجموع الأعداد من 1 إلى 100.

```
Algorithme somme
var cmpt, somme : entier
début
  somme_p ← 0
  cmpt ← 1
 répéter
       somme_p ← somme_p + cmpt
       cmpt ← cmpt + 1
    jusqu'à cmpt  > 100
fin
```

الملاحظة نفسها التي سقناها في حالة الحلقة tant que تنطبق على الحلقة répéter jusqu'à، بحيث يجب الالتزام بإعطاء القيمة الابتدائية (initialisation)، مراقبة قيمة العداد (contrôle)، وزيادة قيمة العداد (incrémentation)، وهي التعليمات التي أدرجناها بالخط العريض المسطر.

- كل حلقة tant que يمكن تحويلها إلى حلقة répéter jusqu'à، بشرط التأكد من سلامة الشرط وصحته وطريقة كتابته المنطقية، بحيث يكون الشرط في الحلقة répéter jusqu'à عكسه تماما في الحلقة tant que.

وكمثال نأخذ الخوارزمية التي تدير كلمات المرور

```
Algorithme pass_word
var i : entier
    code, code_saisi : caractère
     trouve : booléen
début
   i ← 1
   trouve ← faux
   code ← "pass_w"
   répéter
      écrire ("Veuillez saisir votre mot de passe SVP !")
       lire (code_saisi)
        si code_saisi = code
          alors
                écrire ("mot de passe juste")
                trouve ← vrai
         sinon
               écrire ("mot de passe erronée, il vous reste," 3-i, "chances ")
       fsi
         i ← i + 1
        jusqu'à  i  > 3 OU  trouve = vrai
fin
```

الملاحظ في الخوارزمية أعلاه أنها استعملت عكس الشرط المستعمل في الحلقة tant que، بحيث (i > 3 OU trouve = vrai) يمثل نفي الشرط (i <= 3 ET trouve = faux).

II.3 كيفية الاختيار بين استعمال الحلقات الثلاثة (pour, tant que, répéter jusqu'à)

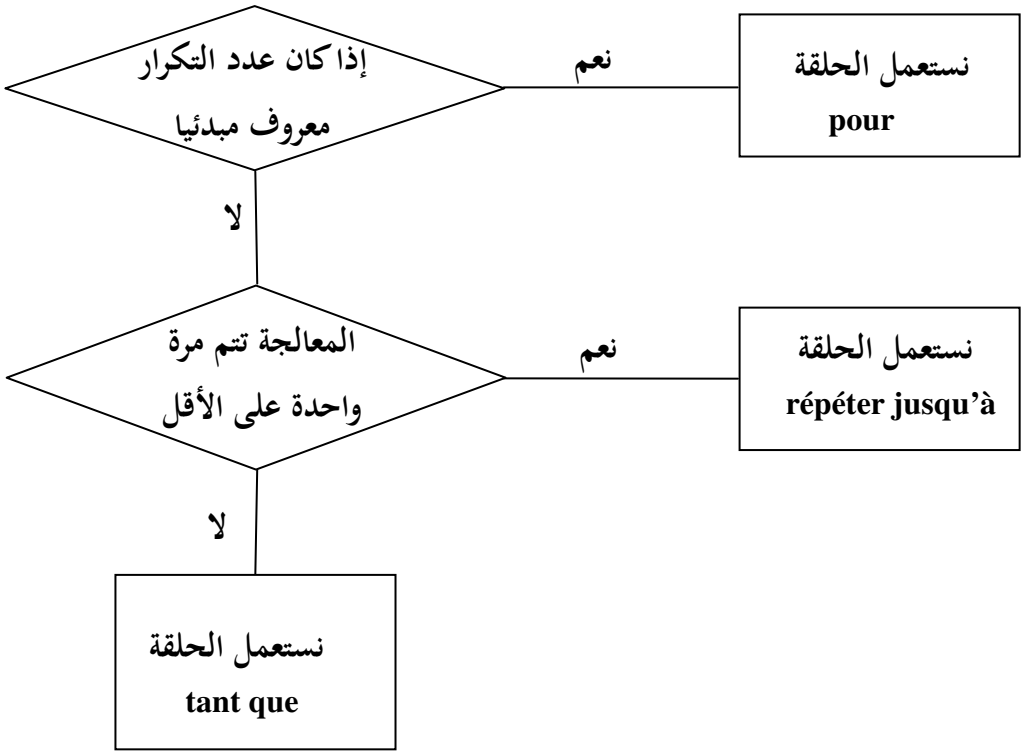
لقد سبق وأن نوهنا من خلال هذا الفصل أن استعمال الحلقات الثلاثة يتم حسب المعرفة المسبقة بعدد التكرار، مع أننا نستطيع دائما استعمال الحلقات tant que وrépéter jusqu'à حتى عندما يكون عدد التكرار معروفا مبدئيا، إلا أنه يفضل استعمال الحلقة pour في هذه الحالة.

والقاعدة العامة التي يستند عليها في الاختيار بين استعمال الحلقات الثلاثة هي:

- عندما يكون عدد التكرار معروف مبدئيا، نستعمل الحلقة pour؛
- عندما يكون عدد التكرار غير معروف مبدئيا نفاضل بين الحلقتين tant que وrépéter jusqu'à، لكن إذا كان عدد التكرار غير معروف مبدئيا ويمكن أن لا ينفذ أي مرة، في هذه الحالة نختار الحلقة tant que، أما إذا كان عدد التكرار غير معروف مبدئيا ويجب أن ينفذ مرة واحدة على الأقل نختار الحلقة répéter jusqu'à.

ولتوضيح كيفية الاختيار بين استعمال الحلقات الثلاثة حسب القاعدة السابقة ارتأينا إدراج الشكل التالي والذي يبينها بطريقة بيانية بسيطة:

الشكل (3): شكل توضيحي لكيفية الاختيار بين استعمال الحلقات الثلاثة



III. تمارين مقترحة:

التمرين الأول:

أكتب خوارزمية تقوم بحساب عملية ضرب عددين طبيعيين موجبين $a \times b$ باستعمال عملية الجمع.

مثال: $a \times b = \underbrace{a + a + \cdots + a}_{b\ fois}$

التمرين الثاني:

أكتب خوارزمية تقوم بحساب a^b باستعمال عملية الضرب علما أن a وb عددان صحيحان.

التمرين الثالث:

لتكن المتتالية الحسابية المعرفة كالآتي: $\begin{cases} U_0 = 1 \\ U_n = U_{n-1} + \frac{1}{2n} \end{cases}$

أكتب خوارزمية تقوم بحساب U_n من أجل $n > 0$ معطى.

التمرين الرابع:

أكتب خوارزمية تحسب الجذر التربيعي للعدد $x > 0$ باستعمال طريقة هيرون (méthode de héron) المعرفة كالتالي:

$$\begin{cases} U_0 = 1 \\ U_n = \frac{1}{2}(U_{n-1} + \frac{x}{U_{n-1}}) \end{cases}$$

قيمة الجذر التربيعي للعدد x تكون دقيقة عند تقارب قيمتي حدين متتاليين أي $|U_n - U_{n-1}| \leq 10^{-6}$

التمرين الخامس:

أكتب خوارزمية تقوم بحساب القيمة:

$$\prod_{i=a}^{b}(5a + b)$$

a و b عددين طبيعيين موجبين و $b > a$.

ماذا تمثل هذه القيمة؟

ملاحظة: $\prod_{i=1}^{n} x = \underbrace{x \times x \times \ldots \times x}_{n\ fois}$

و $\prod_{i=1}^{n} ix = \underbrace{x \times 2x \times 3x \times \ldots \times nx}_{n\ fois}$

التمرين السادس:

أكتب خوارزمية تحسب القيمة e^x $x > 0$ و ذلك باستعمال النشر المحدود التالي:

$$e^x = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots + \frac{x^n}{n!}$$

كلما كانت القيمة $\frac{x^n}{n!}$ صغيرة تكون النتيجة دقيقة (نأخذ مثال $\frac{x^n}{n!} \leq 10^{-3}$)

التمرين السابع:

أكتب خوارزمية تطلب من المستخدم إدخال مجموعة من الأعداد الطبيعية وتقوم بِعَدها وجمعها أولا بأول، بحيث يتوقف المستخدم عن إدخال هذه الأعداد عندما يتجاوز مجموعها 1000، يجب على الخوارزمية عند الانتهاء من إدخال الأعداد حساب معدلها ثم عرضه مع مجموع الأعداد على الشاشة.

التمرين الثامن:

أكتب خوارزمية تطلب من المستخدم إدخال 100 عدد طبيعي وتقوم بحساب عدد الأعداد الأصغر تماما من 50 من بين الـ 100 التي أدخلت، وتضعها في المتغيرة NB_50، ومجموعها وتضعه في المتغيرة S_50، وتقوم بحساب عدد الأعداد

الأكبر تماما من 100 وتضعها في المتغيرة NB_100 ومجموعها وتضعه في المتغيرة S_100، على الخوارزمية أن تعرض على الشاشة القيم NB_50، NB_100، S_50، S_100 بعد حسابها.

التمرين التاسع:

لدينا الخوارزمية التالية:

```
Algorithme  calcul9
 var a, b, d, m, s : entier
début
      lire(a, b)
      d← 0
      m← b
      s← 0
       tant que m <= a
            faire
                 s ← m
                 m ← m +b
                 d ←  d + 1
            fin tan que
          m ←  a - s
    ecrire (" d= ",d, " m= " ,m)
fin
```

نفذ محاكاة التنفيذ باليد (déroulement) للقيم التالية:

1) a =15 , b = 3
2) a= 34 , b =10

- ما هو دور هذه الخوارزمية ؟
- وهل توجد دوال تعطي لنا قيمة m, d بدون استعمال الحلقة ؟
- إذا كان الجواب نعم، أذكر هذه الدوال ثم عوضها داخل الخوارزمية.
- عوض الحلقة tant que داخل الخوارزمية بالحلقة répéter jusqu'à.

التمرين العاشر:

لتكن الخوارزمية التالية :

```
Algorithme calcul10
var A,B,R,S : entier
début
  lire(A,B)
  R ← 1
  tant que B <> 0
   faire
    S = B mod 2
    si S = 0
     alors
      A  ← A*A
      B  ← B div 2
```

```
        sinon
          R ← R*A
          B←B – 1
      fsi
    fin tant que
   écrire("le résultat est :",R)
fin
```

1) نفذ محاكاة التنفيذ باليد (déroulement) للقيم A و B التالية: (B=2 et A=5) (B=3 et A=10)

2) ما هو دور هذه الخوارزمية ؟

3) عوض الحلقة tant que الموجودة في الخوارزمية بالحلقة pour ثم بالحلقة répéter jusqu'à.

التمرين الحادي عشر:

لتكن الخوارزمية التالية :

```
Algorithme calcul11
var n,b,c : entier
début
  lire(n)
  si n < 0
   alors
     écrire("pas de solution")
   sinon si n = 0
            alors
              b ← 1
             écrire(b)
           sinon
            c ← 1
            b ← 1
            tant que c <= n
            faire
             b←b*c
             c ← c+1
            fin tant que
            écrire(b)
          fsi
    fsi
fin
```

1) نفذ محاكاة التنفيذ باليد (déroulement) لقيم n التالية : n=0, n=3 n=5

2) ما هو دور هذه الخوارزمية ؟

3) عوض الحلقة tant que الموجودة في الخوارزمية بالحلقة pour ثم بالحلقة répéter jusqu'à.

التمرين الثاني عشر:

لتكن الخوارزمية التالية:

```
Algorithme  calcul12
  var A, M, Result: entier
début
   lire(A)
   M ←A mod 10
   Result← M
   A ← A div 10
    Tant que  A <> 0
      M ← A mod 10
      Result ← Result + M
      A ← A div 10
   fin tant que
      écrire(Result)
 fin
```

1) أكتب محاكاة التنفيذ باليد (déroulement) للقيم التالية : (A = 3)، (A = 9879) ، (A = 17984)

2) ما هو دور هذه الخوارزمية ؟

3) عوض الحلقة tant que الموجودة في الخوارزمية بالحلقة répéter jusqu'à

التمرين الثالث عشر:

لتكن الخوارزمية التالية:

```
Algorithme  calcul13
  var A, B, C : entier
 Result: réel
début
   lire(A,B)
   C ←0
    si A = 0
      alors
          si B <= 0
            alors
               écrire("Pas de solutions")
            sinon
             Result  ← 0
             écrire(Result)
         fsi
      sinon
      Result ← 1
         si B < 0
            alors
              B ← -B
              C ← 1
        fsi
        tant que B  > 0
          faire
              Result ← Result * A
              B ←  B - 1
         fin tant que
```

```
        si C = 1
          alors
              Result ← 1/Result
        fsi
        écrire(Result)
    fsi
fin
```

1) أكتب محاكاة التنفيذ باليد (déroulement) للقيم التالية : (A =2 , B = -3) ، (A =0 , B = 5)،
(A =3 , B = 0) ، (A =4 , B = 2)
2) ما هو دور هذه الخوارزمية ؟
3) عوض الحلقة tant que الموجودة في الخوارزمية بالحلقة répéter jusqu'à

التمرين الرابع عشر:

اكتب خوارزمية تقوم بقراءة 10 أعداد صحيحة، ثم تقوم بإعلامنا ما إذا كانت هذه الأعداد متتالية أو لا.

ملاحظة: تتم عملية المراقبة أثناء القراءة بحيث في حالة عدم تحقق المتتالية لا نكمل قراءة الأعداد المتبقية.

التمرين الخامس عشر:

اكتب خوارزمية تقوم بقراءة 100 عدد طبيعي، وتقوم بإعلامنا بعدد الأعداد الزوجية Nb_P وعدد الأعداد الفردية Nb_I، ومعدل الأعداد الزوجية M_P وكذا معدل الأعداد الفردية M_I .

IV. حلول التمارين:

حل التمرين الأول:

```
Algorithme Exo1_V1
var  a, b, produit , i : entier
Début
    produit   ← 0
    répéter
    écrire( "Donnez la  valeur de a et b" )
    lire (a, b)
    jusqu'à a > = 0  ET b >= 0
     si a <> 0 ET b<> 0
     alors
          pour i ← 1 à b
            faire
               produit  ← produit + a
            fin pour
     fsi
          écrire ("le produit de a par b est : ", produit)
fin
```

مناقشة الحل:

لقد استعملنا في حل هذا التمرين الحلقة pour وهذا راجع لمعرفتنا المبدئية لعدد التكرار، بحيث a * b يكافئ (a + a + ... + a) b مرة أي أن عدد التكرار هو b.

لقد تم التطرق إلى حالتين في هذه الخوارزمية، الحالة الأولى وهي عندما يكون العددان الطبيعيان موجبين تماما، أما الحالة الثانية فهي عندما يكون أحد هذين العددين معدوم فجداؤهما يكون بالضرورة معدوم، هذه الحالة تدخل مباشرة في كتابة النتيجة ولا داعي لكتابتها بحيث يبقى الجداء معدوما كما تم إعطاؤه القيمة الأولية عند بداية الخوارزمية (produit ← 0)، كما تم استعمال الحلقة répéter jusqu'à في بداية الخوارزمية لإرغام المستخدم على إدخال عددين صحيحين موجبين.

يمكن استعمال الحلقة tant que أو الحلقة répéter jusqu'à عوض الحلقة pour مع توخي الحذر بعدم نسيان أي ركن من الأركان الثلاثة للحلقة tant que أو الحلقة répéter jusqu'à ، والخوارزمية التالية تستعمل الحلقة tant que بدل الحلقة pour.

```
Algorithme Exo1_V2
var  a, b, produit , i : entier
Début
    produit  ← 0
    répéter
    écrire( "Donnez la  valeur de a et b" )
    lire (a, b)
    jusqu'à a > = 0  ET b >= 0
     si a <> 0 ET b<> 0
      alors
          i ← 1
          tant que i <= b
            faire
                produit  ← produit + a
                i ← i + 1
            fin tant que
     fsi
          écrire ("le produit de a par b est : ", produit)
fin
```

لقد قمنا من خلال هذا التمرين بدراسة الحالات المتعلقة بجداء عددين طبيعيين أي عددين صحيحين موجبين، بحيث لو طلب منا جداء عددين صحيحين لجاءت الخوارزمية أكثر تعقيدا بحيث يجب دراسة كل الحالات وهي 9 حالات حسب إشارة a وb، بحيث كلا من العددين a و b يمكن أن يأخذ ثلاث حالات: سالبا، معدوما وموجبا تماما، وبهذا يكون عدد الحالات بالنسبة لـ a وb هو 3^2 ويساوي 9 وهي كالآتي:

1) a > 0 , b > 0

2) a > 0 , b = 0
3) a > 0 , b < 0
4) a = 0 , b > 0
5) a = 0 , b = 0
6) a = 0 , b < 0
7) a < 0 , b > 0
8) a < 0 , b = 0
9) a < 0 , b < 0

هذه الحالات التسعة يمكن اختصاراها في ثلاث حالات رئيسية هي:

- حالة الجداء المعدوم، وتدخل فيها كل الحالات التي فيها أحد العددين a أو b معدوم وهي الحالات (2، (4، (5، (6 و(8.
- حالة الجداء الموجب بحيث يكون فيها كلا العددين a وb موجبين أو سالبين وهي الحالتان (1 و(9.
- حالة الجداء السالب بحيث يكون فيها أحد العددين a أو b سالب، وهي الحالتان (3 و(7.

ويكمن كتابة الخوارزمية التي تدرس حالة جداء عددين صحيحين كالآتي:

```
Algorithme Exo1_V3
var  a, b, produit , i : entier
Début
    produit   ← 0
      écrire( "Donnez la  valeur de a et b" )
    lire (a, b)
     si ( a > 0 ET b > 0) OU (a < 0 ET b < 0)
       alors
           si a <  0
             alors
                a ← -a
          fsi
          si b <  0
            alors
                b ← -b
          fsi
          pour i ← 1 à b
            faire
               produit  ← produit + a
           fin pour
     fsi
     si ( a > 0 ET b < 0) OU (a < 0 ET b > 0)
       alors
           si a <  0
              alors
                 a ← -a
           fsi
           si b <  0
             alors
                b ← -b
           fsi
           pour i ← 1 à b
             faire
                produit  ← produit + a
```

```
                fin pour
            produit ← -produit
        fsi
        écrire ("le produit de a par b est : ", produit)
fin
```

حل التمرين الثاني:

```
Algorithme Exo2
var a, b, puissance , i : entier
Début
        écrire( "Donnez la  valeur de a et b" )
lire (a, b)
  puissance   ← 1
  si a <> 0
    alors
      si b > = 0
        alors
            pour i ← 1 à b
                faire
                    puissance  ← puissance * a
                fin pour
          sinon
            b ← -b
            pour i ← 1 à b
                faire
                    puissance  ← puissance * a
                fin pour
            puissance  ← 1/puissance
      sinon si a = 0 ET b > 0
            alors
                puissance   ← 0
          fsi
    écrire ("a puissance b est : ", puissance)
    fsi
    si a = 0 ET b  <= 0
     alors
       écrire ("la puissance est  non définie ")
    fsi
fin
```

مناقشة الحل:

لقد قامت الخوارزمية أعلاه باستعمال عملية الضرب من أجل الحصول على عملية الأس، بحيث أن a^b هو a * a *a..... * b مرة، مع دراسة كل الحالات الخاصة وحالات عدم التعيين، هاته الحالات جاءت كالآتي:

- a = 0 و b = 0 وهي حالة عدم تعيين 0^0.

- a = 0 و b < 0 وهي تعطينا $\frac{1}{0^{-b}}$ بحيث b– يصبح عدد صحيح موجب، و$\frac{1}{0^{-b}}$ يساوي 0^b، وهي حالة عدم التعيين أيضا.
- a = 0 و b > 0، في هذه الحالة الأس يساوي الصفر 0، $0^b = 0$ مهما كانت قيمة b.
- a <> 0 و b >= 0، في هذه الحالة الأس يساوي إما 1 في حالة b = 0 أو a*a* … * a b مرة، عندما يكون b أكبر تماما من الصفر، في حالة ما يكون b = 0 فإن قيمة المتغيرة puissance تبقى تساوي 1 ولا يتم الدخول إلى الحلقة pour، وبذلك لا تتغير قيمتها.
- a <> 0 و b < 0، في هذه الحالة الأس يصبح يساوي $\frac{1}{a^{-b}}$ ، لذا نغير من إشارة b بكتابة b ← -b، ونحسب a^{-b} بعد ذلك نحسب المقلوب $\frac{1}{a^{-b}}$ بكتابة 1/ puissance ← puissance وهو قيمة الأس في حالة b < 0.

حل التمرين الثالث:

```
Algorithme Exo 3
var  U0, U1 : réel
     n, i : entier
début
        écrire( "Donnez une  valeur de n" )
         répéter
         lire(n)
         jusqu'à n > 0
         U1   ← 1
          pour i ← 1 à n
                    faire
                          U0  ← U1
                           U1 ← U0 +1/2 *(1/i)
                    fin pour
           écrire ("La série Un est : ",U1)
 fin
```

مناقشة الحل:

بدأت هذه الخوارزمية بطلب إدخال قيمة n والتي يجب أن تكون عددا طبيعيا موجبا تماما، وبذلك تم استخدام التعليمة Répéter jusqu'à من أجل إرغام المستخدم على إدخال قيمة موجبة تماما، وذلك من خلال الجملة (jusqu'à n > 0) والتي من شأنها مراقبة القيمة المدخلة، بحيث لا يتم الخروج من الحلقة حتى يصبح الشرط صحيح و n موجبة تماما.

بعد إدخال قيمة n، نقوم بحساب المتتالية المعرفة كالآتي: $\begin{cases} U_0 = 1 \\ U_n = U_{n-1} + \frac{1}{2n} \end{cases}$ فنقوم بإعطاء قيمة حدها الأول U0 = 1 من خلال التعليمة U1 ← 1 ثم نمر إلى حساب الحد الأخير Un والذي يحسب باستخدام الحد ما قبل الأخير Un-1، وبما أننا بصدد حساب متتالية حسابية، فإنه لحساب الحد الأخير Un يجب حساب كل الحدود التي تسبقه انطلاقا من الحد U0، U1، ....، Un-1، وهذا ما تم باستعمال الحلقة pour، والتي تقوم بحساب الحدود من الحد

U1 إلى الحد Un عبر التعليمة U1 ← U0 +1/2 *(1/i) بحيث نضع في المتغيرة U0 قيمة الحد Ui-1 وفي المتغيرة U1 قيمة الحد Ui والذي يساوي $U_{i-1} + \frac{1}{2i}$، وبهذا عند وصول i إلى n يكون قد تم حساب Un.

حل التمرين الرابع:

```
Algorithme exo4
var x, U0,U1 : réel
début
        écrire( "Donnez une  valeur strictement positive de x" )
        répéter
        Lire(x)
        Jusqu'à x > 0
        U0  ← 1
        U1 ← (U0 + x/U0)/2
        tant que  abs(U1-U0) > 10^(-6)
                faire
                        U0  ← U1
                        U1 ← (U0 + x/U0)/2
                fin tant que
        écrire ("la racine carrée de ", x, "  est : ",U1)
fin
```

مناقشة الحل:

بما أننا يصدد حساب الجذر التربيعي لعدد حقيقي فيجب أن يكون هذا العدد موجبا تماما باعتبار طريقة هيرون تحسب الجذر بالنسبة للأعداد الحقيقية الموجبة تماما، لذا فقد بدأت هذه الخوارزمية كسابقتها بطلب إدخال قيمة x والتي يجب أن تكون موجبة تماما، وبذلك تم استخدام التعليمة Répéter jusqu'à من أجل إرغام المستخدم على إدخال قيمة موجبة تماما، وذلك من خلال الجملة (jusqu'à x > 0) والتي من شأنها مراقبة القيمة المدخلة، بحيث لا يتم الخروج من الحلقة حتى يصبح الشرط صحيح و x موجبة تماما.

بعد إدخال قيمة x، نقوم بإعطاء قيمة الحد الأول U0 والتي تساوي 1 (U0 ← 1) بعدها نقوم بحساب قيمة الحد الثاني U1 من خلال التعليمة U1 ← (U0 + x/U0)/2، ثم نمر إلى حساب الجذر التربيعي لـ x من خلال العلاقة التالية:

$$\begin{cases} U_0 = 1 \\ U_n = \frac{1}{2}(U_{n-1} + \frac{x}{U_{n-1}}) \end{cases}$$

والتي تعتبر علاقة تكرارية تنتهي إما بتحديد قيمة n أو بتحديد دقة الحساب من خلال الفرق بين حدين متتاليين، وفي حالة الخوارزمية السابقة فقد تم تحديد طريقة توقف العلاقة التكرارية عند بلوغ الفرق بين حدين متتالين قيمة أصغر أو يساوي 10^{-6}، ومن أجل تحقيق ذلك تمت الاستعانة بالحلقة tant que ووضع شرط الخروج (abs(U1-U0) > 10^(-6)، ويجب التنويه إلى أن استعمال الحلقة tant que جاء لعدم إمكانية استعمال الحلقة pour بحيث أن عدد التكرار لا يمكن معرفته مسبقا، كما تم حساب U0 و U1 قبل الدخول في الحلقة tant que من أجل حساب شرط الخروج، فعلى سبيل

المثال عند إدخال قيمة x = 1 فإن U0 = 1 و $U_1 = \frac{1}{2}(1 + \frac{1}{1})$ أي U1 = 1 وبهذا فإن U1-U0 = 0 وبما أن $0 < 10^{-6}$ فلا يتم الدخول إلى الحلقة tant que ويتم مباشرة كتابة النتيجة التالية:

la racine carrée de 1 est : 1 من خلال تنفيذ التعليمة écrire ("la racine carrée de ", x, " est : ",U1).

حل التمرين الخامس:

```
Algorithme Exo 5
var       P, i, a ,b : entier

début
        écrire( "Donnez deux  valeurs a et b tel que a > 0 et b > a " )
         répéter
         lire(a)
         jusqu'à a > 0
        répéter
         lire(b)
         jusqu'à b > a
    P  ← 1
       pour i ← a à b
                  faire
                        P  ← P *(5*a+b)
                  fin pour
        écrire ("la valeur calculée est égale à  : ",P)
fin
```

مناقشة الحل:

تقوم الخوارزمية أعلاه بحساب ((5*a + b) * (5*a + b) * …* (5*a + b)) وهذا باستعمال الحلقة pour بحيث تبدأ مجال الحساب من a وتنتهي بـ b، ولكن قبل تنفيذ الحلقة pour نقوم بإرغام المستخدم على إدخال قيم a وb يحققان الشروط التالية a > 0 وb > a وهذا باستخدام الحلقة répéter jusqu'à والجمل التالية: jusqu'à a > 0 و jusqu'à b > a.

وبما أن مجال الحلقة pour يبدأ بـ a وينتهي بـ b وفي كل مرة يتم ضرب القيمة القديمة بـ (5*a + b) من خلال التعليمة P ← P *(5*a+b) فإننا بصدد حساب:

$$\underbrace{((5*a + b) * (5*a + b) * \ldots * (5*a + b))}_{b - a + 1 \text{ مرة}}$$

بحيث المجال [a, b] يحتوي على b – a + 1 عدد طبيعي، وبهذا فإن هذه القيمة تمثل: $(5a+b)^{b-a+1}$.

حل التمرين السادس:

```
Algorithme Exo6
var  e_x, x : réel
     i, n, fact : entier
début
        répéter
        écrire( "Donnez la  valeur de x" )
```

```
        lire(x)
        jusqu'à x >  0
        fact  ← 1
        e_x ←1
         i ← 0
         tant que  x^i/fact  >  10^(-3)
                    faire
                          i ← i + 1
                          fact  ← fact * i
                          e_x ←e_x + x^i/fact
                    fin tant que
          écrire ("L'exponentielle de ",  x,  "est  : ",e_x)
 fin
```

مناقشة الحل:

الخوارزمية أعلاه تشبه إلى حد كبير خوارزمية حساب الجذر التربيعي، بحيث بعد إرغام المستخدم على إدخال عدد حقيقي موجب تماما تقوم بحساب القيم الأولية وهي fact = 1 و e_x = 1 وهذا باعتبار 1 = ! 0 و $e^0 = 1$ ثم تقوم بإعطاء القيمة الأولية لـ i وتبدأ بتنفيذ الحلقة tant que مع شرط الخروج 10^(-3) > x^i/fact ثم تمر إلى حساب قيمة e^x من خلال العلاقة التالية:

$$e^x = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots + \frac{x^n}{n!}$$

بحيث في كل تكرار نقوم بإضافة $\frac{x^i}{i!}$ إلى المجموع السابق فيصبح كالآتي: $e^x = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \cdots + \frac{x^{i-1}}{(i-1)!} + \frac{x^i}{i!}$

ويتوقف الحساب عند بلوغ الحد الأخير $\frac{x^i}{i!}$ قيمة أصغر أو يساوي 10^{-3}.

حل التمرين السابع:

```
Algorithme Exo7
var  n, nb, somme: entier
      moyenne : réel
début
   répéter
   écrire ("donnez un nombre naturel")
   lire(n)
   jusqu'à  n > = 0
   nb ←1
   somme ← n
   tant que somme < 1000
   faire
     répéter
     écrire ("donnez un nombre naturel")
     lire(n)
     jusqu'à  n > = 0
     nb ← nb + 1
     somme ← somme + n
   fin tant que
   moyenne ← somme / nb
   écrire("la moyenne des nombres saisis est :", moyenne)
   écrire("la somme des nombres saisis est :", somme)
fin
```

مناقشة الحل:

الخوارزمية أعلاه تقوم بطلب إدخال مجموعة من الأعداد الطبيعية لذا فهي تستعمل أولا التعليمة répéter jusqu'à لإرغام المستخدم على إدخال أعداد صحيحة موجبة، ثم تقوم بوضع العدد المدخل أولا في المتغيرة somme من خلال التعليمة somme ← n والمتغيرة nb تضع فيها 1 وهذه تعتبر حالة خاصة يمكن أن تحدث عبر إدخال قيمة للعدد الأول تفوق 1000، وبهذا لا يتم الدخول إلى الحلقة tant que والذي يشترط في الدخول إليها مجموع الأعداد يقل أو يساوي 1000، ثم وفي حالة ما إذا كان العدد الأول أصغر من 1000 يتم الدخول إلى الحلقة، وفي كل تكرار نقوم بجمع الأعداد ونضعها في المتغيرة somme كما نقوم بتعداد الأعداد المدخلة من خلال التعليمة nb ← nb + 1، ثم بعد الخروج من الحلقة tant que نقوم بحساب معدل الأعداد المدخلة باستعمال التعليمة moyenne ← somme / nb، وفي الأخير يتم عرض النتائج على الشاشة.

حل التمرين الثامن:

```
Algorithme Exo8
var i, n, NB_50, S_50, NB_100, S_100 : entier
début
   NB_50  ← 0
   S_50  ← 0
   NB_100  ← 0
   S_100  ← 0
   pour  i ← 1 à 100
     faire
       répéter
        lire(n)
        jusqu'à n > = 0
        si n <  50
          alors
            NB_50 ← NB_50 + 1
            S_50 ← S_50 + n
        fsi
        si n > 100
          alors
            NB_100 ← NB_100 + 1
            S_100 ← S_100 + n
        fsi
    fin pour
écrire (NB_50, S_50, NB_100, S_100)
 fin
```

مناقشة الحل:

الخوارزمية أعلاه تشبه خوارزمية التمرين السابع، إلا أنها تختلف معها في عدد التكرار المعروف مبدئيا وهو 100، وبهذا استعملنا الحلقة pour ثم عند قراءة الأعداد يتم فرزها وتقسيمها إلى أعداد أصغر من 50 وأكبر من 100 وهذا باستعمال تعليمتين شرطيتين بسيطتين. في الأخير يتم عرض النتائج على الشاشة.

حل التمرين التاسع:

عملية محاكاة التنفيذ (déroulement)

```
Algorithme  calcul9
1)  var  a, b, d, m, s : entier
début
      lire(a, b)
2)      d← 0
3)      m← b
4)     s← 0
       tant que m <= a
            faire
5)                 s ← m
6)                 m ← m +b
7)                 d ←  d + 1
            fin tan que
8)           m ←  a - s
     écrire (" d= ",d, " m= " ,m)
fin
```

1) a =15 , b = 3

الشاشة

15	**3**
5	**0**

الذاكرة

N°	d	m	s
1	?	?	?
2	0	?	?
3	0	3	?
4	0	3	0
5	0	3	3
6	0	6	3
7	1	6	3
5	1	6	6
6	1	9	6
7	2	9	6
5	2	9	9
6	2	12	9
7	3	12	9
5	3	12	12
6	3	15	12
7	4	15	12
5	4	15	15
6	4	18	15
7	5	18	15
8	**5**	**0**	15

2) a =34 , b = 10

الشاشة

34	**10**
3	**4**

الذاكرة

N°	d	m	s
1	?	?	?
2	0	?	?
3	0	10	?
4	0	10	0
5	0	10	10
6	0	20	10
7	1	20	10
5	1	20	20
6	1	30	20
7	2	30	20
5	2	30	30
6	2	40	30
7	3	40	30
8	**3**	**4**	30

- دور هذه الخوارزمية هو حساب حاصل القسمة الصحيحة لـ a على b ووضعه في المتغيرة d وباقي هذه القسمة ووضعه في المتغيرة m.

- نعم توجد دوال تعطي لنا القيم d وm دون استعمال الحلقات و هذه الدوال هي : div تعطي لنا d و mod وتعطي لنا m.

تعويض الدوال داخل الخوارزمية:

```
Algorithme  calcul9
  var a, b, d, m, s : entier
début
      lire(a, b)
      d ← a div b
      m ← a mod b
     écrire (" d= ",d, " m= " ,m)
fin
```

تعويض الحلقة tant que بالحلقة répéter jusqu'à:

```
répéter
              s ← m
              m ← m +b
             d ←  d + 1
jusqu'à m > a
```

مناقشة الحل:

نستطيع تعويض الحلقة tant que بالحلقة répéter jusqu'à ولكن في حالة ما إذا كان a >=b بحيث في هذه الحالة يتم تنفيذ التكرار على الأقل مرة واحدة، أما في حالة ما إذا كان a < b في هذه الحالة التكرار لا ينفذ ولا يتم الدخول إلى الحلقة tant que إطلاقا، وبهذا لا نستطيع تعويضها بالحلقة répéter jusqu'à والتي تنفذ مرة واحدة على الأقل.

أما فيما يخص محاكاة التنفيذ (déroulement)، فالملاحظ أن التعليمات 5، 6 و7 يتم تكرارها عدة مرات وهي التي تقع داخل الحلقة، بحيث لا يتم المرور إلى التعليمة 8 إلا بعد الخروج من الحلقة أي بعد ما تتغير قيمة m وتصبح a < m.

حل التمرين العاشر:

```
Algorithme calcul10
var A,B,R,S : entier
début
 1)  lire(A,B)
 2)   R ← 1
      tant que B <> 0
       faire
 3)       S = B mod 2
        si S = 0
          alors
 4)          A  ← A∗A
 5)         B  ← B div 2
        sinon
 6)          R  ← R∗A
 7)          B←B – 1
     fsi
```

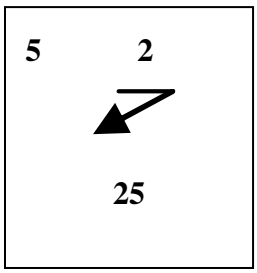
```
    fin tant que
    écrire("le résultat est :",R)
fin
```

1) محاكاة التنفيذ باليد (déroulement) (B=2 et A=5)

الذاكرة

N°	A	B	R	S
1	5	2	?	?
2	5	2	1	?
3	5	2	1	0
4	25	2	1	0
5	25	1	1	0
3	25	1	1	1
6	25	1	25	1
7	25	0	**25**	1

الشاشة

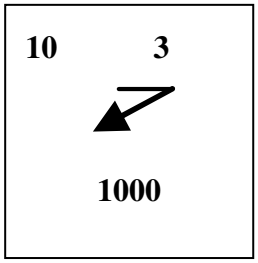


محاكاة التنفيذ باليد (déroulement) (B=3 et A=10)

الذاكرة

N°	A	B	R	S
1	10	3	?	?
2	10	3	1	?
3	10	3	1	1
6	10	3	10	1
7	10	2	10	1
3	10	2	10	0
4	100	2	10	0
5	100	1	10	0
3	100	1	10	1
6	100	1	1000	1
7	100	0	**1000**	1

الشاشة



- دور هذه الخوارزمية هو حساب A^B أي A أس B ووضعه في المتغيرة R.

تعويض الحلقة tant que بالحلقة pour:

- لا يمكن تعويض الحلقة tant que بالحلقة pour لأن عدد التكرار غير معروف مبدئيا.

تعويض الحلقة tant que بالحلقة répéter jusqu'à:

```
R ← 1
répéter
S ← B mode 2
 si S = 0
  alors
       A←A*A
       B← B div2
  sinon
   R← R*A
   B← B-1
 fsi
jusqu' à B = 0
```

مناقشة الحل:

لا نستطيع تعويض الحلقة tant que بالحلقة pour لأن عدد التكرار غير معروف مبدئيا، كما لا نستطيع تعويض

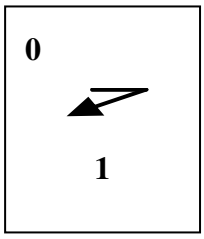
الحلقة tant que بالحلقة répéter jusqu'à في حالة ما إذا كانت التعليمة tant que يمكن أن لا تنفذ ولا مرة، بالنسبة لعملية محاكاة التنفيذ (déroulement) الملاحظ أنه في كل خطوة تنفذ تعليمة واحدة فقط وتتغير قيمة متغير واحد فقط، لذا ننصح بأن يتم ترقيم التعليمات ثم متابعة تغير المتغيرات عند تنفيذ كل تعليمة.

حل التمرين الحادي عشر:

```
Algorithme calcul11
var n,b,c : entier
début
 1)  lire(n)
   si n < 0
   alors
     écrire("pas de solution")
   sinon si n = 0
           alors
2)           b ← 1
            écrire(b)
          sinon
3)          c ← 1
4)          b ← 1
           tant que c <= n
           faire
5)            b←b*c
6)            c ← c+1
           fin tant que
           écrire(b)
         fsi
    fsi
fin
```

1) محاكاة التنفيذ باليد (déroulement) (n = 0)

الشاشة



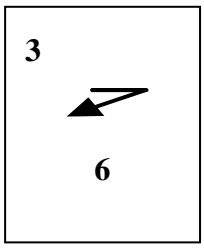
الذاكرة

N°	n	b	c
1	0	?	?
2	0	1	?

0 ! = 1

2) محاكاة التنفيذ باليد (déroulement) (n = 3)

الشاشة



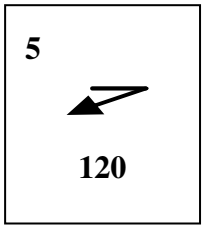
الذاكرة

N°	n	b	c
1	3	?	?
3	3	?	1
4	3	1	1
5	3	1	1
6	3	1	2
5	3	2	2
6	3	2	3
5	3	6	3
6	3	6	4

3! = 3 x 2x 1 = 6

3) محاكاة التنفيذ باليد (déroulement) (n = 5)

الشاشة



الذاكرة

N°	n	b	c
1	5	?	?
3	5	?	1
4	5	1	1
5	5	1	1
6	5	1	2
5	5	2	2
6	5	2	3
5	5	6	3
6	5	6	4
5	5	24	4
6	5	24	5
5	5	120	5
6	5	**120**	6

5! = 5 x 4 x 3 x 2x 1 =

دور هذه الخوارزمية هو البحث عن العاملي لـ n ! مع $n \geq 0$

تعويض الحلقة tant que بالحلقة pour:

```
b ← 1
pour c ← 1 à n
        faire
              b← b*c
      fin pour
```

تعويض الحلقة tant que بالحلقة répéter jusqu'à:

```
c ← 1
b ← 1
répéter
    b←b*c
    c ← c+1
jusqu'à c > n
```

مناقشة الحل:

في الخوارزمية السابقة استطعنا تعويض الحلقة tant que بالحلقة pour وهذا راجع إلى معرفتنا المبدئية لعدد التكرار وهو n إلى من 1 إلى n، كما أننا نستطيع تعويض الحلقة tant que بالحلقة répéter jusqu'à لأن التعليمة tant que ستنفذ على الأقل مرة واحدة.

حل التمرين الثاني عشر:

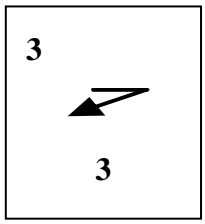
```
Algorithme  calcul12
 var A, M, Result: entier
début
1)   lire(A)
2)   M ←A mod 10
3)   Result← M
4)   A ← A div 10
     tant que  A <> 0
5)      M ← A mod 10
6)      Result ← Result + M
7)      A ← A div 10
    fin tant que
       écrire(Result)
 fin
```

1) محاكاة التنفيذ باليد (déroulement) (A = 3)

الذاكرة

N°	A	M	Result
1	3	?	?
2	3	3	?
3	3	3	3
4	0	3	3

الشاشة



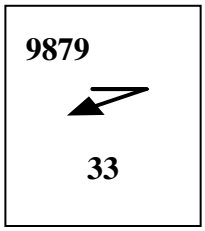
مجموع أرقام العدد 3 هو 3

2) محاكاة التنفيذ باليد (déroulement) (A = 9879)

الذاكرة

N°	A	M	Result
1	9879	?	?
2	9879	9	?
3	9879	9	9
4	987	9	9
5	987	7	9
6	987	7	16
7	98	7	16
5	98	8	16
6	98	8	24
7	9	8	24

الشاشة



مجموع أرقام العدد 9879 هو 9 + 7 + 8 + 9 = 33

5	9	9	24
6	9	9	33
7	0	9	**33**

3) محاكاة التنفيذ باليد (déroulement) (A = 17984)

الذاكرة

N°	A	M	Result
1	17984	?	?
2	17984	4	?
3	17984	4	4
4	1798	4	4
5	1798	8	4
6	1798	8	12
7	179	8	12
5	179	9	12
6	179	9	21
7	17	9	21
5	17	7	21
6	17	7	28
7	1	7	28
5	1	1	28
6	1	1	29
7	0	1	**29**

الشاشة

17984

29

مجموع أرقام العدد 17984 هو 4 + 8 + 9 + 7 + 1 = 29

- دور هذه الخوارزمية هو حساب مجموع الأرقام المكونة لعدد طبيعي

تعويض الحلقة tant que بالحلقة répéter jusqu'à:

```
A ← A div 10
répéter
  M ← A mod 10
  Result ← Result + M
  A ← A div 10
 jusqu'à A = 0
```

مناقشة الحل:

الخوارزمية أعلاه تبحث عن أرقام الآحاد وأرقام العشرات والمئات، و....، وذلك باستعمال الدالتين mod وdiv ثم تقوم بجمع هاته الأرقام، الملاحظ أننا استعملنا mod وdiv على العدد 10 بحيث كل ما قسمنا عددا على 10 نحصل على عدد العشرات المكونة أما باقي القسمة على 10 يمثل عدد الآحاد فيه، وهكذا تتواصل العملية حتى الانتهاء من جمع الأرقام المكونة له.

حل التمرين الثالث عشر:

```
Algorithme  calcul13
  var A, B, C : entier
 Result: réel
début
1) lire(A,B)
2) C ←0
   si A = 0
```

1) محاكاة التنفيذ باليد (déroulement) (A =2, B = -3)

```
    alors
      si B <= 0
        alors
          écrire("Pas de solutions")
        sinon
3)          Result ← 0
          écrire(Result)
      fsi
  sinon
4)  Result ← 1
    si B < 0
      alors
5)      B ← -B
6)      C ← 1
    fsi
   tant que B  > 0
     faire
7)  Result ← Result * A
8)      B ←  B - 1
    fin tant que
       si C = 1
         alors
 9)          Result ← 1/Result
       fsi
       écrire(Result)
    fsi
fin
```

الشاشة

2 -3

0.125

الذاكرة

N°	A	B	C	Result
1	2	-3	?	?
2	2	-3	0	?
4	2	-3	0	1
5	2	3	0	1
6	2	3	1	1
7	2	3	1	2
8	2	2	1	2
7	2	2	1	4
8	2	1	1	4
7	2	1	1	8
8	2	0	1	8
9	2	0	1	**1/8 = 0.125**

$2^{-3} = 0.125$

2) محاكاة التنفيذ باليد (déroulement) (A =0, B = 5)

الشاشة

0 5

0

الذاكرة

N°	A	B	C	Result
1	0	5	?	?
2	0	5	0	?
3	0	5	0	**0**

$0^5 = 0$

3) محاكاة التنفيذ باليد (déroulement) (A =3, B = 0)

الشاشة

3 0

1

الذاكرة

N°	A	B	C	Result
1	3	0	?	?
2	3	0	0	?
4	3	0	0	**1**

$3^0 = 1$

4) محاكاة التنفيذ باليد (déroulement) (A =4, B = 2)

الشاشة

4 2

16

الذاكرة

N°	A	B	C	Result
1	4	2	?	?
2	4	2	0	?
4	4	2	0	1
7	4	2	0	4
8	4	1	0	4
7	4	1	0	16
8	4	0	0	**16**

$4^2 = 16$

- دور هذه الخوارزمية هو حساب A^B ، A أس B مع دراسة كل حالات A وB.

تعويض الحلقة tant que بالحلقة répéter jusqu'à:

```
répéter
  Result ← Result * A
  B ←  B - 1
jusqu'à      B <= 0
```

مناقشة الحل:

مع أنه عموما يمكن تعويض الحلقة tant que في الخوارزمية السابقة بالحلقة répéter jusqu'à ، لكن يجب توخي الحذر في الحالات الخاصة أين لا تنفذ الحلقة tant que ولا مرة، ففي حالة ما إذا B = 0 و A = 3 يجب أن لا تنفذ الحلقة tant que ولا مرة ، وفي حالة تعويضها بالحلقة répéter jusqu'à قد يحصل خطأ في نتيجة الخوارزمية بحيث عوض أن تكون النتيجة 1 باستعمال الحلقة tant que نحصل على 3 بالحلقة répéter jusqu'à، وهذا راجع لتنفيذ التعليمة مرة واحدة على الأقل.

حل التمرين الرابع عشر:

```
Algorithme Exo14
var  i, n, m : entier
     succession : booléen
début
    lire(n)
    m ← n
    i ← 1
   succession ← vrai
  tant que i < = 9 ET succession = vrai
     faire
      lire (n)
      si n <> m + 1
        alors
          succession = faux
     fsi
     i ← i + 1
  fin tant que
   si succession = vrai
     alors
        écrire ("les nombres saisis sont consécutifs")
     sinon
       écrire ("les nombres saisis sont non consécutifs")
   fsi
 fin
```

مناقشة الحل:

تستعمل الخوارزمية أعلاه حلقة tant que لمراقبة تتابع الأعداد الواحد بعد الآخر (n, n+1, n+2, ….,n+9) بحيث في حالة عدم تحقق هذا التتابع (أي نقوم بإدخال عدد لا يساوي العدد الذي يتبع العدد السابق) فإن الحلقة تتوقف ولا يتم مواصلة الدخول إليها من خلال تغير شرط الخروج (succession = faux)، ثم حسب قيمة المتغير المنطقي succession نقوم بعرض النتيجة.

حل التمرين الخامس عشر:

```
Algorithme Exo15
var  i,  n, Nb_P, Nb_I, somme_I, somme_P, M_P, M_I : entier
début
   Nb_P ← 0
   Nb_I ← 0
   somme_I ← 0
   somme_P ← 0
   pour i ← 1 à 100
    faire
      répéter
       lire(n)
      jusqu'à n >= 0
     si n mod 2 = 0
      alors
          Nb_P ← Nb_P + 1
          somme_P ← somme_P + n
     sinon
          Nb_I ← Nb_I + 1
          somme_I ← somme_I + n
    fsi
   fin pour
   M_P ← somme_P/Nb_p
   M_I ← somme_I/Nb_I
   écrire ("le nombre des nombres pairs est : ", Nb_P)
   écrire ("le nombre des nombres impairs est : ", Nb_I)
   écrire ("la moyenne des nombres pairs est : ", M_P)
   écrire ("la moyenne des nombres impairs est : ", M_I)
fin
```

مناقشة الحل:

تستعمل الخوارزمية أعلاه حلقة pour لحساب مجموع الأعداد الزوجية باستعمال المتغيرة somme_P كما تستعمل المتغيرة Nb_P لحساب عدد الأعداد الزوجية، ثم يتم حساب معدل الأعداد الزوجية الأعداد الفردية بقسمة عدد الأعداد على مجموعها، ولتحديد إن كان العدد زوجي أو فردي استعملنا الدالة mod والتي تعطينا باقي القسمة العدد على 2. بحيث إذا كان زوجيا يكون الباقي معدوما أما إذا كان فرديا فيكون الباقي غير معدوم.

الفصل الرابع

الفصل الرابع: مسائل محلولة في الخوارزميات

تمهيد:

خصصنا هذا الفصل لمجموعة من المسائل المقترحة في الخوارزميات، وهذا انطلاقا من المسائل التي أدرجت في امتحانات السنوات الماضية، والهدف من طرح مثل هذه المسائل البسيطة هو تدريب الطالب على التعامل مع الحالات التطبيقية القريبة إلى الواقع، واقترابه أكثر إلى الدراسة الميدانية بدل البقاء في التمارين البيداغوجية الكلاسيكية التي لا تمت بأي صلة إلى الواقع المعيش والمهني.

I. المسألة الأولى:

يريد صاحب موقف سيارات حساب إيراداته اليومية فيكلفك بكتابة خوارزمية تساعده على ذلك.

معطيات الموقف كالآتي :

- يفتح الموقف أبوابه على الساعة الثامنة صباحا (08 :00h)، بحيث يكون فارغا قبل هذا الوقت، ويغلقها على الساعة الثامنة مساءا (20 :00h)، بحيث لا تبقى أية سيارة داخل هذا الموقف.
- يدفع كل صاحب سيارة مبلغا جزافيا (Forfait) يقدر بـ 100 دج كحق الدخول إلى الموقف.
- يدفع صاحب السيارة إضافة على المبلغ الجزافي مبلغ 50 دج عن كل ساعة توقف داخل الموقف.

يجب على الخوارزمية معرفة عدد السيارات التي توقفت خلال اليوم (nb_vehicule) مدة توقف كل سيارة (عدد ساعات التوقف nb_heure) ساعة الدخول (h_entree) ساعة الخروج (h_sortie) مبلغ توقف كل سيارة (prix_park) مجموع مبالغ توقف كل السيارات و الذي يمثل إيرادات الموقف في اليوم (recette).

حل المسألة الأولى:

```
Algorithme  Parking_V1
var   nb_vehicule, nb_heure, h_entree, h_sortie, prix_park, recette, i : entier
début
  répéter
     écrire ("Donnez le nombre de véhicule")
     lire(nb_vehicule)
 jusqu'à nb_vehicule > 0
i ←1
recette ← 0
tant que i <= nb_vehicule
 faire
  répéter
     écrire ("Donnez l'heure d'entrée")
```

```
    lire(h_entree)
jusqu'à h_entree  >= 8  et h_entree <= 20
répéter
    écrire ("Donnez l'heure de sortie")
    lire(h_sortie)
 jusqu'à h_sortie  >= h_entree    et h_sortie <= 20
 nb_heure ← h_sortie - h_entree
 prix_park ← 100 + nb_heure * 50
recette ← recette  + prix_park
 i ← i + 1
fin tant que
 écrire ("La recette journalière du parking est : ", recette)
fin
```

مناقشة الحل:

الخوارزمية أعلاه تقوم بحساب الإيرادات اليومية لموقف سيارات يُسَيَّرُ بالنظام الساعي، وهذا لحث أصحاب المركبات على عدم ترك مركباتهم مدة طويلة داخل الموقف، بحيث كلما طالت مدة مكوث مركباتهم داخل الموقف كلما ارتفع مبلغ التوقف، ولحساب الإيرادات اليومية لهذا الموقف يجب أولا معرفة عدد السيارات التي توقفت في اليوم، ومن أجل هذا استعملنا الحلقة répéter jusqu'à في أول الخوارزمية لإرغام المستخدم على إعطاء عدد سيارات موجب تماما من خلال استعمال العبارة jusqu'à nb_vehicule > 0، بحيث لا يمكن لعدد السيارات أن يكون سالبا، كما لا يمكن أن يكون معدوما بحيث في هذه الحالة الإيرادات اليومية تكون هي الأخرى بالضرورة معدومة.

بعد معرفة عدد السيارات التي توقفت خلال اليوم (nb_vehicule) نمر إلى حساب الإيرادات اليومية لهذا الموقف، من أجل هذا استعملنا حلقة tant que تبدأ بحساب مجموع المبالغ التي دفعها أصحاب المركبات التي توقفت داخل الموقف خلال اليوم الواحدة تلو الأخرى، بحيث كل مركبة نقوم بحساب مدة توقفها داخل الموقف (nb_heure)، وبما أن مدة التوقف هي ساعة الخروج – ساعة الدخول (h_sortie – h_entrée)، فعلى الخوارزمية أن تطلب من المستخدم إدخال ساعة الدخول والتي يجب أن تكون داخل المجال [8 , 20] أي بعد أن يفتح الموقف أبوابه على الساعة 08h : 00 وقبل أن يغلق أبوابه على الساعة 20h : 00، وللحصول على هذا استعملنا الحلقة répéter jusqu'à لإرغام المستخدم على إعطاء ساعة دخول ضمن المجال المذكور من خلال العبارة: jusqu'à h_entre >= 8 ET h_entree <= 20 ، ثم بعد معرفة ساعة الدخول نمر إلى طلب ساعة الخروج والتي يجب بدورها أن تكون بعد ساعة الدخول، بحيث من غير المعقول أن تخرج مركبة من الموقف قبل أن تدخل ! وبذلك تكون مدة مكوثها داخل الموقف سالبة، كما يجب أن تكون ساعة الخروج قبل إغلاق الموقف، وبهذا يجب على ساعة الخروج أن تكون ضمن المجال [h_entree , 20] ولإرغام المستخدم على إدخال ساعة الخروج ضمن المجال المذكور استخدمنا حلقة répéter jusqu'à مع العبارة التالية: jusqu'à h_sortie >= h_entre ET h_sortie <= 20 ، بعد معرفة ساعة الدخول وساعة الخروج لم يتبق إلا حساب مدة التوقف nb_heure ثم حساب مبلغ التوقف والذي يحسب كالآتي: prix_park = 100 + nb_heure * 50، بعد حساب مبلغ توقف المركبة الحالية نمر إلى المركبة التالية مع جمع المبالغ الواحد بعد الآخر حتى الانتهاء من عد كل المركبات.

الجدير بالذكر أننا استعملنا الحلقة tant que لحساب مجموع مبالغ توقف المركبات مع العلم أن عدد المركبات يكون معروفا مبدئيا وبذلك عدد التكرار يكون معروفا مبدئيا، وبهذا كان باستطاعتنا استعمال الحلقة pour والتي تجنبنا من كتابة تعليمة إعطاء القيمة الابتدائية (i ← 1) وكذا زيادة العداد (i ← i + 1) وكذلك شرط الخروج (i <= nb_vehicule)، لذا فالخوارزمية باستعمال الحلقة pour ستكون أكثر بساطة وأقل تعقيدا، وهو ما سنلاحظه من خلال الخوارزمية التالية التي استعملت الحلقة pour عوض الحلقة tant que.

```
Algorithme  Parking_V2
var   nb_vehicule, nb_heure, h_entree, h_sortie, prix_park, recette, i : entier
début
    répéter
       Ecrire ("Donnez le nombre de véhicule")
       lire(nb_vehicule)
     jusqu'à nb_vehicule > 0
   recette ← 0
   pour  i ← 1 à  nb_vehicule
    faire
      répéter
        écrire ("Donnez l'heure d'entrée")
         lire(h_entree)
      jusqu'à h_entree  >= 8  et h_entree <= 20
     répéter
        écrire ("Donnez l'heure de sortie")
        lire(h_sortie)
     jusqu'à h_sortie  >= h_entree    et h_sortie <= 20
    nb_heure ←  h_sortie - h_entree
    prix_park ← 100 + nb_heure * 50
    recette ← recette  + prix_park
 fin pour
    écrire ("La recette journalière du parking est : ", recette)
fin
```

II. المسألة الثانية:

قررت جامعة الجزائر3 إجراء مسابقة الدكتوراه للسنة الجامعية 2021/2020 بحيث يمتحن كل طالب في مقياسين: Module1 وModule2 ومعاملهما على التوالي: Coef1 و Coef2، كل مقياس يصحح مرتين، تصحيح أول وتصحيح ثاني.

مثلا Module1 تمنح له مبدئيا نقطتان Note11 و Note12، إذا كان الفرق بين النقطتين أقل تماما من 4 نقاط يمنح معدل النقطتين للمقياس:

Note1 = (Note11+Note12)/2 Note1) هي النقطة النهائية للمقياس Module1 وNote2 هي النقطة النهائية للمقياس Module2) أما إذا كان الفرق بين النقطتين أكبر أو يساوي 4 نقاط يلجأ المصححون إلى تصحيح ثالث ونقطة ثالثة Note13 أو Note23 ويمنح معدل النقاط الثلاثة للمقياس مثلا: نقطة Module1 هي Note1 = (Note11+Note12+Note13)/3

أكتب خوارزمية تطلب من المستخدم إدخال نقاط المقياسين (النقطة الأولى ، النقطة الثانية والنقطة الثالثة في حالة وجودها) ثم تحسب النقطة النهائية للمقياسين حسب المعطيات السابقة ، بعد ذلك تطلب من المستخدم إدخال معامل المقياسين، ثم تحسب و تعرض على الشاشة معدل الطالب (Moy) في المسابقة، ثم إذا كان معدل الطالب أكبر أو يساوي 12 ونقاط المقياسين لا تقل عن 10 يعتبر الطالب الناجح وتعرض الخوارزمية على الشاشة "طالب ناجح" أما عدا ذلك يعتبر الطالب راسب وتعرض الخوارزمية على الشاشة "طالب راسب".

ملاحظة: معامل المقياسين لا يقل عن 2 ولا يزيد عن 5.

حل المسألة الثانية:

```
Algorithme Doctorat_20-21
var coef1, coef2,: entier
Note11, Note12, Note13, Note1, Note21, Note22, Note23, Note2, Moy: réel
début
    répéter
    écrire ("donner la première note du 1er module")
    lire(Note11)
    jusqu'à Note11  >= 0 ET Note11  <= 20
    répéter
    écrire ("donner la deuxième note du 1er  module")
    lire(Note12)
    jusqu'à Note12  >= 0 ET Note12  <= 20
    si abs(Note11 – Note12) < 4
     alors
         Note1 ← (Note11 + Note12)/2
     sinon
     répéter
     écrire ("donner la troisième note du 1er  module")
     lire(Note13)
     jusqu'à Note13  >= 0 ET Note13  <= 20
    Note1 ← (Note11 + Note12+ Note13)/3
    fsi
    répéter
    écrire ("donner la première note du 2ème  module")
    lire(Note21)
    jusqu'à Note21  >= 0 ET Note21  <= 20
    répéter
    écrire ("donner la deuxième note du 2 ème  module" )
    lire(Note22)
    jusqu'à Note22  >= 0 ET Note22  <= 20
    si abs(Note21 – Note22) < 4
     alors
         Note2 ← (Note21 + Note22)/2
     sinon
     répéter
     écrire ("donner la troisième note du 2 ème  module" )
     lire(Note23)
     jusqu'à Note23  >= 0 ET Note23  <= 20
     Note2 ← (Note21 + Note22+ Note23)/3
    fsi
répéter
     écrire ("donner le coefficient du 1er  module" )
     lire(coef1)
     jusqu'à coef1  >= 2 ET coef1  <= 5
répéter
     écrire ("donner le coefficient du 2 ème module")
```

```
        lire(coef2)
        jusqu'à coef2  >= 2 ET coef2  <= 5
       Moy ← (Note1* coef1 + Note2*coef2)/(coef1+coef2)
       écrire ("la moyenne de l'étudiant est : ", Moy )
       si Moy >=12 ET Note1 >= 10 ET Note2 >=10
        alors
            écrire(" طالب ناجح")
        sinon
            écrire(" طالب راسب")
fin
```

مناقشة الحل:

الخوارزمية أعلاه تقوم بحساب معدلات الطلبة في مسابقة الدكتوراه، وعموما مثل هذه المسابقات والتي تكتسي أهمية بالغة بالنسبة للطلبة من جهة وبالنسبة للإدارة من جهة أخرى، يكون التقييم فيها يخضع لنفس المعايير التي تتبع في امتحانات البكالوريا والمسابقات والامتحانات الرسمية الأخرى، بحيث ورقة امتحان الطالب يجب أن تصحح بواسطة رقم إغفال (code anonymat) تضمن السرية والمساواة بين كل الطلبة، كما تصحح ورقة كل طالب من قبل مصححين اثنين لمزيد من الدقة والعدل في التقييم، لأجل هذا وفي حالة وجود فرق كبير بين التصحيحين يُلجأ إلى تصحيح ثالث ليكون الفيصل في تقييم الطالب بحيث في الواقع يعطى المعدل بين التصحيح الثالث والتصحيح الأقرب إليه ويلغى التصحيح البعيد، وفي حالة الخوارزمية المقترحة ولتبسيط المسألة على الطالب ارتأينا أن نأخذ معدل التصحيحات الثلاثة.

بدأت الخوارزمية بطلب إدخال نقطة الطالب بعد التصحيح الأول في المقياس الأول (Note11) هذه النقطة يجب أن تكون ضمن المجال المقبول [0, 20] بحيث لا تقبل النقاط الأقل من الصفر أو الأكبر من 20 ولإرغام المستخدم على إدخال نقطة مقبولة استعملنا كالعادة حلقة répéter jusqu'à مع شرط الخروج Note11 >= 0 ET Note11 <= 20 وهو ما يضمن أن نقطة التصحيح الأول ستكون ضمن المجال المذكور، ثم نقوم بنفس العملية في إدخال نقطة التصحيح الثاني Note12 ثم نقوم بمقارنة الفرق بين التصحيحين (Note11 – Note12)، يمكن للفرق بين نقطتي التصحيح الأول والتصحيح الثاني أن يكون موجبا في حالة ما إذا كانت نقطة التصحيح الأول أكبر من نقطة التصحيح الثاني، كما يمكن للفرق أن يكون سالبا في حالة ما إذا كانت نقطة التصحيح الأول أصغر من نقطة التصحيح الثاني، لذا استعملنا القيمة المطلقة للفرق (abs(Note11 – Note12))، ثم وباستعمال تعليمة شرطية متناوبة ندرس الحالتين التاليتين: إذا كان الفرق بين التصحيحين أصغر تمام من 4 يُمنح معدل النقطتين للمقياس الأول من خلال التعليمة التالية: Note1 ← (Note11 + Note12)/2، أما إذا كان الفرق أكبر أو يساوي 4 يُطلب من المستخدم إدخال نقطة التصحيح الثالث Note13 باستخدام نفس الطريقة التي تضمن نقاط مقبولة، وتُمنح في هذه الحالة للمقياس معدل نقاط التصحيحات الثلاثة من خلال التعليمة التالية: Note1 ← (Note11 + Note12 + Note13)/3، الطريقة نفسها تتبعها الخوارزمية في الحصول على نقطة المقياس الثاني، ثم يُطلب من المستخدم إدخال معاملي المقياسين حسب الشرط الموضوع وهو أن لا يتعدى معامل كل مقياس 5 ألاَّ يقل عن 2 ولضمان تحقق هذا الشرط استخدمنا الحلقة répéter jusqu'à.

في الأخير تحسب الخوارزمية معدل الطالب في المسابقة باستعمال نقطتي المقياسين المتحصل عليهما موزونة بمعامليهما، وتعرض هذا المعدل على الشاشة، كما تقوم بعرض نتيجة الطالب في المسابقة حسب الشروط الموضوعة من قبل الإدارة.

III. المسألة الثالثة:

يقدم صاحب مساحة كبرى لزبائنه إمكانية الخدمة الذاتية (self service) وذلك باستعمال آلة لتوزيع المنتجات. تحتوي هذه الآلة على ثلاثة منتجات: A1، A2، A3 : Type_prod وسعرهم على التوالي: P1، P2، P3 : Prix_prod. نأخذ على سبيل المثال (P1 = 500 DA, P2 = 400 DA, P3 = 200 DA) يقوم المستخدم باختيار المنتج والكمية المطلوبة Qte_prod، إذ لا يمكن للمستخدم اختيار إلا منتج واحد حسب رغبته، وذلك بتعيين رمز المنتج (A1، A2، A3). بعد إتمام عملية الشراء يقوم الزبون بإدخال المبلغ الواجب دفعه Prix_Achat في الآلة فتقوم بـ :

أ. كتابة: "La somme versée est inférieure à la somme due " عندما يكون المبلغ المدفوع Prix_paye أقل من المبلغ الواجب دفعه Prix_Achat.

ب. كتابة : " L'opération s'est terminée avec succès " عندما يكون المبلغ المدفوع يساوي المبلغ الواجب دفعه ثم تسلم الآلة المنتجات المطلوبة.

ج. كتابة: " L'opération s'est terminée avec succès, récupérez votre monnaie " عندما يكون المبلغ المدفوع أكبر من المبلغ الواجب دفعه، ثم تسلم الآلة المنتجات المطلوبة للزبون، كما تسلم الباقي Reste للزبون باستعمال أوراق فئة 500 دج و قطع فئة 100 دج و 10 دج و 5 دج.

1) أكتب خوارزمية تقوم بإدارة هذه الآلة حسب المعطيات السابقة وتقوم في حالة تسليم الباقي للزبون بإعطاء عدد الأوراق فئة 500 دج nb_500 و عدد القطع فئة 100 دج nb_100 وعدد القطع فئة 10 دج nb_10 وعدد القطع فئة 5 دج nb_5.

2) يريد صاحب المساحة الكبرى حساب عدد الأوراق من فئة 500 دج المستعملة خلال اليوم. ماذا نغير في الخوارزمية السابقة لتمكنه من ذلك. استعمل المتغيرات التالية : nb_500_g : العدد الكلي للأوراق من فئة 500 دج و nb_cl : العدد الكلي للزبائن.

حل المسألة الثالثة:

1)

```
Algorithme Self_service_1
const P1 = 500
      P2 = 400
      P3 = 200
var  type_prod : caractère
     Prix_prod, Qte_prod, Prix_achat, prix_paye, Reste, nb_500, nb_100, nb_10, nb_1 : entier
début
```

```
répéter
écrire("Entrez votre choix de produit :  A1, A2 ou A3")
 lire(Type_prod)
jusqu'à  Type_prod = "A1" OU Type_prod = "A2" OU Type_prod = "A3"
répéter
écrire("Entrez la quantité voulue")
 lire(Qte_prod)
jusqu'à  Qte_prod > 0
si Type_prod = "A1"
  alors
        Prix_achat ← P1 * Qte_prod
  sinon si Type_prod = "A2"
        alors
          Prix_achat ← P2 * Qte_prod
        sinon
          Prix_achat ← P3 * Qte_prod
      fsi
 fsi
écrire (''Entrez la somme due dans la machine'')
lire (Prix_paye)
tant que Prix_paye < prix_achat
   faire
     écrire("la somme versée est inférieur à la somme due ")
     lire(Prix_paye)
   fin tant que
si Prix_paye = Prix_achat
 alors
    écrire("l'opération s'est terminée avec succès, récupérez vos achats")
 sinon
  Nb_500 ← 0
  Nb_100 ← 0
  Nb_10 ← 0
  Nb_5← 0
  Reste ← Prix_paye  - prix_achat
  Nb_500 ← Reste div 500
  Reste  ← Reste  mod 500
  Nb_100 ← Reste div 100
  Reste  ← Reste  mod 100
  Nb_10 ← Reste div 10
  Reste  ← Reste  mod 10
  Nb_5  ← Reste  div 5
  écrire("l'opération s'est terminée avec succès, récupérez vos achats  et récupérez  votre monnaie")
  écrire("vous avez : ", nb_500, " billets  de 500 DA")
  écrire("vous avez : ", nb_100, " pièces  de 100 DA")
  écrire("vous avez : ", nb_10, " pièces  de 10 DA")
  écrire("vous avez : ", nb_5, " pièces  de 5 DA")
 fsi
fin
```

مناقشة الحل:

الخوارزمية السابقة تقوم بإدارة آلة الخدمة الذاتية في المساحة الكبرى المذكورة في نص المسألة، هذا النوع من الآلات

موجود بكثرة في الدول المتطورة، بحيث تستغني أكثر المساحات الكبرى عن موظفين التحصيل ويُستبدَلون بهذه الآلة إضافة إلى نظام مراقبة بالكاميرات، ولتجنب التعقيد وتبسيط الأمور على الطالب، ارتأينا أن يكون عدد المنتجات المتوفرة في المساحة الكبرى هي ثلاثة فقط، إضافة على إمكانية اقتناء نوع واحد من هاته المنتجات في كل عملية شراء، مع ترك الحرية في الكمية المشتراة.

تبدأ الخوارزمية بطلب نوع المنتجات المشتراة وتحصرها في المنتجات الثلاثة المذكورة أنفا A1, A2, A3، وللحصول على هذا تمت الاستعانة بالحلقة répéter jusqu'à، كما تم وضع الشرط التالي: Type_prod = "A1" OU Type_prod = "A2" OU Type_prod = "A3" والذي يرغم المستخدم على اختيار أحد هذه المنتجات الثلاثة، بحيث إذا كان Type_prod ≠ "A1" و Type_prod ≠ "A2" و Type_prod ≠ "A3" فإن المنتج المدخل لا يقبل ولا يتم الخروج من الحلقة répéter jusqu'à حتى يتم إدخال أحد المنتجات الثلاثة، بعد اختيار نوع المنتج يتم إدخال الكمية المشتراة والتي بطبيعة الحال يجب أن تكون موجبة تماما، ثم انطلاقا من نوع المنتج وكميته تقوم الخوارزمية بحساب الثمن الكلي للمشتريات وهذا باستعمال تعليمة شرطية مركبة تدرس الحالات الثلاثة.

انطلاقا من الثمن الكلي للمشتريات ومن المبلغ المدفوع، تقوم الخوارزمية بدراسة الحالات الثلاثة المذكورة في المسألة وهي:

- المبلغ المدفوع أصغر من المبلغ الواجب دفعه؛
- المبلغ المدفوع يساوي المبلغ الواجب دفعه؛
- المبلغ المدفوع أكبر من المبلغ الواجب دفعه.

ففي الحالة الأولى لا يتم تسليم المنتجات للزبون حتى يدفع مبلغ يساوي أو يفوق ثمن المشتريات، أما في الحالة الثانية يتم تسليم المشتريات للزبون، وأما في الحالة الثالثة فيتم تسليم المشتريات للزبون وإرجاع الباقي على شكل أوراق من فئة 500 دج أو قطع من فئة 100، 10 و5 دج.

ولحساب عدد الأوراق والقطع النقدية المستعملة في إرجاع الباقي تمت الاستعانة بالدالتين mod وdiv، بحيث على سبيل المثال إذا كان الباقي يساوي 475 دج، فإن عدد الأوراق من فئة 500 دج هو: 0 = 475 div 500 والباقي بعد إرجاع أوراق 500 دج هو 475 = 475 mod 500 ثم عدد القطع فئة 100 دج هو: 4 = 475 div 100 أما الباقي بعد إرجاع القطع من فئة 100 دج هو 75 = 475 mod 100، أما عدد القطع من فئة 10 دج هو: 7 = 75 div 10 والباقي بعد إرجاع قطع 10 دج هو: 5 = 75 mod 10، ثم عدد القطع من فئة 5 هو: 1 = 5 div 5.

2)

```
Algorithme Self_service_2_V1
  var  nb_cl, nb500_g, i : entier
 début
          répéter
          écrire("Donnez le nombre global de clients ")
           Lire(nb_cl)
          jusqu'à nb_cl  >  0
   i ← 0
```

```
  nb_500_g  ←0
 tant que i <= nb_cl
    faire
        nb_500_g ← nb_500_g  + nb_500
         i ← i + 1
    fin tan que
  écrire ("le nombre global de billets de type 500 DA est : ", nb_500_g)
fin

Algorithme Self_service_2_V2
 var  nb_cl, nb500_g, i : entier
 début
          répéter
          écrire("Donnez le nombre global de clients ")
           Lire(nb_cl)
          jusqu'à nb_cl  >  0
  nb_500_g  ←0
pour  i  ← 1 à  nb_cl
    faire
        nb_500_g ← nb_500_g  + nb_500
    fin pour
  écrire ("le nombre global de billets de type 500 DA est : ", nb_500_g)
fin
```

مناقشة الحل:

الخوارزميتان السابقتان تحسبان عدد الأوراق الكلي من فئة 500 دج الذين استعملوا في إرجاع الباقي للزبائن خلال اليوم (nb_500_g)، من أجل هذا وبعد معرفة العدد الكلي للزبائن (nb_cl) الذين قاموا بشراء منتجات من هذه المساحة الكبرى تم استخدام الحلقة tant que أو الحلقة pour لحساب مجموع الأوراق من فئة 500 دج من خلال إضافة عدد الأوراق من فئة 500 دج لكل زبون إلى العدد الإجمالي وإعادة العملية حتى الانتهاء من كل الزبائن.

IV. المسألة الرابعة:

قررت إدارة الوكالة الوطنية للطرق السريعة إطلاق نظام الدفع الخاص بالطريق السيار شرق–غرب حسب الطريقة التالية :

تقسم المركبات إلى ثلاث فئات، فئة السيارات النفعية والسياحية (Type_V = "V_LG") وفئة الحافلات وسيارات النقل الجماعي (Type_V = "V_TV")، وفئة الشاحنات ومركبات الوزن الثقيل (Type_V = "V_PL") ، يدفع أصحاب الفئة الأولى 1.5 دج عن كل كلم تقطعه مركباتهم عبر الطريق السيار أما أصحاب الفئة الثانية فيدفعون 3 دج عن كل كلم، بينما يدفع أصحاب الفئة الثالثة 5 دج عن كل كلم. وضعت الوكالة نقاط دفع عند مداخل و مخارج الطريق السيار بحيث عند دخول أي مركبة إلى الطريق يسجل برنامج خاص على بطاقة السير كلم البدء Km_D و فئة المركبة Type_V وعند خروج المركبة من الطريق يسجل البرنامج كلم الوصول Km_A على نفس البطاقة، ثم تسلم هذه البطاقة إلى الموظف المكلف بالتحصيل الذي يقوم بحساب المبلغ المستحق الواجب دفعه من طرف صاحب المركبة، يبلغ

طول الطريق السيار شرق–غرب 1216 كلم، بحيث يبدأ بالنقطة الكيلومترية 0 انطلاقا من الحدود الشرقية وينتهي بالنقطة الكيلومترية 1216 عند الحدود الغربية، وبهذا فإن أقصى نقطة كيلومترية هي 1216.

1) أكتب خوارزمية تقوم بمساعدة هذا الموظف لحساب المبلغ المستحق Montant_P الذي يتوجب على صاحب المركبة دفعه إلى موظف التحصيل.

2) في نهاية اليوم يريد مسؤول نقطة خروج معينة، حساب الإيرادات اليومية Recette لهذه النقطة، علما أن عدد المركبات التي عبرت خلال اليوم في هذه النقطة هي NB_V ، أكتب خوارزمية تحقق له ذلك.

حل المسألة الرابعة:

```
1)
Algorithme Autoroute_Est_Ouest_1
 var  KM_D, KM_A: entier
      Montant_P : réel
      Type_V : caractère
début
   répéter
   écrire("Donnez le type du véhicule")
         lire(Type_V)
        jusqu'à Type_V = "V_LG" OU Type_V = "V_TV"  OU  Type_V = "V_PL"
   répéter
   écrire("Donnez le kilomètre du départ")
         lire(KM_D)
        jusqu'à KM_D > =  0 ET KM_D  < 1216
  répéter
   écrire("Donnez le kilomètre d'arrivée")
         lire(KM_A)
        jusqu'à KM_A >   KM_D  ET KM_A  <= 1216
 si Type_V = "V_LG"
   alors
      Montant_P ← (KM_A – KM_D) * 1.5
   sinon si Type_V = "V_TV"
              alors
                 Montant_P ← (KM_A – KM_D) * 3
              sinon
                 Montant_P ← (KM_A – KM_D) * 5
         fsi
 fsi
 écrire ("le montant à payer est : ", Montant_P)
fin
```

مناقشة الحل:

الخوارزمية أعلاه تحسب المبلغ الواجب دفعه من طرف صاحب مركبة استعملت الطريق السيار شرق–غرب، تبدأ الخوارزمية بطلب إدخال نوع المركبة والذي يجب أن يكون ضمن الثلاثة الفئات المعتمدة من طرف إدارة الطرق السريعة، ثم بعد تحديد فئة المركبة يتم طلب إدخال كلم البدء والذي يجب أن يكون ضمن المجال [1 , 1216]، بعد ذلك يتم

إدخال كلم الوصول والذي بدوره يجب أن يكون ضمن المجال [KM_D, 1216]، بحيث لا يجب أن يكون كلم الوصول أصغر أو يساوي كلم الانطلاق، ثم في الأخير تقوم الخوارزمية بعملية حسابية بسيطة تحسب من خلالها المبلغ الواجب دفعه من طرف صاحب المركبة.

2)

```
Algorithme Autoroute_Est_Ouest_2
  var  KM_D, KM_A, NB_V, i : entier
       Montant_P, Recette : réel
       Type_V : caractère
 début
    répéter
    écrire("Donnez le nombre de véhicule")
          lire(NB_V)
         jusqu'à NB_V > 0
    Recette ← 0
    pour i ← 1 à NB_V
    répéter
    écrire("Donnez le type du véhicule")
          lire(Type_V)
         jusqu'à Type_V = "V_LG" OU Type_V = "V_TV"  OU  Type_V = "V_PL"
    répéter
    écrire("Donnez le kilomètre du départ")
          lire(KM_D)
         jusqu'à KM_D > =  0 ET KM_D  < 1216
   répéter
    écrire("Donnez le kilomètre d'arrivée")
          lire(KM_A)
         jusqu'à KM_A >   KM_D  ET KM_A  <= 1216
  si Type_V = "V_LG"
     alors
        Montant_P ← (KM_A – KM_D) * 1.5
    sinon si Type_V = "V_TV"
              alors
                 Montant_P ← (KM_A – KM_D) * 3
              sinon
                 Montant_P ← (KM_A – KM_D) * 5
          fsi
  fsi
  Recette ← Recette + Montant_P
  fin pour
  écrire ("le montant de la recette journalière est : ", Recette)
fin
```

مناقشة الحل:

الخوارزمية أعلاه تحسب مجموع المبالغ المحصلة في نقطة من نقاط المراقبة، ولتحقيق ذلك يجب معرفة عدد السيارات التي عبرت هذه النقطة (NB_V)، ثم بعد معرفة هذا العدد وبواسطة حلقة pour تقوم هذه الخوارزمية بجمع المبالغ المدفوعة من طرف أصحاب الواحد تلو الآخر بدءا من أول مركبة حتى آخر واحدة.

V. المسألة الخامسة:

تقدم روضة الأطفال قصر البراءة خدماتها لزبائنها الدائمين والمؤقتين حسب المعطيات التالية :

- بالنسبة للزبائن الدائمين

➢ الأطفال فئة الرضع (العمر من سنة واحدة إلى سنتين أي 1 ≤ العمر < 2) يدفع الزبون مبلغا جزافيا قدره 300 دج لليوم.

➢ الأطفال فئة التحضيري1 (العمر من سنتين إلى 4 سنوات أي 2 ≤ العمر < 4) يدفع الزبون مبلغا جزافيا قدره 400 دج لليوم.

➢ الأطفال فئة التحضيري2 (العمر من 4 سنوات إلى 6 سنوات أي 4 ≤ العمر < 6) يدفع الزبون مبلغا جزافيا قدره 500 دج لليوم.

- بالنسبة للزبائن المؤقتين تقدم الروضة خدمة حضانة الأطفال مهما كان عمر الطفل ويدفع الزبون مبلغا قدره 1000 دج لليوم.

تفتح الروضة أبوابها على الساعة 07h:00 وتغلق على الساعة 17h :00 مع الإبقاء على خدمة استثنائية من الساعة 17h :00 إلى الساعة 20 h :00.

يدفع كل زبون دائم إضافة على المبلغ الجزافي مبلغ 100 دج عن كل ساعة تأخر بعد الساعة 17h :00 ، ويدفع كل زبون مؤقت إضافة على المبلغ المدفوع مبلغ 200 دج عن كل ساعة تأخر بعد الساعة 17h :00.

يريد مدير الروضة حساب إيراداته اليومية فيطلب منك المساعدة بكاتبة خوارزمية تقوم بذلك.

استعمل المتغيرات التالية:

عمر الطفل: Age_enf، ساعة استرجاع الطفل: H_recup، عدد الساعات الإضافية (بعد الساعة 17h :00): Nb_h_sup ، المبلغ المدفوع من طرف الزبون : Montant_paye ، العدد الكلي للأطفال: Nb_enf ، نوع الزبون : Type_cl (زبون دائم Type_cl = 'A' زبون مؤقت Type_cl = 'P')، إيرادات الروضة اليومية: Recette.

حل المسألة الخامسة:

```
Algorithme Crèche
var  Age_enf, H_recup, Nb_h_sup, Montant_paye, Nb_enf, Recette , i : entier
     Type_cl : caractère
début
   répéter
   écrire("Donnez le nombre total des enfants" )
   lire(Nb_enf)
  jusqu'à Nb_enf  >= 0
  Recette ← 0
  pour  i ← 1 à Nb_enf
```

```
faire
   répéter
     écrire("Donnez le type de client  A :  Abonné,  P :  de  passage" )
     lire(Type_cl )
   jusqu'à (Type_cl  = "A")  OU  (Type_cl  = "P")
   répéter
    écrire("Donnez l'heure de récupération de l'enfant")
    lire(H_recup)
   jusqu'à (H_recup >= 7)  ET (H_recup <= 20)
  si   Type_cl  = "A"
    alors
        répéter
         écrire("Donnez l'âge de l'enfant")
         lire(Age_enf)
        jusqu'à (Age_enf >=1) et (Age_enf <6)
       si H_recup <= 17
         alors
              si  Age_enf <2
                alors
                     Montant_paye ←300
                sinon si  Age_enf <4
                           alors
                                Montant_paye ←400
                           sinon
                                Montant_paye ←500
                          fsi
              fsi
         sinon
         Nb_h_sup ← H_recup – 17
          si  Age_enf <2
           alors
              Montant_paye ←300  +  Nb_h_sup*100
           sinon si  Age_enf <4
                    alors
                        Montant_paye ←400  +  Nb_h_sup*100
                    sinon
                        Montant_paye ←500  +  Nb_h_sup*100
                   fsi
          fsi
      fsi
   sinon  si H_recup <= 17
             alors
                Montant_paye ←1000
             sinon
               Nb_h_sup ← H_recup – 17
               Montant_paye ←1000  +  Nb_h_sup*200
          fsi
  fsi
   Recette ← Recette + Montant_paye
  fin pour
  écrire( "La recette journalière de la crèche est : " , Recette)
fin
```

مناقشة الحل:

تقوم الخوارزمية أعلاه بحساب الإيرادات اليومية لروضة أطفال تقدم خدماتها لفئتين من الزبائن، الزبائن الدائمين (client Abonné) والزبائن المؤقتين (client de passage)، من أجل هذا قسمت الخوارزمية بواسطة تعليمة شرطية متناوبة إلى قسمين رئيسيين حسب نوع الزبون (Type_cl) بحيث إذا كان ("A" = Type_cl) فإن الزبون هو زبون دائم أما إذا كان ("P" = Type_cl) فالزبون هو زبون مؤقت، وتكفلت الحلقة répéter jusqu'à بإرغام مستخدم هذه الخوارزمية على إدخال نوع الزبون الصحيح، بحيث كل المعالجة التي تأتي بعد هذه الحلقة تتعلق بنوع الزبون، ففي حالة الزبون الدائم يحسب المبلغ الواجب دفعه حسب المعطيات المقدمة في نص المسألة، ويجب على الخوارزمية معرفة عمر الطفل وساعة استرجاعه، هذه الأخيرة يجب أن تكون ضمن المجال [20 , 7] وفي حالة كانت أكبر من 17 يحسب ثمن إضافي حسب عدد الساعات بعد الساعة 17، أما في حالة الزبون المؤقت، لا يُطلب من مستخدم الخوارزمية إدخال عمر الطفل، ونكتفي بطلب ساعة الاسترجاع حتى يحسب المبلغ الواجب دفعه من طرف الزبون.

الجدير بالذكر أن هذه الخوارزمية احتوت على تعليمتين شرطيتين متداخلتين تتعلقان بعمر الطفل، حسب المجالات الثلاثة التالية:

المجال الأول: [1, 2[، هذا المجال تم التأكد منه من خلال المقارنة (Age_enf < 2) بحيث لا داعي لإضافة الجزء (Age_enf >= 1) بحيث تكفلت الحلقة répéter jusqu'à بحصر عمر الطفل داخل المجال [1, 6[.

المجال الثاني: [2, 4[، تم التأكد من هذا المجال من خلال المقارنة (Age_enf < 4) بحيث لا داعي لإضافة الجزء (Age_enf >= 2) لأنه داخل في نفي المقارنة (Age_enf < 2).

المجال الثالث: [4, 6[، تم التأكد من هذا المجال من خلال نفي المقارنة (Age_enf < 4) أي (Age_enf >= 4) أما الجزء الثاني من المقارنة (Age_enf < 6) فهو مضمون بالمجال السابق [1, 6[والذي تكفلت به الحلقة répéter jusqu'à.

في الأخير قامت الخوارزمية السابقة بحساب إيرادات هذه الروضة (Recette)، بحيث استخدمت حلقة pour لجمع المبالغ المدفوعة من طرف الزبائن، الدائمين والمؤقتين على حد سواء، فتبدأ الحلقة من الزبون رقم 1 ثم الذي يليه، وهكذا حتى يتم تعداد كل الزبائن، ونحصل في الأخير على مجموع المبالغ التي دفعت وهي تمثل الإيرادات اليومية.

VI. المسألة السادسة:

تقوم عيادة ابن سينا المختصة في جراحة القلب والأوعية الدموية بثلاث أنواع من العمليات الجراحية: (OP1, OP2, OP3) لفائدة المرضى الذين يعانون من مشاكل قلبية. تكاليف إجراء عملية جراحية واحدة تحسب كالآتي :

$$Frais_Op = Prix_Op + Nb_Jr \times Frais_Sejour$$

حيث أن Prix_Op وهو ثمن العملية الجراحية والذي يتعلق بنوع العملية الجراحية (Type_Op) معطى كالآتي :

إذا كان Type_Op = Op1 فإن Prix_Op = 100 000,00 DA؛

وإذا كان Type_Op = Op2 فإن Prix_Op = 250 000,00 DA؛

أما إذا كان Type_Op = Op3 فإن Prix_Op = 500 000,00 DA.

Fr_Sejour تمثل تكاليف مكوث يوم واحد في العيادة وتحسب بالعلاقة التالية :

Frais_Sejour = Prix_Op * 0.02.

Nb_Jr تعبر عن عدد أيام مكوث المريض في العيادة قبل وبعد إجراء العملية الجراحية.

1) كُلفتم من طرف مدير هذه العيادة بكتابة خوارزمية تحسب تكاليف عملية جراحية أجريت لمريض ما. يجب على الخوارزمية معرفة نوع العملية الجراحية (Type_Op)، وعدد الأيام التي قضاها هذا المريض في العيادة (Nb_Jr).

2) تلقت العيادة تعليمة من وزارة الصحة تلزمها بتطبيق تخفيض قدره 50% من ثمن العملية الجراحية (Prix_Op) على الأطفال (المرضى الذين تقل أعمارهم عن 16 سنة) هذا التخفيض تتكفل به الدولة ويتم تعويضه للعيادة في آخر السنة. أعد كتابة الخوارزمية السابقة لتأخذ فيها بعين الاعتبار هذه التعليمة. يجب على الخوارزمية معرفة عمر المريض (Age_Patient).

3) في نهاية السنة كلفكم المدير بمساعدة محاسب العيادة لحساب الإيرادات السنوية (Recette) وكذا مبلغ التخفيض الإجمالي (MT_reduction) (بعد تطبيق التعليمة الوزارية) وذلك بكتابة خوارزمية تقوم بذلك. يجب على الخوارزمية معرفة العدد الكلي للمرضى الذين أجريت لهم عملية جراحية خلال السنة (Nb_Patient)

حل المسألة السادسة:

1)

```
Algorithme  clinique
var   frais_op, prix_op, frais_sejour : réel
      nb_jours_sejour : entier
      Type_op : caractère
début
        répéter
         écrire("Donnez le type de l'opération")
          lire(Type_op)
          jusqu'à  (Type_op =  "op1")  OU  (Type_op  =  "op2")  OU  (Type_op = "op3")
        répéter
          écrire("Donnez le nombre de jours  passé  à la clinique" )
          lire( nb_jours_sejour)
          jusqu'à  nb_jours_sejour  >  0
         si type_op = "op1"
              alors
                    prix_op ← 100000.00
              sinon si type_op = "op2"
                          alors
                             prix_op ← 250000.00
                          sinon
                            prix_op ← 500000.00
                       fsi
         fsi
```

```
        frais_sejour ← prix_op * 0.02
        frais_op ← prix_op + nb_jours_sejour * frais_sejour
        écrire (" les frais de l' opération pour ce malade sont : " , frais_op)
fin
```

مناقشة الحل:

تقوم الخوارزمية أعلاه بحساب تكاليف عملية جراحية حسب المعطيات التي جاءت في نص المسألة، بحيث تبدأ الخوارزمية بطلب إدخال نوع العملية الجراحية، وبعد معرفة نوع العملية الجراحية وعدد الأيام التي قضاها المريض داخل العيادة تحسب الخوارزمية تكاليف العملية الجراحية باستعمال العلاقة الحسابية البسيطة التي جاءت في نص المسألة، وباستعمال تعليمة شرطية متداخلة شرطها هو الأنواع الثلاثة للعملية الجراحية.

2)

```
Algorithme  clinique
var frais_op, prix_op, frais_sejour : réel
   nb_jours_sejour, age_patient : entier
     Type_op : caractère
 début
        répéter
          écrire( "Donnez le type d'opération" )
           lire(Type_op)
        jusqu'à  (Type_op =  "op1")   OU  (Type_op  =  "op2")  OU (Type_op = "op3")
        répéter
           écrire( "Donnez le nombre de jours  passé  à la clinique" )
           lire( nb_jours_sejour)
        jusqu'à  nb_jours_sejour  >  0
       répéter
          écrire( "Donnez l'âge du patient " )
           lire(age_patient )
          jusqu'à  age_patient  > 0
          si Type_op = "op1"
                alors
                      prix_op ← 100000.00
                sinon si Type_op = "op2"
                            alors
                                  prix_op ← 250000.00
                            sinon  prix_op ← 500000.00
                       fsi
          fsi
         frais_sejour ← prix_op*0.02
         si age_patient < 16
              alors
                  prix_op ← prix_op * 0.5
         fsi
                   frais_op ← prix_op + nb_jours_sejour*frais_sejour
         écrire ("les frais de l'opération pour ce malade est : " , frais_op)
fin
```

مناقشة الحل:

الخوارزمية أعلاه تشبه إلى حد كبير الخوارزمية السابقة، ويكمن وجه الاختلاف بينهما في زيادة الجزء المتعلق بعمر المريض، بحيث عندما يكون سن المريض أصغر تماما من 16 سنة يطبق على ثمن العملية الجراحية التي أجريت له تخفيض 50%، بينما يبقى حساب ثمن المكوث في العيادة من دون تغيير.

3)

```
Algorithme  clinique
var frais_op, prix_op, frais_sejour, Recette, M_reduction : réel
    Nb_Patient , nb_jours_sejour, age_patient  i : entier
    Type_op : caractère
 début
        Recette ← 0
        MT_reduction ← 0
      répéter
         écrire( "Donnez le nombre global  des patients" )
         lire(Nb_Patient)
       jusqu'à  Nb_Patient  >  0
i ← 1
tant que  i  < =  Nb_Patient
  faire
        répéter
          écrire( "Donnez le type d'opération" )
          lire(Type_op)
         jusqu'à  (Type_op =  "op1")   OU  (Type_op  =  "op2")  OU (Type_op = "op3")
        répéter
          écrire("Donnez le nombre de jours  passé  à la clinique")
          lire( nb_jours_sejour)
        jusqu'à  nb_jours_sejour  >  0
       répéter
         écrire( "Donnez l'âge du patient " )
         lire(age_patient )
        jusqu'à  age_patient  > 0
         si Type_op = "op1"
               alors
                     prix_op ← 100000.00
               sinon si Type_op = "op2"
                           alors
                                 prix_op ← 250000.00
                           sinon  prix_op ← 500000.00
                      fsi
         fsi
         frais_sejour ← prix_op*0.02
        si age_patient < 16
             alors
                prix_op ← prix_op * 0.5
                MT_reduction ←   MT_reduction   +  prix_op
         fsi
         frais_op ← prix_op + nb_jours_sejour * frais_sejour
         Recette ← Recette + frais_op
```

```
        i ← i+1
    fin tant que
            écrire ("La recette annuelle de la clinique  est : " , Recette)
            écrire ("Le  montant global de la réduction est : " , MT_reduction   )
fin
```

مناقشة الحل:

تقوم الخوارزمية السابقة بحساب الإيرادات السنوية للعيادة الطبية، كما تقوم هذه الخوارزمية بحساب مجموع التخفيضات التي قدمتها العيادة لفائدة المرضى الأطفال، وللقيام بهذه المهمة استعملت الخوارزمية تعليمة تكرارية تمثلت في الحلقة tant que، وبدأت بتعداد المرضى الذين أجريت لهم عملية جراحية خلال السنة (Nb_Patient)، قامت الخوارزمية بحساب مجموع مبالغ تكاليف العمليات الجراحية التي أجريت للمرضى خلال السنة وهو ما يمثل إيرادات العيادة (Recette) كما قامت بحساب مجموع التخفيضات (MT_Reduction) من خلال تعليمة شرطية تعليمة شرطية بسيطة شرطها هو عمر المريض.

VII. المسألة السابعة:

يقدم مطعم الطيبات لزبائنه مجموعة من الأطعمة والخدمات حسب مستويين:

المستوى الأول : وهو المستوى الاقتصادي (Type_Niveau = Niveau_Eco) حيث يدخل الزبون إلى القاعة1؛

المستوى الثاني : وهو مستوى رجال الأعمال (Type_Niveau = Niveau_Aff) حيث يدخل الزبون إلى القاعة2.

تقدم للزبون قائمة الأطعمة المتوفرة وهي المقبلات والأطباق الرئيسية.

في القاعة1 يوجد نوع واحد من المقبلات وهو: "H_d11" = type_Hd وسعره Prix_Hd = 250 DA، أما الأطباق الرئيسية فمتوفرة بنوعين وهي: "Plat11" = Type_Plat وسعره Prix_Plat = 850 DA، "Plat12" = Type_Plat وسعره Prix_Plat = 1 250 DA؛

في القاعة2 يوجد نوعان من المقبلات وهي: 'H_d21' = type_Hd وسعره Prix_Hd = 950 DA، 'H_d22' = type_Hd وسعره Prix_Hd = 1 800 DA؛ أما الأطباق الرئيسية فمتوفرة بنوعين وهي: 'Plat21'=Type_Plat وسعره Prix_Plat =6 500 DA، 'Plat22'= Type_Plat وسعره Prix_Plat = 9 500 DA.

1) أكتب خوارزمية تقوم بمساعدة موظف التحصيل على حساب المبلغ الواجب دفعه من طرف زبون ما (Montant_paye)، بعد معرفة المستوى المختار (Type_Niveau = "Niveau_Eco" ou "Niveau_Aff") ونوع المقبلات type_Hd و نوع الأطباق الرئيسية Type_Plat.

2) يريد صاحب المطعم حساب عدد الزبائن من المستويين (Niveau_Eco et Niveau_Aff) ؛ وكذا إيرادات كل مستوى، أكتب خوارزمية تساعده على ذلك، يجب على الخوارزمية معرفة العدد الكلي للزبائن Nb_Cli ، عدد الزبائن في

المستوى الاقتصادي Nb_Eco ومستوى الأعمال Nb_Aff ، إيرادات المستوى الاقتصادي Recette_Eco ومستوى الأعمال Recette_Aff.

حل المسألة السابعة:

1)

```
Algorithme Restau_1
var  Montant_paye, Prix_Hd, Prix_Plat : entier
     Type_Niveau,  Type_Hd, Type_Plat: caractère
début
          répéter
          écrire( "Donnez le Niveau choisi" )
          lire(Type_Niveau)
          jusqu'à Type_Niveau = "Niveau_Eco" OU Type_Niveau = "Niveau_Aff"
          si    Type_Niveau = "Niveau_Eco"
            alors
                répéter
                écrire( ''Donnez le type de l'hors d'œuvre'' )
                lire(Type_Hd)
                jusqu'à Type_Hd = "Hd11"
                Prix_Hd ← 250
                Répéter
                écrire( "Donnez le type du plat principal" )
                lire(Type_Plat)
                jusqu'à Type_Hd = "Plat11" OU Type_Plat = "Plat12"
                    si   Type_Plat = "Plat11"
                       alors
                            Prix_Plat ← 850
                       sinon
                            Prix_Plat ← 1250
                    fsi
            sinon
                répéter
                écrire( "Donnez le type de l'hors d'œuvre" )
                lire(Type_Hd)
                jusqu'à Type_Hd = "Hd21"  OU  Type_Hd = "Hd22"
             si   Type_ Hd = "Hd21"
               alors
                    Prix_Hd ← 950
               sinon
                    Prix_Hd ← 1800
             fsi
             répéter
             écrire( "Donnez le type du plat principal" )
             lire(Type_Plat)
             jusqu'à Type_Hd = "Plat21" OU  Type_Plat = "Plat22"
             si   Type_Plat = "Plat21"
               alors
                  Prix_Plat ← 6500
               sinon
                  Prix_Plat ← 9500
             fsi
```

```
        fsi
         Montant_paye ← Prix_Hd +  Prix_Plat
        écrire( "Le montant payé par le client est : " , Montant_paye)
fin
```

مناقشة الحل:

تقوم الخوارزمية أعلاه بحساب المبلغ الواجب دفعه من طرف زبون بالمطعم المذكور في المسألة السابقة، عند تقدم الزبون إلى موظف التحصيل، والذي يستعمل الخوارزمية السابقة، يجب تزويده بكل المعلومات اللازمة لحساب ثمن المأكولات، لذا فأول ما تطلبه الخوارزمية هو المستوى المختار (Type_Niveau = "Niveau_Eco" ou "Niveau_Aff")، وبالتالي حسب المستوى المختار تطلب الخوارزمية نوع المقبلات ونوع الطبق الرئيسي، ولحصر النتائج المدخلة من طرف المستخدم تم الاستعانة بتعليمة تكرارية تمثلت في الحلقة répéter jusqu'à، بعد معرفة نوع المقبلات ونوع الطبق الرئيسي يتم حساب المبلغ الواجب دفعه عن طريق عملية حسابية بسيطة، ثم يتم عرض النتيجة على الشاشة.

2)

```
Algorithme Restau_2
var Nb_Cli, Nb_Eco, Nb_Aff, Recette_Eco, Recette_Aff,  Montant_paye, Prix_Hd, Prix_Plat, i : entier
      Type_Niveau,  Type_Hd, Type_Plat: caractère
 début
      répéter
      écrire( "Donnez le nombre total de clients" )
      lire(Nb_Cli)
      jusqu'à Nb_Cli  > 0
      i ← 1
      Nb_Eco ←0
      Nb_Aff ←0
      Recette_Eco← 0
      Recette_Aff ← 0
      tant que i <= Nb_Cli
       faire
            répéter
            écrire( "Donnez le Niveau choisi" )
            lire(Type_Niveau)
             jusqu'à Type_Niveau = "Niveau_Eco" OU  Type_Niveau = "Niveau_Aff"
            si    Type_Niveau = "Niveau_Eco"
               alors
                     répéter
                     écrire( "Donnez le type de l'hors d'œuvre" )
                     lire(Type_Hd)
                     Jusqu'à Type_Hd = "Hd11"
                     Prix_Hd ← 250
                     répéter
                     écrire( "Donnez le type du plat principal" )
                     lire(Type_Plat)
                     jusqu'à Type_Hd = "Plat11" OU  Type_Plat = "Plat12"
                              si   Type_Plat = "Plat1"
                                 alors
```

```
                                Prix_Plat ← 850
                            sinon
                                Prix_Plat ← 1250
                        fsi
                    Nb_Eco ← Nb_Eco + 1
                    Montant_paye ← Prix_Hd +  Prix_Plat
                    Recette_Eco  ← Recette_Eco +  Montant_paye
            sinon
                répéter
                écrire( "Donnez le type de l'hors d'œuvre" )
                lire(Type_Hd)
                jusqu'à Type_Hd = "Hd21"  OU  Type_Hd = "Hd22"
               si   Type_ Hd = "Hd 1"
                            alors
                                Prix_Hd ← 950
                            sinon
                                Prix_Hd ← 1800
               fsi
                répéter
                écrire( "Donnez le type du plat principal" )
                lire(Type_Plat)
                jusqu'à Type_Hd = "Plat21" OU  Type_Plat = "Plat22"
                        si   Type_Plat = "Plat1"
                            alors
                                Prix_Plat ← 6500
                            sinon
                                Prix_Plat ← 9500
                        fsi
                    Nb_Aff ← Nb_Aff + 1
                    Montant_paye ← Prix_Hd +  Prix_Plat
                    Recette_Aff  ← Recette_Aff +  Montant_paye
        fsi
       i ← i +1
    fin tantque
  écrire( "Le nombre de clients du niveau économique  est : " , Nb_Eco)
  écrire( "Le montant payé par ces clients est : " , Recette_Eco  )
  écrire( "Le nombre de clients du niveau affaire  est : " , Nb_Aff)
  écrire( "Le montant payé par ces clients est : " , Recette_Aff )
fin
```

مناقشة الحل:

تقوم الخوارزمية أعلاه، بحساب الإيرادات اليومية للمطعم حسب نوع المستوى وكذا عدد الزبائن حسب كل مستوى، بعد معرفة العدد الكلي للزبائن الذين ترددوا على المطعم، تستعمل الخوارزمية الحلقة tant que لتعداد الزبائن الواحد تلو الآخر، بحيث عندما يكون الزبون من المستوى الاقتصادي تحسب المبلغ الواجب دفعه وتضيفه إلى الإيرادات حسب المستوى الاقتصادي، كما تضيف 1 إلى عدد الزبائن من المستوى الاقتصادي، وتفعل نفس الشيء مع الزبائن من المستوى رجال الأعمال، وعند انتهاء عد كل الزبائن تخرج الخوارزمية بالنتائج المتعلقة بـ:

- عدد الزبائن من المستويين (Niveau_Eco et Niveau_Aff) عدد الزبائن في المستوى الاقتصادي Nb_Eco ومستوى الأعمال Nb_Aff؛
- وكذا إيرادات كل مستوى، Recette_Eco وRecette_Aff، وفي الأخير تعرض الخوارزمية هذه النتائج على الشاشة.

VIII. المسألة الثامنة:

ينتج مصنع السيارات DZWagen نوعين من السيارات T1 وT2 وكل نوع متوفر بصنفين: الصنف العادي T1N و T2N وسعرهما على التوالي P1 و P2 وصنف ذوي الاحتياجات الخاصة T1H و T2H وسعرهما على التوالي P3 وP4، يسوق هذا المصنع منتوجه عبر مجموعة من الوكالات. بحيث كل وكالة تشتري من المصنع عددا معينا من السيارات يوميا لتبيعها إلى زبائنها حسب الطلب.

نأخذ على سبيل المثال:

P1 = 1 500 000 DA, P2 = 2 200 000 DA, P3 = 1 800 000 DA et P4 = 2 600 000 DA

1) أكتب خوارزمية تحسب وتعرض على الشاشة إيرادات المصنع اليومية (Recette) بعد معرفة عدد الوكالات التي اشترت في هذا اليوم (Nb_Ag) وعدد السيارات التي اشترتها كل وكالة Nb_V منها Nb_T1N و Nb_T1H وNb_T2N وNb_T2H وهو يمثل عدد السيارات حسب كل صنف، ومبلغ مشتريات كل وكالة (Prx_Ag) وهو يمثل عدد كل صنف مضروب في سعر كل صنف.

2) تلقى المصنع تعليمة من وزارة الصناعة تجبره على تطبيق تخفيض قدره 40% على سيارات ذوي الاحتياجات الخاصة على أن يتم تعويضه من طرف الدولة يوميا بعد إعلامها بمبلغ التخفيض اليومي (مجموع كل مبالغ التخفيض في اليوم)، أكتب الجزء المتغير من الخوارزمية (لا تعيد كتابة كل الخوارزمية) والذي يحسب الإيرادات اليومية للمصنع و مجموع مبالغ التخفيض اليومي MT_Rd

حل المسألة الثامنة:

```
Algorithme DZWagen_1
const  P1 = 1500000
         P2 = 2200000
         P3 = 1800000
         P4 = 2600000
var   Nb_V, Nb_Ag, Nb_T1N, Nb_T1H, Nb_T2N, Nb_T2H, Prx_Ag,
       Recette,  i : entier
début
   répéter
   écrire( "Donnez le nombre des Agences" )
     lire(Nb_Ag)
     jusqu'à Nb_Ag > 0
    i ←1
    Recette ← 0
```

```
tant que i <= Nb_Ag
  faire
    répéter
     écrire( "Donnez le nombre de véhicules achetés" )
     lire(Nb_V)
    jusqu'à Nb_V > 0
    répéter
    écrire( "Donnez le nombre de véhicules de type T1N " )
    lire(Nb_T1N)
    jusqu'à Nb_T1N < = Nb_V  ET Nb_T1N > = 0
    répéter
    écrire( "Donnez le nombre de véhicules de type T1H " )
    lire(Nb_T1H)
    jusqu'à Nb_T1H < = Nb_V - Nb_T1N  ET Nb_T1H > = 0
    répéter
    écrire( "Donnez le nombre de véhicules de type T2N " )
    lire(Nb_T2N)
    jusqu'à Nb_T2N < = Nb_V - Nb_T1N - Nb_T1H  ET Nb_T2N > = 0
    Nb_T2H  ← Nb_V - Nb_T1N - Nb_T1H- Nb_T2N
    Prx_Ag ← Nb_T1N * P1 +  Nb_T2N * P2 +  Nb_T1H * P3 + Nb_T2H * P4
    Recette ← Recette + Prx_Ag
     i ← i + 1
  fin tant que
 écrire( "La recette journalière de l'usine est : " , Recette)
fin
```

مناقشة الحل:

ما يميز هذه الخوارزمية عن سابقاتها من الخوارزميات التي اقتُرِحت لحل المسائل السابقة، هو طريقة تقييد إدخال المعطيات من طرف المستخدم، بحيث بعد إدخال عدد الوكالات التي قامت بعملية الشراء ليوم معين، تقوم الخوارزمية أعلاه، بحساب الإيرادات اليومية للمصنع حسب مشتريات كل وكالة من الوكالات السابقة الذكر، فتبدأ بِعَدِّها واحدة بعد أخرى، وعند معالجة مشتريات كل وكالة، تطلب من المستخدم إدخال عدد السيارات التي اقتنتها الوكالة قيد المعالجة، بحيث يُرغَم المستخدم (باستعمال حلقة répéter jusqu'à) على إدخال عدد موجب تماما من السيارات (NB_V) ثم بعد إدخال العدد الكلي للسيارات المقتناة من طرف الوكالة، تمر الخوارزمية إلى تفصيل عدد السيارات حسب كل صنف من الأصناف الأربعة (Nb_T1N, Nb_T2N, Nb_T1H, Nb_T2H)، فيُطلب من المستخدم إدخال عدد السيارات من الصنف Nb_T1N والذي يجب أن يكون أكبر أو يساوي الصفر (يمكن أن يكون عدد سيارات من صنف معين معدوم بحيث لم تقم الوكالة باقتناء أي سيارة من هذا الصنف)، كما يجب أن يكون عدد هذا الصنف أصغر من العدد الكلي للسيارات المشتراة (Nb_T1N < = NB_V)، ثم بعد معرفة عدد السيارات من الصنف Nb_T1N تمر الخوارزمية إلى طلب إدخال عدد السيارات من الصنف التالي وهو Nb_T2N، والذي يجب أن يكون أكبر أو يساوي الصفر، كما لا يجب أن يتعدى Nb_V - Nb_T1N بحيث يجب أن يساوي مجمع الأصناف المقتناة من طرف الوكالة العدد الكلي للسيارات التي اشترتها الوكالة (NB_V)، لذا تحرص الخوارزمية على مراقبة المعطيات المدخلة

من طرف المستخدم بكل دقة لاجتناب حدوث أخطاء في حساب المبالغ المالية وإيرادات المصنع اليومية، ثم تواصل الخوارزمية إدخال كل المعطيات الخاصة بالأصناف التي اشترتها الوكالة، بحيث يجب تحقق العلاقة التالية:

$$NB_V = Nb_T1N + Nb_T1H + Nb_T2N + Nb_T2H$$

لذا فعند وصولنا إلى آخر صنف وهو Nb_T2H لا داعي لطلب إدخال عدده من طرف المستخدم فهو يستنتج مباشرة من العلاقة السابقة، بحيث انطلاقا من العلاقة السابقة عدد السيارات من الصنف Nb_T2H تساوي (Nb_T2H = Nb_V - Nb_T1N - Nb_T1H- Nb_T2N) وهو ما تم كتابته داخل الخوارزمية من خلال التعليمة:

Nb_T2H ← Nb_V - Nb_T1N - Nb_T1H- Nb_T2N.

في الأخير، وبعد معرفة أعداد الأصناف الأربعة من السيارات التي اشترتها الوكالة، تقوم الخوارزمية بحساب مبلغ مشتريات الوكالة بضرب عدد سيارات كل صنف في سعر هذا الصنف، ومن ثم إضافة هذا المبلغ إلى المجموع الكلي لمشتريات الوكالات الأخرى حتى يتم معالجة كل الوكالات والحصول على الإيرادات اليومية لمصنع DZWagan، ويتم عرض هذه الإيرادات على الشاشة.

```
Algorithme DZWagen_2
 var   Rd, MT_Rd  : entier
 début
     MT_Rd ← 0
    Prx_Ag ← Nb_T1N*P1 +  Nb_T2N*P2 +  Nb_T1H*P3*0.6 + Nb_T2H*P4*0.6
    Rd ← Nb_T1H*P3*0.4 + Nb_T2H*P4*0.4
    Recette ← Recette + Prx_Ag
    MT_Rd  ← MT_Rd + Rd
fin
```

مناقشة الحل:

بعد تطبيق التخفيض، يكفي إضافة المتغيرتين الجديدتين والمتعلقتين بقيمة التخفيض عن السيارات الخاصة بفئة ذوي الاحتياجات الخاصة ومجموع هذه التخفيضات (Rd, MT_Rd) ، ثم يحسب المبلغ الواجب دفعه من طرف الوكالة اعتمادا على هذا التخفيض، كما تحسب إيرادات المصنع ومجموع مبالغ التخفيض باستعمال حلقة tant que تقوم بتعداد كل الوكالات التي قامت بالشراء في هذا اليوم.

IX. المسألة التاسعة:

تقدم وكالة الأبرار للسياحة والأسفار خدماتها لزبائنها في رحلات سياحية لمدة أسبوع حسب المعطيات التالية:

الوجهة الأولى Dest1: تحتوي على فندقين، Hotel11 وHotel12 وتكاليف الإقامة بهما هي على التوالي: 48 000,00 دج و75 000,00 دج.

الوجهة الثانية Dest2: وتحتوي هي كذلك على فندقين Hotel21 و Hotel22 وتكاليف الإقامة بهما هي على التوالي: 65 000,00 دج و 95 000,00 دج.

ثمن تذكرة السفر يحسب كالآتي:

الوجهة الأولى Dest1: الفئة 1er classe ثمنها 55 000,00 دج أما الفئة 2eme classe فثمنها 42 000,00 دج.

الوجهة الثانية Dest2: الفئة 1er classe ثمنها 64 000,00 دج أما الفئة 2eme classe فثمنها 50 000,00 دج.

يحصل الزبون الذي يقل عمره عن 18 سنة على تخفيض قدره 30% من ثمن الإقامة في الفندق وتخفيض قدره 20% من ثمن التذكرة

1) أكتب خوارزمية تحسب و تعرض على الشاشة المبلغ الواجب دفعه إلى الوكالة Montant_G بالنسبة إلى زبون معين، بعد معرفة عمره: Age_Voy وجهته : Dest = Dest1 ou Dest2 ثم الفندق المختار: Hotel، فئة التذكرة : Classe_Billet = classe1 ou classe2، ثمن التذكرة Prix_Billet و ثمن الفندق: Prix_Hotel. المبلغ الواجب دفعه إلى الوكالة Montant_G هو مجموع ثمن التذكرة و ثمن الفندق.

2) أعد كتابة الخوارزمية لحساب المبلغ الكلي الواجب دفعه إلى الوكالة بالنسبة لـ N زبون (N>=2) ، استعمل المتغير Montant_G_N.

حل المسألة التاسعة:

```
Algorithme El_Abrar_1
var  Age_voy, Montant_G, Prix_Billet : entier
       Dest, Hotel, Classe_Billet : caractère
début
        répéter
          écrire("Donner l'âge du client")
          lire(Age_voy)
        jusqu'à Age_voy > 0
        répéter
        écrire("Donner la classe du billet")
        lire(Classe_Billet)
        jusqu'à Classe_Billet =  "classe1" OU Classe_Billet = "classe2"
        répéter
          écrire("Donner la destination du client")
          lire(Dest)
        jusqu'à Dest = "Dest1" OU Dest = "Dest2"
       si Dest = " Dest1"
         alors
             répéter
             écrire("Donner le nom de l'hôtel")
              lire(Hotel)
              jusqu'à Hotel = "Hotel11" OU  Hotel = "Hotel12"
             si Hotel = "Hotel11"
                alors
                     Prix_hotel ← 48000
                sinon
                     Prix_hotel ← 75000
            fsi
                 si Classe_Billet = "classe1"
```

```
                    alors
                        Prix_Billet ← 55000
                    sinon
                      Prix_Billet ← 42000
                   fsi
        sinon
            répéter
            écrire("Donner le nom de l'hôtel")
             lire(Hotel)
             jusqu'à Hotel = "Hotel21" OU  Hotel = "Hotel22"
            si Hotel = "Hotel21"
               alors
                    Prix_hotel ← 65000
               sinon
                    Prix_hotel ← 95000
              fsi
               si Classe_Billet = "classe1"
                    alors
                       Prix_Billet ← 64000
                   sinon
                     Prix_Billet ← 50000
                  fsi
          si Age-voy < 18
              alors
                  Prix_hotel ← Prix_hotel * 0.7
                  Prix_Billet ← Prix_Billet * 0.8
          fsi
    fsi
       Montant_G ← Prix_hotel + Prix_Billet
      écrire("Le montant net à payer est :", Montant_G)
fin
```

مناقشة الحل:

قد تبدو الخوارزمية السابقة للوهلة الأولى أنها طويلة ومعقدة، إلا أنها في الواقع بسيطة جدا وسهلة الكتابة، وتكمن البساطة والسهولة في هذه الخوارزمية في احتوائها على مجموعة من التعليمات الشرطية المتناوبة بحيث كل المعطيات جاءت على وجهين اثنين فقط: وجهتين للسفر، فندقين في كل وجهة، فئتين من التذاكر، تخفيض واحد يخص العمر أكبر من 18 وأصغر من 18، وكل هذه المعطيات تمثل مكمن البساطة والسهولة في هذه المسألة، ولو اتسمت بالتعقيد لكانت مكونة من مجموعة التعليمات الشرطية المتداخلة والمتعددة الشروط.

بحيث بدأت الخوارزمية بطلب إعطاء عمر المسافر وفئة التذكرة التي اقتناها ثم يُسأل عن وجهته، وحسب الوجهة المختارة قسمت هذه الخوارزمية إلى جزأين باستعمال تعليمة شرطية متناوبة، ثم تحسب تكاليف الفندق وثمن التذكرة استنادا إلى المعطيات المبينة في المسألة، ويتم حسب عمر المسافر حساب تكاليف الفندق وثمن التذكرة ثم يُجمعان معا ليكونا المبلغ الواجب دفعه إلى الوكالة.

```
Algorithme El_Abrar_2
var  Age_voy, Montant_G, Montant_G_N, N, i,  Prix_Billet : entier
       Dest, Hotel, Classe_Billet : caractère
début
  répéter
   écrire("Donner le nombre de client")
   lire(N)
   jusqu'à N > 0
  pour i←1 à N
  faire
      répéter
        écrire("Donner l'âge du client")
        lire(Age_voy)
      jusqu'à Age_voy > 0
      répéter
      écrire("Donner la classe du billet")
      lire(Classe_Billet)
      jusqu'à Classe_Billet =  "classe1" OU Classe_Billet = "classe2"
      répéter
        écrire("Donner la destination du client")
        lire(Dest)
      jusqu'à Dest = "Dest1" OU Dest = "Dest2"
     si Dest = " Dest1"
       alors
            répéter
            écrire("Donner le nom de l'hôtel")
             lire(Hotel)
             jusqu'à Hotel = "Hotel11" OU  Hotel = "Hotel12"
            si Hotel = "Hotel11"
               alors
                     Prix_hotel ← 48000
               sinon
                     Prix_hotel ← 75000
           fsi
                si Classe_Billet = "classe1"
                   alors
                         Prix_Billet ← 55000
                   sinon
                     Prix_Billet ← 42000
                fsi
       sinon
            répéter
            écrire("Donner le nom de l'hôtel")
             lire(Hotel)
             jusqu'à Hotel = "Hotel21" OU  Hotel = "Hotel22"
            si Hotel = "Hotel21"
               alors
                     Prix_hotel ← 65000
               sinon
                     Prix_hotel ← 95000
           fsi
               si Classe_Billet = "classe1"
```

```
                    alors
                        Prix_Billet ← 64000
                    sinon
                        Prix_Billet ← 50000
                    fsi
                si Age-voy < 18
                    alors
                        Prix_hotel ← Prix_hotel * 0.7
                        Prix_Billet ← Prix_Billet * 0.8
                fsi
            fsi
            Montant_G ← Prix_hotel + Prix_Billet
            Montant_G_N ← Montant_G_N + Montant_G
        fin pour
        écrire("Le montant global net à payer est : ", Montant_G_N)
    fin
```

مناقشة الحل:

هذه الخوارزمية تمثل الحالة العامة للخوارزمية السابقة، بحيث تقوم بحساب مجموع المبالغ المدفوعة من طرف N زبون، وبما أن عدد الزبائن معروف ومعطى، فإن عدد التكرار كذلك سيكون معروفا مبدئيا ومعطى كذلك، لذا فقد تم الاستعانة بحلقة Pour تقوم بجمع كل المبالغ المدفوعة من طرف الزبائن انطلاقا من الزبون رقم 1 إلى الزبون الأخير رقم N.

X. المسألة العاشرة:

قررت وزارة السكن إطلاق برنامج عدل3 وتحسبا لذلك بدأت بعملية إحصاء المكتتبين وتصنيفهم حسب عدد الأولاد فالمكتتب الذي له ثلاثة أولاد فما فوق تمنح له شقة من أربع غرف F4 أما المكتتب الذي له أقل من ثلاثة أولاد تمنح له شقة من ثلاثة غرف F3.

أكتب خوارزمية تطلب من المستخدم إدخال العدد الكلي للمكتتبين NB_SC ثم تطلب منه إدخال عدد الأولاد لكل مكتتب NB_ENF ثم حسب عدد الأولاد تقوم بتصنيفه من فئة F3 أو F4 وتقوم بحساب عدد المكتتبين الذين ستمنح لهم شقق من نوع F4 NB_F4 وعدد المكتتبين الذين ستمنح لهم شقق من نوع F3 NB_F3، ثم تطبع على الشاشة هذين العددين وذلك لكي يتسنى للوزارة إطلاق المشروع الخاص بالبناء حسب نتائج عملية الإحصاء.

ملاحظة: NB_SC = NB_F4 + NB_F3

بعد عدة شكاوي واحتجاجات قام بها المكتتبون الذين لديهم أكثر من ستة أطفال، قامت وزارة السكن بتعديل المشروع بحيث يمنح للمكتتبين الذين لديهم أكثر من خمسة أطفال شقق من خمس غرف F5 بدل الشقق من 4 غرف F4، أعد كتابة خوارزمية تستعمل عدد المكتتبين الذي منحت لهم مبدئيا شقق من فئة F4، ليتم إعادة إحصائهم حسب عدد الأطفال وتنمح لهم شقق فئة F5 أو تثبت لهم الشقق من فئة F4.

حل المسألة العاشرة:

```
Algorithme AADL3_1
var   NB_SC, NB_ENF, NB_F4, NB_F3, i : entier
début
     répéter
       écrire( "Donnez le nombre total des souscripteurs" )
       lire(NB_SC)
      jusqu'à NB_SC >  0
     NB_F3  ← 0
     NB_F4  ← 0
    pour i ← 1 à NB_SC
      faire
        répéter
         écrire( "Donnez le nombre d'enfants pour ce souscripteur" )
         lire(NB_ENF)
        jusqu'à NB_ENF  >= 0
        si NB_ENF < 3
          alors
             NB_F3 ← NB_F3 + 1
          sinon
              NB_F4 ← NB_F4 +1
        fsi
      fin pour
     écrire( "Le nombre d'appartement de type F3 est : " , NB_F3)
     écrire( "Le nombre d'appartement de type F4 est : " , NB_F4)
  fin
```

مناقشة الحل:

تقوم الخوارزمية السابقة بتصنيف المكتتبين وتوزيعهم على مجموعتين رئيسيتين، المجموعة التي ستحصل على شقق من نوع F3 والمجموعة التي ستحصل على شقق من نوع F4. استعملت هذه الخوارزمية تعليمة تكرارية تمثلت في الحلقة pour وهذا راجع إلى المعرفة المسبقة بعدد المكتتبين NB_SC، ثم انطلاقا من معرفة عدد أطفال كل مكتتب، يصنف هذا الأخير إلى مجموعة المكتتبين الذين سيحصلون على F3 أو مجموعة المكتتبين الذين سيحصلون على F4.

في الأخير تقوم الخوارزمية بعرض عدد المكتتبين الذين سيحصلون على F3 وعدد المكتتبين الذين سيحصلون على F4 على الشاشة.

```
Algorithme AADL3_2
var   NB_ENF, NB_F4, NB_F5, i, N : entier
début
     répéter
       écrire( "Donnez le nombre total des souscripteurs ayant obtenue  à priori un F4" )
       lire(NB_F4)
      jusqu'à NB_F4  >  0
     NB_F5  ← 0
     N ← NB_F4
    pour i ← 1 à N
```

```
    faire
      répéter
       écrire( "Donnez le nombre d'enfants pour ce souscripteur" )
       lire(NB_ENF)
      jusqu'à NB_ENF  >= 3
      si NB_ENF > = 5
        alors
           NB_F5 ← NB_F5 + 1
           NB_F4 ← NB_F4 - 1
      fsi
    fin pour
    écrire( "Le nombre d'appartements de type F4 est : " , NB_F4)
    écrire( "Le nombre d'appartements de type F5 est : " , NB_F5)
fin
```

مناقشة الحل:

تقوم الخوارزمية أعلاه بإعادة تصنيف فئة المكتتبين الذين سيحصلون على شقق من نوع F4 بحيث في حالة ما إذا كان للمكتتب أطفال يفوق عدد الخمسة، تمنح له بدل الشقة F4 شقة F5، ولهذا فالخوارزمية في هذه الحالة تضيف واحد إلى عدد الشقق من الصنف F5، وفي المقابل تنقص واحد من عدد الشقق من الصنف F4، وتواصل الخوارزمية معالجة كل المكتتبين الذين كانت ستمنح لهم شقق F4 حتى الانتهاء منهم الواحد تلو الآخر.

وفي الأخير تعرض الخوارزمية على الشاشة عدد المكتتبين الذين سيحصلون على شقق F5 بعد عملية المراجعة هذه، والمكتتبين الذين أبقيت لهم شقق من الصنف F4.

المسألة الحادية عشرة:

قررت إدارة محطة البنزين باب الشرق، إطلاق نظام الدفع الآلي الخاص بالتزود بالوقود، فاتصلت بك لإنشاء خوارزمية تقوم بذلك، بحيث يختار الزبون المدخل "A" للتزود بالبنزين الممتاز "sup" وسعره P1 والبنزين الممتاز بدون رصاص "sup_sp" الذي سعره هو P2، أما المدخل "B" فيختاره الزبون للتزود بالمازوت "gasoil" وسعره P3، بينما يختار الزبون الذي يود التزود بالسيرغاز "sirghaz" المدخل "C"، وسعر اللتر الواحد من السيرغاز هو P4.

1) قم بكاتبة هذه الخوارزمية، بحيث تطلب من الزبون اختيار المدخل (Entrée = "A" ou "B" ou "C") ثم اختيار نوع الوقود (type_car = "sup" ou "sup_sp" ou "gasoil" ou "sirghaz") والكمية Qte المراد التزود بها، فتقوم هذه الخوارزمية بحساب المبلغ الواجب دفعه Prix_car وتعرضه على الشاشة.

نأخذ على سبيل المثال: P1 = 45.50, P2 = 45.20, P3 = 21.75, P4 = 9.35.

2) أكتب خوارزمية تقوم بحساب إيرادات هذه المحطة اليومية Recette إذا علمت أن عدد المركبات التي تزودت بالوقود خلال اليوم هي NB_V.

XI. حل المسألة الحادية عشرة:

```
Algorithme Station_Service1
const  P1 = 45.50
         P2 = 45.20
         P3 = 21.75
         P4 = 9.35
var   Qte: entier
       Prix_car : réel
       Type_car, Entree : caractère
début
      répéter
      écrire( "Donnez la quantité du carburant" )
      lire(Qte)
      jusqu'à Qte > 0
      répéter
      écrire( "Donnez l'entrée choisie" )
      lire(Entree)
      jusqu'à Entree = "A" OU  Entree = "B" OU  Entree = "C"
      si Entree = "A"
        alors
           répéter
           écrire( "Donnez le type de carburant" )
           lire(type_car)
            jusqu'à type_car = "sup" OU  type_car = "sup_sp"
             si type_car =  "sup"
             alors
                 Prix_car ← P1 * Qte
              sinon
                 Prix_car ← P2 * Qte
            fsi
       sinon si  Entree = "B"
                  alors
                       Prix_car ← P3 * Qte
                  sinon
                       Prix_car ← P4 * Qte
              fsi
    fsi
   écrire( "Le montant à payer est : " , Prix_car)
   fin
```

مناقشة الحل:

تقوم الخوارزمية أعلاه بحساب المبلغ الواجب دفعه من طرف الزبون عند تزوده بالوقود، تبدأ الخوارزمية بطلب إدخال كمية الوقود، بحيث كمية الوقود هي معلومة مشتركة بين كل الزبائن، ثم بعد معرفة الكمية، يتم إدخال رمز المدخل (A, B ou C) بحيث كل رمز مدخل متعلق بنوع الوقود الذي سيتم التزود به، بعد اختيار رمز المدخل، يتم اختيار نوع الوقود ثم يتم حساب المبلغ الواجب دفعه، وفي الأخير تقوم الخوارزمية بعرض هذا المبلغ على الشاشة.

```
Algorithme Station_Service2
const  P1 = 45.50
         P2 = 45.20
         P3 = 21.75
         P4 = 9.35
var   Qte, NB_V, i: entier
       Prix_car, Recette : réel
       Type_car, Entree : caractère
début
      répéter
         écrire( "Donnez le nombre véhicule" )
          lire(NB_V)
       jusqu'à NB_V > 0
      Recette ← 0
      Pour i ← 1 à NB_V
        faire
          répéter
          écrire( "Donnez la quantité du carburant" )
          lire(Qte)
           jusqu'à Qte > 0
        répéter
         écrire( "Donnez l'entrée choisie" )
         lire(Entree)
        jusqu'à Entree = "A" OU  Entree = "B" OU  Entree = "C"
       si Entree = "A"
         alors
            répéter
             écrire( "Donnez le type de carburant" )
             lire(type_car)
           jusqu'à type_car = "sup" OU  type_car = "sup_sp"
             si type_car =  "sup"
               alors
                   Prix_car ← P1 * Qte
               sinon
                   Prix_car ← P2 * Qte
             fsi
        sinon si  Entree = "B"
                    alors
                          Prix_car ← P3 * Qte
                    sinon
                          Prix_car ← P4 * Qte
                fsi
       fsi
       Recette ← Recette + Prix_car
      fin pour
    écrire( "La recette journalière de cette station service est : " , Recette)
   fin
```

مناقشة الحل:

تقوم الخوارزمية أعلاه بحساب إيرادات محطة الوقود اليومية، ولحساب ذلك تستعمل حلقة pour لجمع مبالغ التزود بالوقود لكافة المركبات التي تزودت به خلال اليوم، والبالغ عدد NB_V الواحدة تلو الأخرى، وفي نهاية عملية الجمع تقوم الخوارزمية بعرض هذه الإيرادات على الشاشة.

الخاتمة

الخاتمة:

لقد تم تخصيص هذه المطبوعة لطلبة ميدان العلوم الاقتصادية والعلوم التجارية وعلوم التسيير، لذا راعينا فيها البساطة واليسر، وحاولنا قد المستطاع الابتعاد عن التعقيد والغموض واستعملنا لغة عربية بسيطة وسلسة للتعريف بمبادئ كتابة الخوارزميات، وكيفية التعامل مع مختلف المسائل التي من الممكن أن تواجه الطالب في حياته المهنية أو الأكاديمية، وحرصنا أن تكون فصول هذه المطبوعة الأربعة مترابطة ومتسلسلة ابتداء من الفصل الأول الذي تم تخصيصه للتعريف بمجال المعلوماتية عموما وبجزأيها المادي والبرمجي أو ما يعرف بـ (software, hardware) ، كما تم تناول بعض التعاريف الخاصة بالشكل العام للخوارزمية والفرق بينها وبين البرنامج، وكيفية كتابة خوارزميات بسيطة إضافة إلى بعض التعليمات الأساسية كتعليمة التخصيص وتعليمة القراءة والكتابة. وانطلاقا من الفصل الثاني، والذي خصصناه للتعليمات الشرطية بكل أنواعها، فهي تعتبر من أهم التعليمات المستعملة في الخوارزميات، بدأنا في جذب انتباه الطالب إلى الإمكانيات التي تتيحها هذه التعليمات لحل مجموعة من المسائل السهلة، وهذا من خلال الأمثلة التي صاحبت الدرس والتي أُدرجت لشرح التعليمات الشرطية مع مناقشة وتحليل الأجوبة المقترحة حتى يتسنى للطالب الاستيعاب والفهم السليم للخوارزميات المقترحة، وحتى يتعود على كتابة الخوارزميات وقراءتها وفهمها، وحتى تكتمل الفائدة ويكتسب الطالب المهارات اللازمة لكتابة الخوارزميات بشكل صحيح وينجذب نحو هذا الميدان، احتوى هذا الفصل على مجموعة من التمارين المقترحة والمحلولة التي أدرجت في نهايته مع مناقشة الحلول ومحاولة إدراج حلول أخرى لنفس التمرين، وهذا كله يندرج في تلقين الطالب أساسيات كتابة الخوارزميات والتي يمكن أن تتعدد وتختلف لحل نفس المسألة أو التمرين. ثم بالوصول إلى الفصل الثالث، بدأ الطالب يكتشف الإمكانيات التي تتيحها التعليمات التكرارية، وهي التعليمات التي تُستعمل لحل المسائل التكرارية وهي كثيرة الاستعمال والاستخدام في كافة الشؤون والأحوال اليومية التي قد تواجهنا في حياتنا المهنية أو الشخصية، تم اختتام هذا الفصل بمجموعة كبيرة من التمارين المستقاة من مختلف سلاسل التمارين السنوات الماضية، كما تضمن هذا الفصل اقتراحات حلول هذه التمارين مع مناقشة هذه الحلول ومحاولة إعطاء حلول أخرى ويترك المجال للطالب لنقدها أو تنقيحها أو إثراءها.

وجاء الفصل الرابع كمحصلة لكل ما تم التطرق إليه في الفصول السابقة، فقد احتوى على مجموعة شاملة من المسائل المستقاة من مختلف الامتحانات التي اقترحتها في السنوات الماضية، والتي مست ميادين عدة وانتقلت بالطالب من التمارين الأكاديمية والبيداغوجية إلى المسائل الواقعية والتطبيقية، بحيث اهتمت كل مسألة بميدان معين، واحتوت كل

مسألة على ملخص لكل التعليمات التي تم التطرق إليها في الفصول الثلاثة الأولى لهذه المطبوعة، كما احتوى هذا الفصل على الحلول المقترحة لحل هذه المسائل، واحتوى كذلك، إضافة إلى الحلول، مناقشة مفصلة لتلك الحلول ومحاولة تفصيل طريقة الحل وطريقة كتابة الخوارزمية، وكذا طريقة التفكير للوصول إلى الحلول المقترحة، ومع أن هذا الفصل احتوى على أكثر من عشرة مسائل، إلا أن جميعها يشترك في طريقة الوصول إلى الحل من خلال تقسيم عناصر المسألة إلى مدخلات ومخرجات وطريقة المعالجة.

وفي الأخير لا يسعني إلا أن أشكر الله الكريم المنان على منّه وتوفيقه لي في إتمام هذا العمل، والذي أرجو أن يستفيد منه طلبتنا الأعزاء ويجدوا فيه المادة العلمية الصحيحة والبسيطة.

قائمة المراجع

قائمة المراجع

Livres :

- Léry Jean-Michel, **Algorithmique en C, C++, Java, Python et PHP**, Paris, France, Ed. Ellipse, 2019.
- Knuth Donald Ervin, Cegielski Patrick, **Algorithmes**, Paris, France, CSLI Publications, 2011.
- Harris Simon, Ross James, **Beginning Algorithms**, Indianapolis, Indiana, the USA, Wiley Publishing Inc., 2006.
- Kleinberg John, Tardos Éva, **Algorithm Design**, New York, The USA, PEARSON Education, Inc. 2006.

Livres électroniques :

- Bullinaria John, **Data Structures and Algorithms**, School of Computer Science, University of Birmingham, Birmingham, UK, Version of 27 March 2019, disponible sur le lien: *https://www.cs.bham.ac.uk/~jxb/DSA/dsa.pdf*
- Cormen Thomas H., Leiserson Charles E., et al., **Introduction to Algorithms, Third Edition**, The MIT Press, Cambridge, Massachusetts London, England, 2009, disponible sur le lien:

https://edutechlearners.com/download/Introduction_to_algorithms-3rd%20Edition.pdf