

Devoir de synthèse N°3 (Nouveau régime)		
Commissariat régional de l'éducation de Mahdia	Section : Sciences de l'Informatique	
	Épreuve : Algorithmique et programmation	
Niveau : 4 ^{ème} année	Date : Jeudi 12 mai 2022	Durée : 3 h
Nom & Prénom : Classe :		Note : /20

*N.B. : Le sujet comporte 4 pages à remettre à la fin de l'épreuve.
Chaque solution développée sous forme d'un algorithme doit être accompagnée des tableaux de déclaration nécessaires.*

Exercice 1 (3 points)

Pour chacune des propositions suivantes, mettre dans chaque case, la lettre V si la réponse est correcte ou la lettre F dans le cas contraire.

- 1) Soit un fichier texte "Nombre.txt" contenant des lignes formées uniquement par des chiffres, pour calculer le nombre de lignes contenant des entiers pairs dans ce fichier, on utilise la séquence d'instructions :

Ouvrir("Nombre.txt",FT,"r") nb ← 0 Tantque non fin_fichier (FT) Faire Lire_ligne (FT, C) Si C mod 2 = 0 alors nb ← nb + 1 FinSi FinTantque Fermer (FT)	Ouvrir("Nombre.txt",FT,"r") nb ← 0 Tantque non fin_fichier (FT) Faire Lire_ligne (FT, CH) Si estnum(CH) Alors C ← Valeur(CH) Si C mod 2 = 0 alors nb ← nb + 1 FinSi FinSi FinTantque Fermer (FT)	Ouvrir("Nombre.txt",FT,"r") nb ← 0 Tantque non fin_fichier (FT) Faire Lire_ligne (FT, CH) C ← Valeur(CH) Si C mod 2 = 0 alors nb ← nb + 1 FinSi FinTantque Fermer (FT)
--	---	---

☐
☐
☐

- 2) L'instruction algorithmique $b \leftarrow \text{pos}(c, ch) = -1$ est équivalente à la séquence :

b ← Vrai Si pos(c,ch) ≠ -1 alors b ← Faux FinSi	Si pos(c,ch) ≠ -1 alors b ← Faux sinon b ← Vrai FinSi	b ← non (pos(c,ch) ≠ -1)
--	---	--------------------------

☐
☐
☐

- 3) Soit l'algorithme de la fonction intitulée "Inconnu" suivant :

```

Fonction Inconnu (Ch : Chaîne) : Entier
Début
    Si Ch = "" Alors
        Retourner 0
    Sinon
        Retourner Ord(Ch[0]) - Ord("0") + 10 * Inconnu (Sous-chaîne(Ch,1,Long(Ch)))
    FinSi
Fin

```

Pour la chaîne de caractères Ch = "3245", la fonction **Inconnu**, retourne la valeur :

☐ 14

☐ 5423

☐ 3245

- 4) Soit un nombre hexadécimal **N** formé uniquement par des chiffres et des lettres de "A" à "F". Pour convertir **N** de la base hexadécimale à la base décimale, on utilise la séquence d'instructions :

$X \leftarrow 0$ $P \leftarrow 1$ Pour i de $\text{long}(N)-1$ à 0 (pas $=-1$) Faire Si $\text{Estnum}(N[i])$ alors $X \leftarrow X + P * \text{valeur}(N[i])$ Sinon $X \leftarrow X + P * (\text{ord}(N[i]) - 55)$ FinSi $P \leftarrow P * 16$ Fin Pour	$X \leftarrow 0$ Pour i de 0 à $\text{long}(N)-1$ Faire Si $\text{Estnum}(N[i])$ alors $X \leftarrow 16 * X + \text{valeur}(N[i])$ Sinon $X \leftarrow 16 * X + (\text{ord}(N[i]) - 55)$ FinSi Fin Pour	$Ch \leftarrow ""$ Répéter $R \leftarrow N \text{ Mod } 16$ Si $R < 10$ alors $Ch \leftarrow \text{Convch}(R) + Ch$ Sinon $Ch \leftarrow \text{Chr}(R+55) + Ch$ FinSi $N \leftarrow N \text{ Div } 16$ Jusqu'à $N = 0$
--	--	---

Exercice 2 (3 points)

Au mois de janvier 2022, un jeune agriculteur a décidé de créer son propre projet d'élevage de lapins, pour cela il a acheté un couple de lapereaux (petits lapins). Ces lapins ont quelques caractéristiques spéciales :

- Un couple de lapereaux passe un mois pour devenir jeune.
- Un jeune couple de lapin donne naissance chaque début de mois à un nouveau couple de lapereaux.

Au mois de février, ce couple de lapereaux devient jeune et peut donner naissance à un couple de lapereaux le mois suivant.

Au mois de mars, ce couple de jeunes lapins donne naissance à un nouveau couple de lapereaux. Le nombre de couples est égal à 2.

Au mois d'avril, le premier couple donne naissance à un nouveau couple de lapereaux et le couple de lapereaux du mois de mars devient jeune pour donner naissance à des petits le mois suivant. Le nombre de couples est devenu égal à 3.

Pour calculer le nombre de lapins qui seront produits pendant n mois (avec $n \geq 3$), le jeune agriculteur a utilisé la suite arithmétique U définie par :

$$\begin{cases} U_1 = 1 \\ U_2 = 1 \\ U_n = U_{n-1} + U_{n-2} \text{ pour } n \geq 3 \end{cases}$$

Travail demandé

- 1) Écrire l'algorithme d'un module récursif **Nombretotal(n)** permettant de retourner le nombre total de lapins produits pendant n mois.
- 2) Écrire l'algorithme d'un module **NombreMois(X)** permettant de retourner le nombre de mois nécessaire pour que le nombre total de lapins produits atteigne la valeur X . (avec $X \geq 2$ saisi au niveau du programme appelant). Pour ce faire, utiliser la fonction **Nombretotal(n)**.
- 3) En se référant au tableau ci-dessous, donner le nombre de lapins qui seront produits le **01/02/2023** sachant que le mois de janvier est le premier mois d'une année.

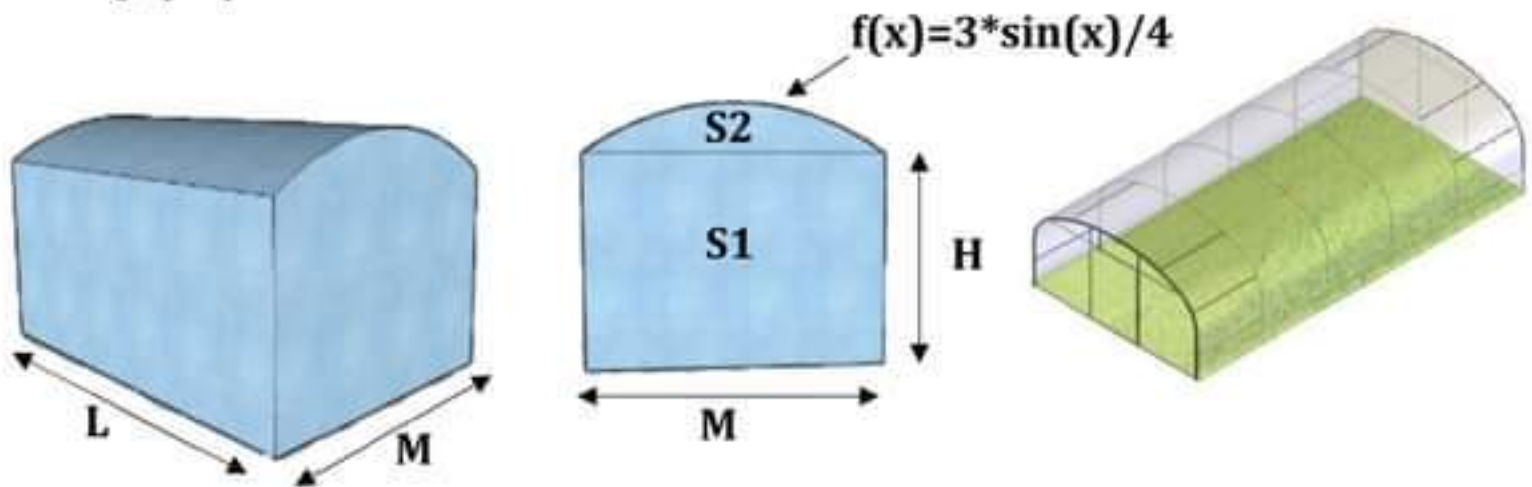
n	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
U_n	1	1	2	3	5	8	13	21	34	55	89	144	233	377	610	987

- 4) Écrire l'algorithme d'un module **Total(D)** permettant de retourner le nombre total de lapins produits à une date donnée **D** sous la forme "jj/mm/aaaa". Pour ce faire, utiliser la fonction **Nombretotal(n)**.

Exercice 3 (4 points)

Dans le cadre de l'encouragement à l'utilisation des serres agricoles, l'État fixe des aides (des primes) aux agriculteurs pratiquant ce type d'agriculture. Ces aides incitatives sont valorisées en fonction du volume de serres utilisées par chaque agriculteur.

Une serre agricole est de longueur L , de largeur M , de hauteur H , et sa forme est obtenue par la représentation graphique suivante :



Le volume d'une serre est égal à $(S1+S2) * L$, avec :

- La surface $S1 = H * M$,
- La surface $S2$ est obtenue par la représentation graphique de la fonction $f(x) = (3 * \sin(x)) / 4$ sachant que $x \in [0..M]$.

Travail demandé

- 1) Écrire l'algorithme d'une fonction qui permet de calculer une valeur approchée de $\sin(x)$ à 10^{-8} près en utilisant la formule suivante :

$$\sin(x) = \frac{x}{1!} - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \frac{x^9}{9!} - \dots$$

- 2) Pour aider les paysans à calculer la valeur de leurs primes, écrire l'algorithme d'un module qui permet de calculer une valeur approchée du volume d'une serre pour un nombre de divisions N donné.

N.B. : L , M , H et N sont passés en paramètres.

Problème (10 points)

L'internet a fait évoluer les modes de consommation. En effet et depuis ces dernières décennies, l'achat en ligne et la livraison à domicile sont en plein essor.

La livraison à domicile constitue un mode de livraison qui offre une rapidité de réception du produit grâce à la prise en charge par un transporteur professionnel.

Lorsqu'un transporteur se rend d'une ville à une autre dans un réseau routier, il choisit le plus court chemin, c'est-à-dire celui dont la distance est la plus petite.

On se propose d'aider un transporteur à trouver le meilleur chemin pour se rendre à partir d'une ville de départ à une ville d'arrivée dans une région composée de 10 villes représentées par des lettres majuscules de "A" à "J".

Pour ce faire, on se propose d'écrire un programme qui permet de :

- remplir un fichier texte "Distance.txt" par la distance entre chaque couple de villes sans redondance sachant que le couple (A, B) est identique au couple (B, A) et la distance est un entier strictement positif. Chaque ligne du fichier aura la forme : (Ville1,Ville2)=Distance (on suppose qu'une ligne du fichier ne contient aucun espace).

- saisir la ville de départ et la ville d'arrivée représentées par deux lettres majuscules distinctes de l'intervalle ["A".."J"],
- remplir un fichier d'enregistrements "**Chemin.dat**" par **N** chemins **différents** (avec $3 \leq N \leq 7$) et la distance parcourue pour chaque chemin.

Chaque enregistrement est composé de deux champs :

- **Chemin** : un chemin présenté par une suite de lettres majuscules **distinctes** qui commence par la lettre de la ville de départ et se termine par la lettre de la ville d'arrivée.
 - **Distance** : La distance d'un chemin est la somme des distances de passage des différentes villes qui la composent en se référant au fichier "**Distance.txt**".
- afficher les **trois** meilleurs chemins (les distances les plus faibles).

Exemple

Pour $N = 5$, le contenu du fichier "**Distance.txt**" ci-dessous et le contenu du fichier "**Chemin.dat**" suivant :

ABDF	ABCEDF	ACEF	ACEDF	ABCEF
23	36	18	34	20

Le programme affichera :

Les trois meilleurs chemins sont : **ACEF**, **ABCEF** et **ABDF**

Le fichier "**Distance.txt**"

(A,B) = 5
 (A,C) = 5
 (A,D) = 10
 (A,E) = 9
 (A,F) = 18
 (A,G) = 7
 (A,H) = 15
 (A,I) = 25
 (A,J) = 10
 (B,C) = 2
 (B,D) = 8
 ...
 (C,E) = 10
 ...
 (D,E) = 9
 (D,F) = 10
 ...
 (E,F) = 3

Explication du résultat

La distance parcourue du chemin **ABDF** est égale à **23**
 En effet (A, B)=5, (B, D)=8 et (D, F)=10 donc $5+8+10 = 23$

La distance parcourue du chemin **ABCEDF** est égale à **36**
 En effet (A, B)=5, (B, C)=2, (C, E)=10, (E, D)=9 et (D, F)=10 donc $5+2+10+9+10 = 36$

La distance parcourue du chemin **ACEF** est égale à **18**
 En effet (A, C)=5, (C, E)=10 et (E, F)=3 donc $5+10+3 = 18$

La distance parcourue du chemin **ACEDF** est égale à **34**
 En effet (A, C)=5, (C, E)=10, (E, D)=9 et (D, F)=10 donc $5+10+9+10 = 34$

La distance parcourue du chemin **ABCEF** est égale à **20**
 En effet (A, B)=5, (B, C)=2, (C, E)=10 et (E, F)=3 donc $5+2+10+3 = 20$

On vous demande d'élaborer :

- 1) un algorithme du programme principal, solution à ce problème, en le décomposant en modules,
- 2) l'algorithme de chaque module.