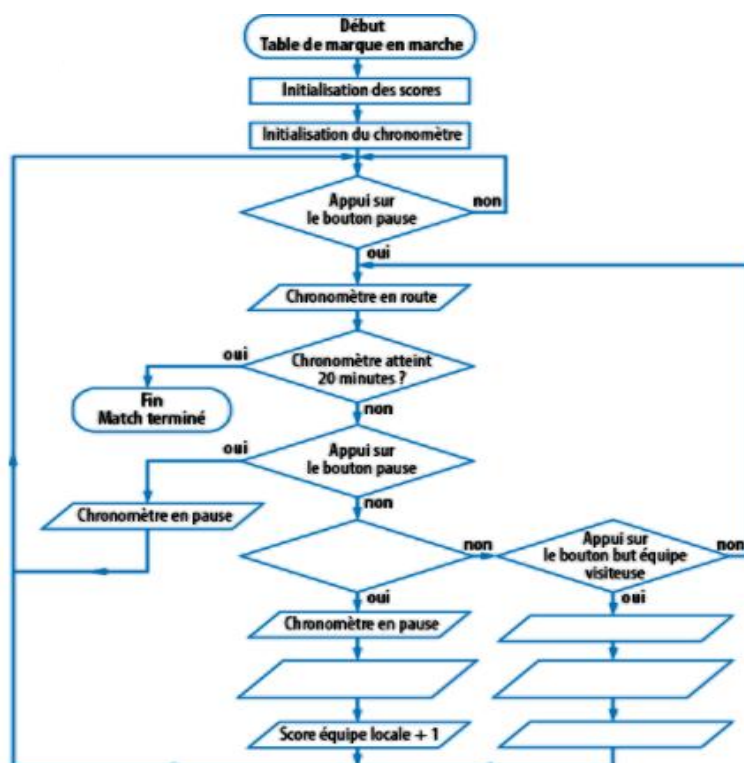


ECRITURE ET EXECUTION DES ALGORITHMES INTEGRANT UNE STRUCTURE DE DONNEES

- ✓ Décrire les structures de contrôle ;
- ✓ Citer cinq structures de données ;
- ✓ Déclarer un tableau ;
- ✓ Parcourir un tableau pour effectuer la lecture et l’affichage ;
- ✓ Ecrire l’algorithme de recherche séquentielle dans un tableau ;
- ✓ Ecrire l’algorithme de recherche du maximum ou du minimum dans un tableau ;
- ✓ Exécuter pas à pas un algorithme de recherche
- ✓ séquentielle dans un tableau ;



Leçon 30 : Algorithme de tri

28

UNITE D'ENSEIGNEMENT 28 : GENERALITES SUR LES ALGORITHMES (RAPPELS)

OBJECTIFS PEDAGOGIQUES :

- Enoncer les généralités sur les algorithmes;
- Citer quelques structures de contrôle ;
- Citer cinq structures de données ;
- Exécuter pas à pas un algorithme simple ;

CONTROLE DE PRESREQUIS :

1. Définir algorithme, algorithmique, instruction, variable, compteur
2. Donner les structure minimale d'une algorithme
3. Donner la syntaxe de déclaration d'une variable et celle d'une constante

SITUATION PROBLEME :

Un particulier voudrait un petit programme qui permet de dire si un individu est mineur ou majeur. Un de vos amis qui a la charge de le faire mais il est bloqué sur l'algorithme et il vous demande de l'aide.

Consignes :

1. Qu'est-ce qu'un algorithme ? (**Réponse(s) attendue(s)** : suite finie ordonnée d'instruction permettant de résoudre un pb)
2. Quels sont les différentes parties d'un algorithme ? (**Réponse(s) attendue(s)** : l'entête, partie des déclaration et le corps)
3. Dans quelle partie de l'algorithme place-t-on les instructions simples? (**Réponse(s) attendue(s)** : dans le corps de l'algorithme)
4. De quel structure algorithmique a-t-on besoin pour tester si un individu est mineur ou pas dans l'algorithme (**Réponse(s) attendue(s)** : De la structure alternative « si »)

RESUME

I- CONCEPTS GENERAUX SUR LES ALGORITHMES

Un algorithme est une suite finie ordonnées d'instructions permettant de résoudre un problème donnée. Une **instruction** quant à elle est un ordre qu'on donne à un ordinateur. On distingue plusieurs types d'instructions en algorithmique dite instructions simple, qui sont :

↔ **La lecture** : il permet de faire saisir à l'utilisateur la valeur d'une variable au clavier. Sa syntaxe est : **Lire** (nom_variable) ;

↔ **L'écriture ou affichage à l'écran** : il permet d'afficher la valeur d'une variable ou un message à l'écran. Sa syntaxe est : **Ecrire** (nom_variable) ; ou **Ecrire** ("message", nom_variable) ; ou **Ecrire** ("message") ;

↔ **L'affectation** : instruction permettant d'affecter une valeur à une variable. On utilise la syntaxe suivante : nom_variable ← valeur ; ou nom_variable ← expression ; dans le cas, les expressions sont composées d'opérateurs et d'**opérandes**. Les différents opérateurs peuvent être les **opérateurs arithmétiques** (addition(+), soustraction (-), division (/), multiplication (*), division entière (div), reste d'une division entière ou modulo (mod),...), les **opérateurs de comparaisons** (=, <, <=, >=, >, <> (différent)...), les **opérateurs logiques** (|| (OU logique), && (ET logique),...).

Pour utiliser une variable ou une constante dans un algorithme, on doit la définir par son nom suivie de la nature (type) des données qu'il contiendra. Les types de base en algorithmique sont : **Entier**, **Numerique**, **Reel**, **Booleen**, **Caractere**, **Chaine**,.... Il arrive parfois que la nature de certaines données n'existe pas parmi les types de base en algorithmique. Dans ce cas on utilisera un nouveau concept : les **structures de données**.

II- LES STRUCTURES DE DONNEES

Une **structure de données** est un type de variable fabriquée à partir d'autres types de variables (généralement les types de base). On peut citer comme exemple de structure données :

- ✓ **Les tableaux** : c'est une suite de plusieurs variables de même type repérée par des index
- ✓ **Les enregistrements** : c'est aussi un ensemble de plusieurs variables mais à la différence des tableaux, il peuvent être de différents type. Ils sont



généralement utilisés lorsqu'on voudrait utiliser des variables composées de autres variables de types différents comme une personne qui définie par son nom, son prenom,...

- ✓ **Liste** : c'est ensemble de variables de même type comme les tableaux. A la seule différence que dans les listes pour accéder à un élément il faudrait soit connaître son précédent ou son suivant.
- ✓ **Pile** : est une structure de données dynamique ayant la propriété lors d'une extraction, l'élément extrait est celui qui a été introduit le récemment. On parle de **LIFO** (Last In First Out ou dernier entré, premier sorti).
- ✓ **File d'attente ou Queue** (anglais) : est une structure de données dynamique telle que lors d'une extraction, l'élément extrait est celui qui s'y trouve depuis le plus longtemps. On parle de **FIFO** (First In First Out ou premier entré, premier sorti)

III- LES STRUCTURES DE CONTRÔLE

Les **structures de contrôle** sont là pour permettre à l'ordinateur d'accomplir des actions plus complexes. On distingue principalement deux types de structures de contrôles : **structure alternative ou conditionnelle**, **structure répétitive ou itérative**.

1- Structure alternative ou conditionnelle

La structure la plus simple est la conditionnelle. On veut exécuter une action ou une autre selon qu'un test est vrai ou faux. On distingue deux types de structure alternative dont les syntaxes sont ci-dessous :

Structure alternative réduite	Structure alternative complète
<u>Si</u> (condition) <u>alors</u> bloc instructions ; <u>FinSi</u>	<u>Si</u> (condition) <u>alors</u> bloc instructions 1; Sinon bloc instructions 2; <u>FinSi</u>

2- Structure répétitive, itérative ou boucle

D'autres permettent de répéter une séquence d'instructions un nombre fini de fois alors que d'autre le font un nombre indéterminé de fois et s'exécutent selon une condition. On distingue :



- Les boucle ayant dont le nombre de répétition est finie : la boucle **pour**

Syntaxe boucle « Pour »
<u>Pour</u> compteur <u>de</u> val_initial <u>à</u> val_final <u>faire</u> bloc instructions ; <u>FinPour</u>

- Les boucles **répéter... jusqu'à** et **tant que... faire** sont ce qu'on appelle des **boucles conditionnelles**.

Structure boucle tantque	Structure boucle répéter
<u>Tantque</u> (condition vraie) <u>faire</u> bloc instructions ; <u>FinTanque</u>	<u>Repeter</u> bloc instructions; <u>Jusqu'à</u> (condition vraie)

Exemple : Algorithme qui permet de calculer le PGCD de deux nombres entiers **a** et **b** lus au clavier.

Algorithme PGCD ;

Var a, b : Entier ;

Debut

Ecrire(" entrer la valeur du premier nombre") ;

Lire(a) ;

Ecrire(" entrer la valeur du deuxième nombre") ;

Lire(b) ;

Tantque (a<>b) **faire**

Si (a>b) **alors**

 a←a-b ;

Sinon

 b←b-a ;

FinSi

FinTanque

Ecrire(" PGCD=", a) ;

Fin

Exécution l'algorithme ci-dessus pour **a=5** et **b=15**



On sait que $\text{PGCD}(5,15)=5$, vérifions si l'algorithme ci-dessus donne le même résultat

« entrer la valeur du premier nombre »

$a=5$

« entrer la valeur du deuxième nombre »

$b=15$

$a \neq b$ = vrai, $a > b$ = faux, $b=b-a=15-5=10$, $a=5$

$a \neq b$ = vrai, $a > b$ = faux, $b=b-a=10-5=5$, $a=5$

$a \neq b$ = faux, arrêt boucle

« PGCD= 5 »

SITUATION D'INTEGRATION :

Une entreprise voudrait un programme qui gère les mots de passe sur ses portes. Les mots de passe de ces portes sont constitués comme suit : ils sont constitués de cinq chiffres, dont le premier chiffre à droite et le dernier chiffre à gauche sont premiers entre eux. On rappelle que deux nombres **a** et **b** si leur PGCD donne 1. On répète la saisie du mot de passe jusqu'à ce que l'utilisateur saisisse un mot de passe correct. Celui qui est chargé de créer cette application a des difficultés au niveau de l'algorithme et il vous demande de l'aide.

- 1- Définir les termes **algorithme** et **variable**
- 2- Citer deux instructions simples que vous connaissez
- 3- Ecrire une fonction appelée « PGCD » qui prend en paramètre deux nombres **a** et **b**, puis calcule et renvoie leur PGCD
- 4- Expliquez pourquoi dans l'algorithme qui calcule le PGCD on ne peut pas utiliser une boucle « pour »
- 5- Ecrire l'algorithme permettant de résoudre le problème ci-dessus en utilisant la fonction de la question 3). Cette algorithme affichera « **accès autorisé** » dans le cas où le mot de passe est correct et « **accès refusé** » dans le cas contraire.
- 6- Vérifiez si votre est correct si un utilisateur saisisse **21568** et puis l'autre saisisse **41453** comme mot de passe

