

Les opérations de base

Exercices :

1 - Donner la valeur des expressions suivantes:

- VRAI ET FAUX OU VRAI
- $(2 < 3)$ ET $(4 > 5)$
- $(2+6 = 10)$ OU $(8/2 > 3)$

2 - Ecrire un algorithme qui demande trois noms à l'utilisateur et l'informe ensuite s'ils sont rangés ou non dans l'ordre alphabétique.

3 - Ecrire un algorithme qui demande l'âge d'un enfant à l'utilisateur.

Ensuite, il l'informe de sa catégorie :

- "Poussin" de 6 à 7 ans
- "Pupille" de 8 à 9 ans
- "Minime" de 10 à 11 ans
- "Cadet" après 12 ans

4 - Lire trois valeurs entières a, b et c. afficher le maximum des trois.

5 - Trouver le prix à payer sachant qu'une facture inclut une somme de 40dhs de frais fixes et que s'ajoute un prix en fonction de la consommation :

- 1 dhs/kWH pour les 100 premiers kilowatts heures
- 0,7 dhs/kWH pour les 150 suivants
- 0,4 dhs/kWH au-delà

6 - Construire algorithme permettant de lire une note, de vérifier si cette note est entre 0 et 20, et de déterminer la mention associée à cette note :

- insuffisant en dessous de 10
- passable de 10 à 11
- assez bien de 12 à 13
- bien de 14 à 15
- très bien de 16 à 20

7 - Ecrire un algorithme permettant de résoudre une équation du second degré.

$$Ax^2 + Bx + C = 0$$

Les opérations de base

Exercices (TD1) :

1 - Ecrire un algorithme qui demande un nombre de départ, et qui ensuite affiche les dix nombres suivants. Par exemple, si l'utilisateur entre le nombre 17, le programme affichera les nombres de 18 à 27.

2 - Ecrire un algorithme qui demande successivement 5 nombres à l'utilisateur, et qui lui dise ensuite quel était le plus grand parmi eux. La saisie des nombres s'arrête lorsque l'utilisateur entre un zéro.

3 - Ecrire un algorithme qui fait la conversion d'un nombre décimal en nombre binaire. On utilise la division sur 2 et concaténation des quotients de chaque division.

4 - Ecrire un algorithme qui permet de simuler les affichages d'un compte à rebours à partir d'un temps (minutes, secondes) donné.

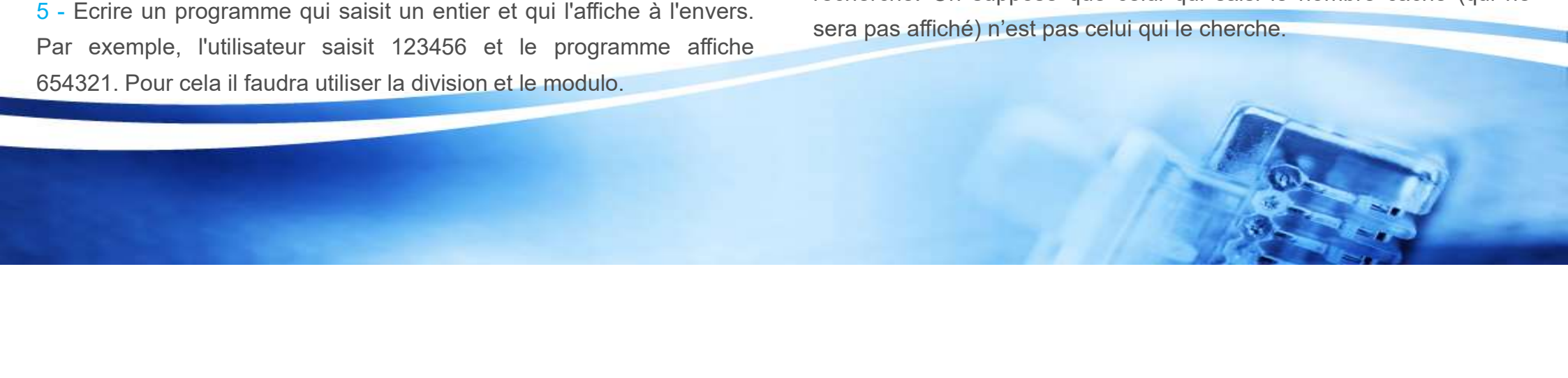
5 - Ecrire un programme qui saisit un entier et qui l'affiche à l'envers. Par exemple, l'utilisateur saisit 123456 et le programme affiche 654321. Pour cela il faudra utiliser la division et le modulo.

6 - Un nombre entier est dit parfait s'il est égal à la somme de ses diviseurs à part lui-même.

Exemple : $6 = 1+2+3$, donc 6 est un nombre parfait. Ecrire le programme qui permet de vérifier si un nombre donné n'est parfait ou non.

7 - Modifier l'algorithme précédent pour permettre à l'utilisateur de saisir un nombre entier n et afficher les nombres parfaits de 1 à n.

8 - Ecrire un algorithme qui permet d'essayer de deviner la valeur d'un nombre entier caché. Si la valeur saisie est supérieure ou inférieure, la machine affiche respectivement "c'est moins" ou "c'est plus" jusqu'à soit que le nombre soit trouvé ou que l'utilisateur abandonne la recherche. On suppose que celui qui saisit le nombre caché (qui ne sera pas affiché) n'est pas celui qui le cherche.



Les opérations de base

Exercices (TD1) :

9 - Ecrire un algorithme qui permet de vérifier le mot de passe saisi au clavier. L'utilisateur a droit à 3 chances pour que la machine lui affiche le succès de l'authentification sinon un message de compte bloqué sera affiché.

Dans le cas où il veut encore retenter l'accès au compte une nouvelle chance lui est proposée sous forme de question secrète à laquelle il devra répondre. Si la réponse est incorrecte l'accès lui est définitivement refusé pour l'exécution en cours.

Note : le mot de passe correct est 'Bonjour' et la réponse correcte à la question secrète est 'ISTA'.

10 - Ecrire un programme qui demande à l'utilisateur de taper des entiers strictement positifs et qui affiche leur moyenne. Lorsqu'on tape une valeur négative, le programme affiche ERREUR et demande de retaper une valeur. Lorsqu'on tape 0, cela signifie que le dernier entier a été tapé. On affiche alors la moyenne.

Si le nombre d'entiers tapés est égal à 0, on affiche PAS DE MOYENNE.

11 - Écrire un programme qui demande de saisir 10 entiers et qui affiche le nombre d'occurrences de la valeur la plus haute.

12 - Écrire un programme qui permet de faire des opérations sur un entier. Le programme affiche la valeur de l'entier puis affiche le menu suivant :

1. Ajouter 1
2. Multiplier par 2
3. Soustraire 4
4. Quitter

Le programme demande alors de taper un entier entre 1 et 4. Si l'utilisateur tape une valeur entre 1 et 3, on effectue l'opération, on affiche la nouvelle valeur de l'entier puis on réaffiche le menu et ainsi de suite jusqu'à ce qu'on tape 4. Lorsqu'on tape 4, le programme se termine.

Les opérations de base

Exercices (TD2) :

13 - Ecrire un algorithme qui demande à l'utilisateur de saisir un entier N et d'afficher la suite de Fibonacci jusqu'à N éléments.

La suite de Fibonacci commence par les éléments 0 1 et chaque élément est calculé à partir de la somme des 2 éléments qui le précède.

Par exemple une suite de Fibonacci de 6 éléments est la suivante :

0 1 1 2 3 5

14 - Écrivez un algorithme pour compter la fréquence des chiffres dans un nombre donné.

Par exemple dans l'entier 58321553 :

le chiffre 0 existe 0 fois

le chiffre 1 existe 1 fois

le chiffre 2 existe 1 fois

le chiffre 3 existe 2 fois

Etc...

15 - Écrire un algorithme qui demande de saisir un entier N et un entier M. Par exemple pour les valeurs, N=5 et M=9, le résultat affiché doit être :

9 99 999 9999 99999

16 - Écrire un algorithme qui permet de calculer la somme des entiers impaires jusqu'à un entier N saisi par l'utilisateur.

17 – Ecrire un algorithme qui permet d'afficher le résultat suivant :

1
22
333
4444
55555
666666
7777777
88888888
999999999

Pour ajouter un retour à la ligne vous pouvez utiliser l'écriture "\n"

Les opérations de base

Exercices (TD2) :

18 - Ecrire un algorithme qui demande à l'utilisateur de saisir un nombre N et qui affiche le résultat suivant pour N = 5 :

```
*  
* *  
* * *  
* * * *  
* * * * *  
* * * * *  
* * * *  
* * *  
* *  
*  

```

Pour ajouter un retour à la ligne vous pouvez utiliser [Ecrire "\n"](#)

19 - Ecrire un algorithme qui demande à l'utilisateur de saisir un nombre N et qui affiche un pyramide où le nombre de lignes correspond à N, par exemple si N = 5 le résultat est :

```
      *  
     * *  
    * * *  
   * * * *  
  * * * * *  
 * * * * *  
* * * * *  

```