

Série d'exercices

Les algorithmes arithmétiques

Exercice N°1 :

Soit la fonction *Anonyme* Suivante :



```

Fonction Anonyme(ch:Chaine de caractère; x: Entier): booléen.....
Début
    aux ← "0123456789ABCDEF"
    Tant Que (pos(Majus(ch[0]),Souschaine(aux,0,x))≠-1) ET (ch≠"") Faire
        ch ← Efface(ch,0,1)
    Retourner ch=""
Fin
  
```

Travail demandé :

1. Quel est le type du résultat de cette fonction ?
2. Donnez la trace d'exécution de la fonction *Anonyme* pour chacun des cas suivants :
 - *Anonyme*("5B9C ",12) **donne faux**
 - *Anonyme*("56819",8) **donne faux**
3. Quel est le rôle de cette fonction ? **vérifier si le nombre ch est appartient a base x ou non**

Exercice N°2 :

Soit la fonction inconnue suivante :

```

Fonction inconnue (n :entier) : ...chaine.....
Début
    Si (n=0) alors
        Retourner ""
    Sinon Si (n mod 2 = 0) Alors
        Retourner inconnue (n div 2) + "0"
    Sinon
        Retourner inconnue (n div 2) + "1"
    Fin Si
Fin
  
```

Travail demandé :

1. Quel est le type du résultat de cette fonction ?
2. Donnez la trace d'exécution de la fonction « inconnue » pour $n = 12$ **donne "1100"**
3. Quel est le rôle de cette fonction ? **convertir un nombre décimal en un nombre binaire**

Exercice N°3 :

Soit la fonction inconnue suivante :

Fonction Inconnue(N :Entier) :Chaine de caractère

Début

Ch ← ""

Répéter

Aux ← Convch(N mod 6)

Ch ← aux + Ch

N := N div 6

Jusqu'à N=0

Retourner Ch

Fin

Travail demandé :

1. Exécuter manuellement cette fonction sur les valeurs $N=78$ et $N=25$
2. Quel est le rôle de cette fonction

Exercice N°4 :

Soit l'algorithme de la fonction **Inconnue** suivante :

Fonction **Inconnue**(Ch :chaîne de caractère ;i,k :Entier) :.....

Début

Si $i=0$ Alors

Retourner 0

Sinon Si $Ch[i]="1"$ Alors

Retourner k+Inconnue(Ch,i-1,k*2)

Sinon

Inconnue ← Inconnue(Ch,i-1,k*2)

Finsi

Fin

Travail demandé :

1. Quel est le type de résultat de cette fonction ?
2. Donnez le résultat de la fonction Inconnue pour chacun des cas suivants :
 - **Inconnue("1010",4,1)**
 - **Inconnue("11011",5,1)**
3. En déduire le rôle de cette fonction.

Exercice N°5 :

Pour vérifier si un entier est divisible par 7 ou non, on utilise la règle de divisibilité suivante :

- 1) On conserve tous les chiffres de N sauf le dernier, et on lui retranche deux fois le dernier

Exemple : Pour N=8745 on fait l'opération : $874 - 2 \times 5 = 864$

- 2) On procède de même avec le résultat : $86 - 4 \times 2 = 78$

- 3) On répète l'action 2 plusieurs fois jusqu'à obtenir un résultat appartenant à l'intervalle $[-9..9]$

→ N est divisible par 7 lorsque le résultat final est l'un de ces trois valeurs : -7, 0 ou 7

Travail demandé :

Proposer un algorithme d'un module permettant de vérifier si un entier N est divisible par 7 ou non.

Exercice N°6 :

Un nombre est divisible par 11, si la différence entre la somme des chiffres de rang impair et la somme des chiffres de rang pair est divisible par 11 (Exemple : 18282 est divisible par 11 car $(1+2+2)-(8+8) = -11$ qui est divisible par 11)

Travail demandé :

Proposer un algorithme d'un module permettant de vérifier si un entier N est divisible par 11 ou non.

Exercice N°7 :

Soit N un entier naturel de grande taille avec un nombre de chiffres le composant compris entre **20** et **200**. N est divisible par **8**, si **l'un des cas suivants** est vérifié :

1^{er} cas : Le chiffre des centaines est **pair** et le nombre formé par les 2 derniers chiffres (les plus à droite) est multiple de **8**.

2^{ème} cas : Le chiffre des centaines est **impair** et le nombre formé par les 2 derniers chiffres (les plus à droite) diminué de 4 est multiple de **8**.

Exemple 1 : Si N=1245896578541236593224, alors :

- Son chiffre des **centaines** est **pair** puisqu'il est égal à **2**

- Le nombre formé par les **2 derniers chiffres** (les plus à droite), est **multiple de 8** puisqu'il est égal à **24** qui est multiple de **8**.

→ Donc, le premier cas est vérifié, et **N** est multiple de **8**.

Exemple 2 : Si **N=1245896578541236593120**, alors :

- Son chiffre des **centaines** est **impair** puisqu'il est égal à **1**
- Le nombre formé par les **2 derniers chiffres** (les plus à droite), **diminué de 4**, est **multiple de 8** puisqu'il est égal à **20** et que **16** (**20-4**) est multiple de **8**

→ Donc, le deuxième cas est vérifié, et **N** est multiple de **8**.

Exemple 3 : Si **N=1245896578541236593221**, alors :

- Son chiffre des **centaines** est **pair** puisqu'il est égal à **2**
- Le nombre formé par les **2 derniers chiffres** (les plus à droite), **n'est multiple de 8** puisqu'il est égal à **21**.

→ Donc, **N** n'est multiple de **8**.

Travail demandé :

Ecrire un algorithme d'un module intitulé **Div8**, permettant de vérifier la divisibilité d'un entier **N** par **8** selon le principe décrit précédemment.

Exercice N°8 :

Pour savoir si une chaîne non vide formée par des chiffres seulement est divisible par 137, il suffit de séparer cette chaîne par tranche de 4 chiffres en partant des unités et d'insérer alternativement des **-** et des **+** entre les tranches à partir du début de la chaîne en commençant par un **-**. On effectue l'opération ainsi écrite et si le résultat est divisible par 137, alors la chaîne considérée est divisible par 137.

Exemple

Soit la chaîne **ch= « 2510792736157732104 »**

On la sépare par tranche de quatre chiffres à partir des unités

251| 0792| 7361|5773|2104

On intercale alternativement des **+** et des **-** à partir du début en commençant par **-**

251 - 0792 + 7361 - 5773 + 2104

On effectue l'opération ainsi écrite

251-0792+7361-5773+2104 =**3151**

On vérifie aisément que **3151** est divisible par 137 donc la chaîne 2510792736157732104 est divisible par 137

```
def div_137(ch):
    s=0
    op=-1
    while ch!="":
        if len(ch)>=4:
            s=s+int(ch[len(ch)-4:len(ch)])*op
            ch=ch[0:len(ch)-4]
            op=-op
        else:
            s=s+int(ch)*op
            ch=""
    return (abs(s)%137==0)
```

Travail demandé :

Ecrire un algorithme d'un module qui permet de vérifier si une chaîne ch formée seulement par des chiffres est divisible par 137

Exercice N°9 :

On se propose de calculer le PGCD de N entiers positifs pour cela on adopte la méthode suivante :

- Remplir la 1^{ère} ligne de la matrice M par N entiers positifs
- La 2^{ème} ligne de la matrice M sera remplie par les PGCD de chaque deux entiers adjacents de la 1^{ère} ligne
- La 3^{ème} ligne de la matrice M sera remplie par les PGCD de chaque deux entiers adjacents de la 2^{ème} ligne
- Appliquer le même principe pour les autres lignes jusqu'à arrivé à une ligne qui à un seul nombre qui est le PGCD chercher

Exemple : Pour N=4 :

M	1	2	3	4
1	<u>84</u>	<u>72</u>	56	22
2	<u>12</u>	8	2	
3	4	2		
4	<u>2</u>			

$PGCD(\underline{84}, \underline{72}) = \underline{12}$

$NOMBRE_PGCD(\underline{84}, \underline{72}, \underline{56}, \underline{22}) = \underline{2}$

N.B :

- Le candidat n'est pas appelé à remplir la première ligne de la matrice M
- Le candidat doit utiliser la méthode des différences afin de calculer le PGCD de deux entiers adjacents

Principe : cette méthode consiste à remplacer A par A-B si $A > B$ ou B par B-A si $B > A$, la même opération sera répétée jusqu'à obtenir $A = B$, auquel cas le PGCD cherché est A ou B.

Exemple : $PGCD(\underline{84}, \underline{72}) = PGCD(84-72, 72)$

$$= PGCD(12, 72)$$

$$= PGCD(12, 60)$$

$$= PGCD(12, 48)$$

$$= PGCD(12, 36)$$

$$= PGCD(12, 24)$$

$$= PGCD(12, 12)$$

$$\rightarrow PGCD(\underline{84}, \underline{72}) = \underline{12}$$

Travail demandé :

Ecrire un algorithme d'un module intitulé **NOMRE_PGCD** permettant de calculer le PGCD des entiers se trouvant dans la première ligne

Exercice N°10 :

L'ISBN (International Standard Book Number) est un numéro qui permet d'identifier le titre d'un livre.

Ce numéro est formé de 20 caractères regroupés en 4 parties :

- La première partie correspond à la zone linguistique qui est un nombre de 3 chiffres hexadécimaux distincts, et qui commence obligatoirement par une lettre :

Exemples : A12 pour Arabe

FB2 pour Français

- Les deux autres parties (indiquant l'éditeur et le numéro d'ordre dans la production de l'éditeur) sont formées uniquement par des chiffres : 8 chiffres par partie.
- La dernière partie (chiffre ou lettre) correspond à la clé de contrôle dont sa valeur dépend du reste de la division d'un nombre intermédiaire N par 11
 - Si ce reste est non nul et formé d'un seul chiffre, la clé sera égale au reste.
 - Si ce reste est 10 la clé sera notée X.
 - Si le reste est 0 la clé sera égale à la somme des chiffres de la représentation binaire de la première partie

N.B : Le nombre intermédiaire N est obtenu en regroupant les deux chiffres de même position de la 2^{ème} et la 3^{ème} partie de numéro de l'ISBN suivis des deux chiffres de la position suivante jusqu'à ajouter les deux chiffres de la 8^{ème} position.

Exemple1 :

Numéro saisi : **A12-41213104-92651027**

Le calcul intermédiaire donne une valeur de N égale à **4912261531100247**, cette valeur a pour reste **5** dans la division par **11** par conséquent la clé est égale à **5**.

⇒ Numéro final affecté au livre : **A12-41213104-92651027-5**

Exemple2 :

Numéro saisi : **A12-41213104-92651022**

Le calcul intermédiaire donne une valeur de N égale à **4912261531100242**, cette valeur a pour reste **0** dans la division par **11** par conséquent la valeur de la clé est égale à la somme des chiffres de la représentation binaire de A12

$$(A12)_{16} = (101000010010)_2 \rightarrow 1+0+1+0+0+0+0+1+0+0+1+0=4$$

⇒ Numéro final affecté au livre : **A12-41213104-92651022-4**

NB : Vous devez utiliser la règle de divisibilité par 11 suivante:

Un nombre N est divisible par 11 si la valeur absolue de la différence entre la somme des ses chiffres de rang impairs avec ses chiffres de rang pairs est divisible par 11.

Exemple :

Soit N= **91928595** :

$$ABS((1+2+5+5)-(9+9+8+9)) \bmod 11 = 22 \bmod 11 = 0$$

⇒ On conclut que N est divisible par 11

Travail demandé :

On se propose d'écrire un programme permettant de saisir le numéro ISBN d'un livre, déterminer la clé affectée à ce numéro et afficher le numéro final du livre.

- Décomposer ce problème en modules.
- Analyser chacun des modules envisagés.

Exercice N°11 :

Très longtemps, la table de référence la plus utilisée, pour le codage des caractères, fut la table ASCII (American Standard Code for Information Interchange), table compilant un jeu de 128 caractères.

L'objectif de ce problème est de remplir une partie de la table du code ASCII à partir d'une **matrice M** de 6 lignes (numérotées de 2 à 7) et de 16 colonnes (numérotés de 0 à 15) contenant les **95 caractères**, sans utiliser la fonction prédéfinies **ORD**.

M	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
2		!	"	#	\$	%	£	'	()	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
6	`	A	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	P	Q	r	s	t	u	v	w	x	y	z	{		}	~	

N.B :

- Le premier élément de la matrice M[2,0] contient le caractère **espace**.
- Le dernier élément M[7,15] est **non défini**.

Pour déterminer le code ASCII d'un caractère **CC**, on suit les étapes suivantes :

1. convertir respectivement le numéro de la ligne et le numéro de la colonne du caractère **CC** en deux nombres hexadécimaux.
2. concaténer, dans l'ordre, les deux nombres hexadécimaux trouvés afin de calculer le code ASCII hexadécimal du caractère.
3. convertir le code ASCII hexadécimal, déjà trouvé lors de la deuxième étape, en un nombre décimal représentant le code ASCII décimal du caractère **CC**.

Exemple :

- le caractère "**K**" existe dans la ligne 4 et la colonne 11 ($M[4,11]$).
- La conversion de **4** en base hexadécimale donne "**4**".
- La conversion de **11** en base hexadécimale donne "**B**".
- La concaténation de "**4**" et "**B**" donne "**4B**" qui est le code ASCII hexadécimal du caractère "**K**".
- La conversion de "**4B**" en base décimale donne **75** qui est le code ASCII décimal du caractère "**K**".

Travail demandé :

On suppose que le contenu de la matrice est déjà stocké dans un fichier texte nommé "**car.txt**" et telle que chaque ligne contient un seul caractère :

- ♦ Copier le contenu du fichier "**car.txt**" dans une matrice **M**, Le remplissage de la matrice se fait ligne par ligne.
- ♦ Remplir, à partir de la matrice **M**, un fichier de données nommé "**ASCII.dat**" qui contient autant d'enregistrements que de caractères dans la matrice de façon à ce que chaque enregistrement contienne trois champs :
 - **Car** : le caractère en question.
 - **Code_hexa** : le code ASCII en base hexadécimale du caractère.
 - **Code_dec** : le code ASCII en base décimale du caractère.

N.B : Tous les fichiers seront enregistrés sur la racine **C** :

1. Décomposer le problème en modules
2. Dédire les algorithmes des modules envisagés précédemment

Exercice N°12 :

Soit p et q deux entiers naturels tels que $5 \leq p \leq q \leq 32$. On se propose de remplir une matrice **M** à p lignes et q colonnes par des zéro (0) et des uns (1) de la façon suivante :

On mettra 1 dans toute cellule $M[i,j]$ si les représentations de i et de j dans la base 2 ont au moins une fois le chiffre 1 à la même position sinon on y mettra 0. On commencera à lire à partir de la droite.

Exemples

- $M[6,3]$ aura la valeur 1 car les deux représentations binaires ont le chiffre 1 à la même position (deuxième position à partir de la droite), En effet

$$6 = (1\underline{1}0)_2$$

$$\text{Et } 3 = (\underline{1}1)_2$$

- $M[9,4]$ aura la valeur 0 car les deux représentations binaires n'ont pas de 1 à la même position, En effet

$$9 = (1001)_2$$

$$\text{Et } 4 = (100)_2$$

On se propose ensuite de chercher tous les chemins rectilignes menant de la première ligne à la dernière ligne en n'empruntant que des cellules contenant des 1 et en sautant au maximum une seule cellule contenant 0. Par conséquent **un chemin rectiligne est une colonne ne comportant pas deux cellules consécutives contenant 0**.

Pour tout chemin trouvé, on affichera son numéro (l'indice j). En plus les données relatives à ce problème seront enregistrées dans un fichier texte intitulé « *chemin.txt* » et placé sur la racine du lecteur D. ce fichier comportera dans sa première ligne le naturel p suivi d'une espace, suivi du naturel q et comportera dans chacune des lignes qui suivent le numéro du chemin trouvé, la dernière ligne contiendra le nombre de chemins trouvés.

Exemple : Soit $p=5$ et $q=6$ donc la matrice M sera :

M	1	2	3	4	5	6
1	1	0	1	0	1	0
2	0	1	1	0	0	1
3	1	1	1	0	1	1
4	0	0	0	1	1	1
5	1	0	1	1	1	1

Le contenu du fichier

« *chemin.txt* » sera :

« *chemin.txt* »

5_6
1
3
5
6

Travail demandé :

1. Décomposer le problème en modules
2. Dédire les algorithmes des modules envisagés précédemment

Exercice N°13 :

Une image numérique en couleurs est un ensemble de pixels structurés sous forme d'une matrice M formée par N lignes et C colonnes.

Chaque case de la matrice contient un code hexadécimal de 6 caractères qui correspond à une couleur.

Notre objectif consiste à transformer cette image en niveaux de gris.

Cette transformation consiste à remplacer le contenu de chaque case (Les 6 caractères) par un code hexadécimal correspondant composé de 2 caractères seulement.

Pour déterminer ce code on suit les étapes suivantes :

Pour chaque pixel :

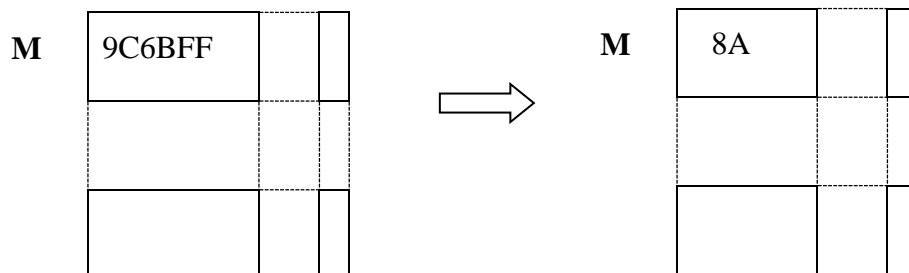
- ✓ Les 6 caractères seront divisés en 3 blocs de 2 caractères représentant les 3 valeurs décimales R, V et B.

Exemple :

Soit le pixel une case de la matrice contenant :

$$\begin{array}{ccc} \mathbf{9} & \mathbf{C} & \mathbf{6} & \mathbf{B} & \mathbf{F} & \mathbf{F} \\ \underbrace{\hspace{1cm}} & \underbrace{\hspace{1cm}} & \underbrace{\hspace{1cm}} & & & \\ \mathbf{R=156} & & \mathbf{V=107} & & \mathbf{B=255} & \end{array}$$

- ✓ On calcule la variable $Y=0.299*R + 0.587*V + 0.114*B \rightarrow Y=138.91$
- ✓ On tronque le réel Y pour obtenir un entier décimal $D \rightarrow D=138$
- ✓ On détermine la représentation hexadécimale de D sur 2 caractères pour obtenir un code H.
 $D=(138)_{10} \rightarrow H=(8A)_{16}$
- ✓ Dans la matrice M, le code de 6 caractères sera remplacé par le code H.



Travail demandé :

On se propose d'écrire un programme qui permet de :

- Saisir le nombre de lignes L et le nombre de colonnes C de la matrice M ($20 < L < 30$ et $30 \leq C \leq 40$)
- Remplir cette matrice M par les codes hexadécimaux des pixels représentant l'image en couleurs (chaque case contienne 6 caractères)
- Remplacer chaque code de la matrice par le code hexadécimal correspondant à sa transformation en niveaux de gris (2 caractères), en utilisant la démarche décrite ci-dessus.
- Afficher le contenu de la matrice résultat.

Travail demandé :

1. Décomposer le problème en modules
2. Déduire les algorithmes des modules envisagés précédemment

Exercice N°14 (Pratique):

Ecrire un programme Python qui permet de :

- Remplir aléatoirement une matrice carrée de taille $N \times N$ ($5 \leq N \leq 20$) par des 0 et des 1
- Générer à partir de chaque ligne de cette matrice une chaîne binaire formée par la concaténation des différentes valeurs de cette ligne
- Générer un fichier intitulé « Div_11.txt » contenant les chaînes binaires dont leur représentation en base décimale est divisible par 11

N.B :

Pour vérifier si cette chaîne est divisible par 11 ou non on doit appliquer la règle de divisibilité suivante :

Un nombre est divisible par 11, si la différence entre la somme des chiffres de rang impair et la somme des chiffres de rang pair est divisible par 11 (**Exemple** : 18282 est divisible par 11 car $(1+2+2)-(8+8) = -11$ qui est divisible par 11)

- Afficher le contenu du fichier « Div_11.txt »

Exemple :

M	1	2	3	4	5	6
1	1	1	0	1	1	1
2	1	0	0	1	0	0
3	1	1	0	1	1	0
4	0	0	1	0	1	1
5	1	1	0	0	0	0
6	0	0	0	0	0	0

« 110111 » = 55 Divisible par 11

« 100100 » = 36 N'est pas divisible par 11

« 110110 » = 54 N'est pas divisible par 11

« 001011 » = 11 Divisible par 11

« 110000 » = 48 N'est pas divisible par 11

« 000000 » = 0 Divisible par 11

⇒ Le fichier « Div_11.txt » sera :

110111

001011

000000