

LYCÉES SECONDAIRES

Okba Ibn Nefaa

Ibn Jazzar

Chedli Atallah

Kairouan

SECTION : SCIENCES DE
L'INFORMATIQUE

Enseignants :

M. SALEH TALBI

Mme. Imen Ben Saad

Mme. Ibtihel khadraoui

M. Hammouda Essid

BAC BLANC 2022

EPREUVE THÉORIQUE

ALGORITHMIQUE ET PROGRAMMATION
(Nouveau régime)

DURÉE : 3h

DATE: 12/05/2022

COEFFICIENT : 3

EXERCICE 1: (3 points)

Soit l'algorithme de la fonction suivante :

Fonction traitement (ch :.....) :

Début

$i \leftarrow -1$

$j \leftarrow \text{long}(\text{ch})$

Répéter

$i \leftarrow i+1$

$j \leftarrow j-1$

Jusqu'à ($\text{ch}[i] \neq \text{ch}[j]$) ou ($i \geq j$)

$B \leftarrow \text{"False"}$

Si $\text{ch}[i] = \text{ch}[j]$ alors

$B \leftarrow \text{"True"}$

Fin si

.....

Fin

Questions :

1. Compléter les pointillés dans cette fonction.
2. Quelles sont les valeurs retournées par la fonction traitement si on l'applique sur les deux chaînes suivantes.
 - $\text{ch} = \text{"1010"}$
 - $\text{ch} = \text{"1011101"}$
3. Dédire le rôle de cette fonction.
4. En déduire une solution récursive de la fonction traitement.

EXERCICE 2: (2 points)

Une matrice carrée M de taille $N \times N$ est dite dérangement si la valeur de chaque élément de la matrice est :

- différente de l'indice de la ligne correspondante.
- différente de l'indice de la colonne correspondante.
- comprise entre $[1, N \times N]$.

Exemples:

0	1	2	3
0	6	5	2
1	3	4	8
2	3	7	9
3	2	5	4

Matrice1 : N'est pas un dérangement

2 se trouve dans la colonne d'indice 2

0	1	2	3
0	6	5	9
1	3	4	8
2	3	7	9
3	2	5	4

Matrice2 : N'est pas un dérangement

1 se trouve dans la ligne d'indice 1

0	1	2
0	6	5
1	3	4
2	3	7

Matrice3 : N'est pas un dérangement

17 > $N \times N$

0	1	2
0	1	5
1	3	4
2	3	7

Matrice4 : est dérangement

Travail à faire :

Ecrire l'algorithme d'un module nommé **DERANGEMENT** qui permet de vérifier si une matrice carrée M de taille $N \times N$ est dérangement ou non.

EXERCICE 3: (5 points)

Le calcul de la racine carrée d'un réel x par la méthode de newton est définie par la suite u de la façon suivante :

$$U \begin{cases} U_0 = X \\ U_{n+1} = (U_n + X/U_n)/2 \end{cases}$$

Cette suite converge vers \sqrt{X} . Le calcul est arrêté lorsque $|X - U_n^2| < 10^{-4}$.

Question N° 1:

Ecrire l'algorithme d'une fonction nommée **Racine(x: réel) : réel** qui calcule une valeur approchée de la racine carrée d'un réel x à 10^{-4} près.

Question N° 2:

Soit l'intégrale de la fonction $f : X \rightarrow \sqrt{(X * X) + 1}$ définie sur $[0..3]$. On se propose de calculer l'aire de cette fonction par la **méthode de rectangle gauche** en utilisant la fonction **Racine** élaborée en question (1) et selon le principe suivant:

- ✓ Diviser l'intervalle $[0..3]$ en N sous intervalles avec N un entier donné ($10 \leq N \leq 100$).
- ✓ Sauvegarder dans un tableau d'enregistrements T :
 - L'abscisse X de chaque sous intervalle.
 - La surface S de chaque rectangle.
- ✓ Finalement l'aire résultante de la fonction f est égale à la somme des surfaces des rectangles déjà calculées et stockées dans le tableau T .

On vous donne l'algorithme du programme principal suivant :

Algorithme Aire

Début

Répéter

Ecrire ("n=")

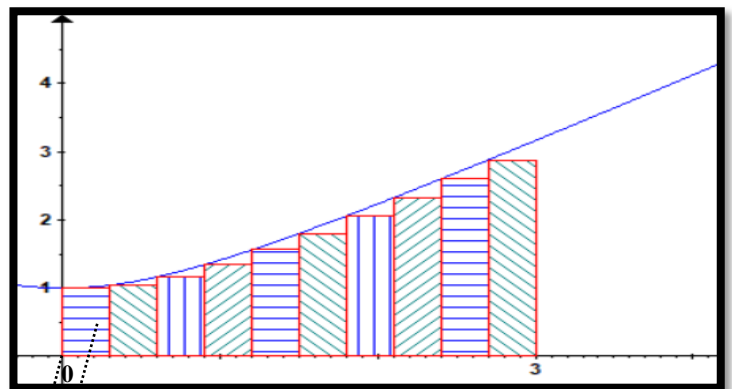
Lire(n)

Jusqu'à ($10 \leq N \leq 100$)

Remplir_tab(T, N)

Ecrire ("Aire=", $SOMME(T, N)$)

Fin



Pour $N=10$, on obtient le tableau T suivant →

Avec :

$X_0=0$ et S_0 = surface du premier rectangle

X_0	X_1	X_2	X_8	X_9
S_0	S_1	S_2					S_8	S_9

- a. En utilisant la fonction **Racine** de la question (1), écrire l'algorithme de la procédure **Remplir_tab** permettant de remplir le tableau d'enregistrements T par les abscisses X de chaque sous intervalle et les surfaces S de chaque rectangle correspondant.
- b. Ecrire l'algorithme de la fonction **SOMME** qui permet de calculer l'aire résultante de la fonction f , sachant que l'aire = $S_0 + S_1 + \dots + S_{n-1}$

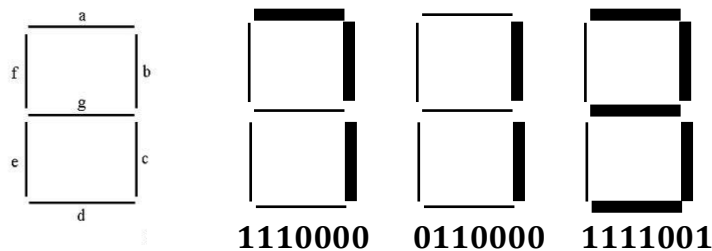
EXERCICE 4:(10 points)

Une calculatrice, ou calculette, est une machine conçue pour simplifier, et fiabiliser, des opérations de calculs. Dans les années 1970, elles se miniaturisent pour devenir portables grâce à l'affichage à sept segments.

Les afficheurs 7 segments sont un type d'afficheur très présent sur les calculatrices et les montres à affichage numérique : les chiffres s'écrivent en allumant ou en éteignant des segments. Quand les 7 segments sont allumés, on obtient le chiffre 8.

- Un segment allumé est représenté par le caractère 1
- Un segment éteint est représenté par le caractère 0

Voici quelques exemples représentés avec l'affichage à 7 segments :



- ✓ Pour que la calculatrice affiche la valeur 7 il faut que les segments a, b, c doivent être allumés et les segments d, e, f, g éteints (donc le code correspondant au chiffre 7 est 1110000).
- ✓ Pour que la calculatrice affiche la valeur 3 il faut que les segments a, b, c, d, g doivent être allumés et les segments e, f doivent être éteints. (donc le code correspondant au chiffre 3 est 1111001)
- ✓ On note que l'état des segments allumés ou éteints commence par ordre alphabétique du segment a puis segment b ... jusqu'au segment g.

Etant donnée un fichier d'enregistrement intitulé "calculatrice.dat", où chaque enregistrement est composé de deux champs :

- Un champ **chiffre** de type entier contenant un chiffre de 0 à 9.
- Un champ **code** de type chaîne de caractères contenant les états des segments (suite de 0 et 1).

On obtient donc le contenu du fichier "calculatrice.dat" suivant :

chiffre	code
0	1111110
1	0110000
2	1101101
3	1111001
4	0110011
5	1011011
6	1011111
7	1110000
8	1111111
9	1110111

En Utilisant le fichier "calculatrice. "calculatrice.dat" :

❖ Calculer depuis un fichier texte nommé "operations.txt", les opérations y enregistrées et sauvegarder le résultat de chaque opération dans un autre fichier texte nommé "resultat.txt" afin de représenter le résultat sur l'afficheur.

→ Chaque ligne du fichier "operations.txt" contient une opération d'addition des entiers positifs dont chaque chiffre d'un entier est représenté par 7 caractères binaires.

Exemple :

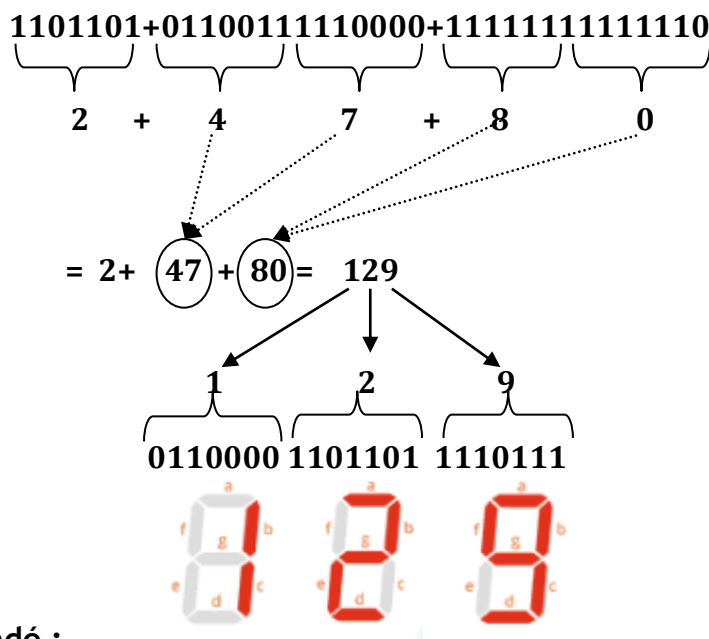
Si le contenu du fichier "operation.txt" est le suivant :

```
1101101+01100111110000+1111111111110
1111001+1011111
```

Après le calcul, le programme remplit et affiche le contenu du fichier "resultat.txt" comme suit :

```
011000011011011110111
1110111
```

En effet : Le résultat du calcul de la première ligne du fichier "operation.txt" est déterminé comme suit :



Travail demandé :

1. Décomposer le problème en modules.
2. Ecrire les algorithmes de chacun des modules envisagés.

NB :

Le candidat n'est pas appelé à remplir les deux fichiers "calculatrice.dat" et "operation.txt".

Bon Travail