

RÉPUBLIQUE TUNISIENNE MINISTÈRE DE L'ÉDUCATION □□□□ EXAMEN DU BACCALAURÉAT SESSION 2015 Section : Sciences de l'informatique	Épreuve : ALGORITHMIQUE ET PROGRAMMATION
	Durée : 3 h
	Coefficient : 2,25
	Session principale

Section : N° d'inscription : Série :
 Nom et prénom :
 Date et lieu de naissance :

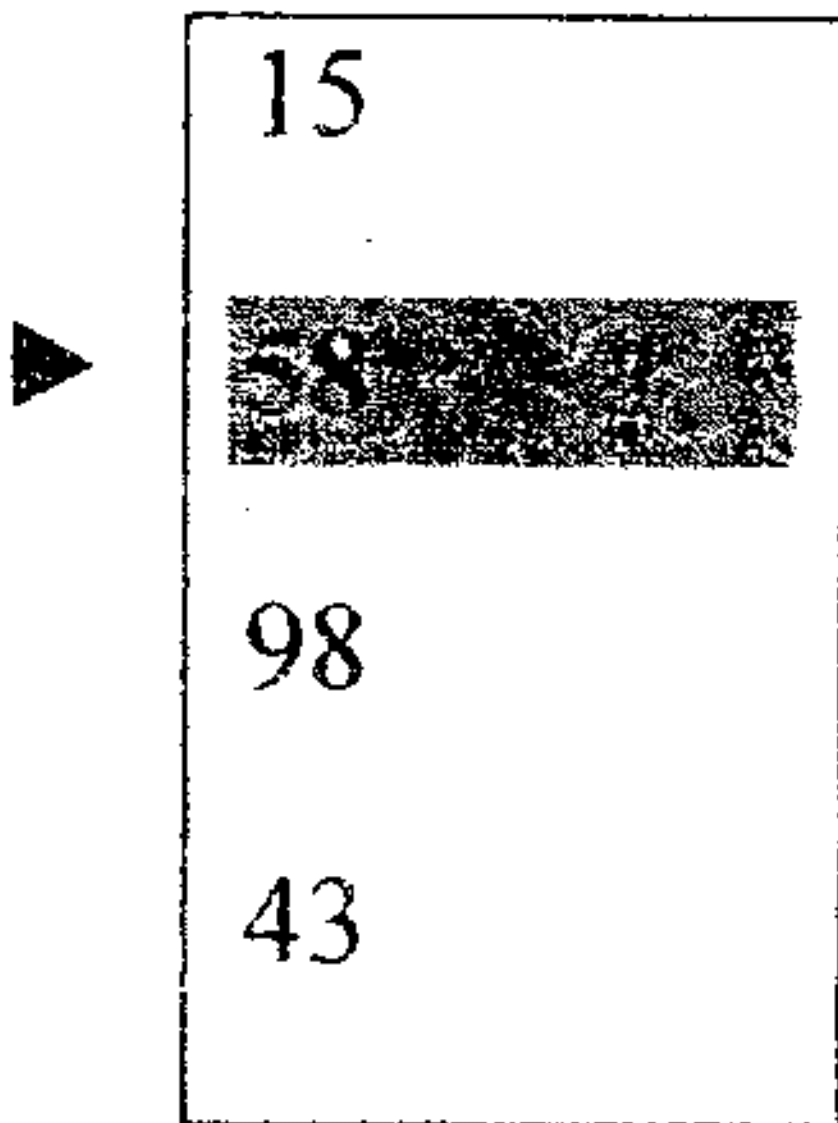
Signatures des
surveillants

Le sujet comporte 4 pages numérotées de 1/4 à 4/4.
 Les réponses à l'exercice1 doivent être rédigées sur les pages 1/4 et 2/4
 qui doivent être remises avec la copie

Exercice 1 (5,25 points)

Dans un contexte informatique et pour chacune des propositions données ci-dessous, mettre dans chaque case, la lettre **V** si la proposition est correcte, ou la lettre **F** dans le cas contraire.

1) Soit un fichier d'entiers, ayant comme nom logique **F**. On suppose que le pointeur est positionné sur le deuxième entier comme indiqué ci-dessous.



N.B : Toutes les instructions données dans a), b) et c) sont valides.

a) Le contenu de **X** après l'exécution de l'instruction **Lire(F , X)** est :

☐ 15

☐ 58

☐ 98

b) L'instruction **Pointer(F , 3)** permet de positionner le pointeur sur l'entier :

☐ 58

☐ 98

☐ 43

c) Le contenu de **Y** après l'exécution de l'instruction **Y ← Fin_fichier(F)** est :

☐ 43

☐ 4

☐ faux

Ne rien écrire ici

2) Soit la suite U définie par:
$$\begin{cases} U_0 = 1 \\ U_n = 2 * U_{n-1} + n \end{cases} \quad (\text{avec } n \text{ un entier supérieur ou égal à } 1)$$

a) U est une suite récurrente d'ordre :

☐ 1

☐ 2

☐ 5

b) Le 3^{ème} terme de la suite U (U_2) est égal à :

☐ 5

☐ 8

☐ 9

c) L'algorithme permettant de calculer U_n (avec $n \geq 1$) est :

☐ 0) Def FN terme (n : entier) : entier
1) $t[0] \leftarrow 1$
2) Pour i de 1 à n faire
 $t[i] \leftarrow 2 * t[i-1] + n$
 Fin pour
3) terme $\leftarrow t[n]$
4) Fin terme

☐ 0) Def FN terme(n : entier) : entier
1) Si $n=0$ alors terme $\leftarrow 1$
 Sinon
 terme $\leftarrow 2 * \text{FN terme}(n-1) + n$
 Fin si
2) Fin terme

☐ 0) Def FN terme(n : entier) : entier
1) $Up \leftarrow 1$
2) Pour i de 2 à n faire
 $Up \leftarrow 2 * Up + i$
 Fin pour
3) terme $\leftarrow Up$
4) Fin terme

Exercice 2 (3 points)

En arithmétique, un **auto-nombre** est un entier naturel N qui ne peut pas s'écrire sous la forme d'un nombre M ajouté à la somme des chiffres de M .

Exemples :

- Pour $N = 21$,
 N n'est pas un auto-nombre, puisqu'il peut être généré à partir de la somme d'un nombre M égal à 15 et les chiffres qui le constituent (1 et 5) c'est-à-dire $21 = 15 + 1 + 5$.
- Pour $N = 20$,
 N est un auto-nombre puisqu'il ne peut pas être généré à partir de la somme d'un nombre M et les chiffres qui le constituent.

Travail demandé :

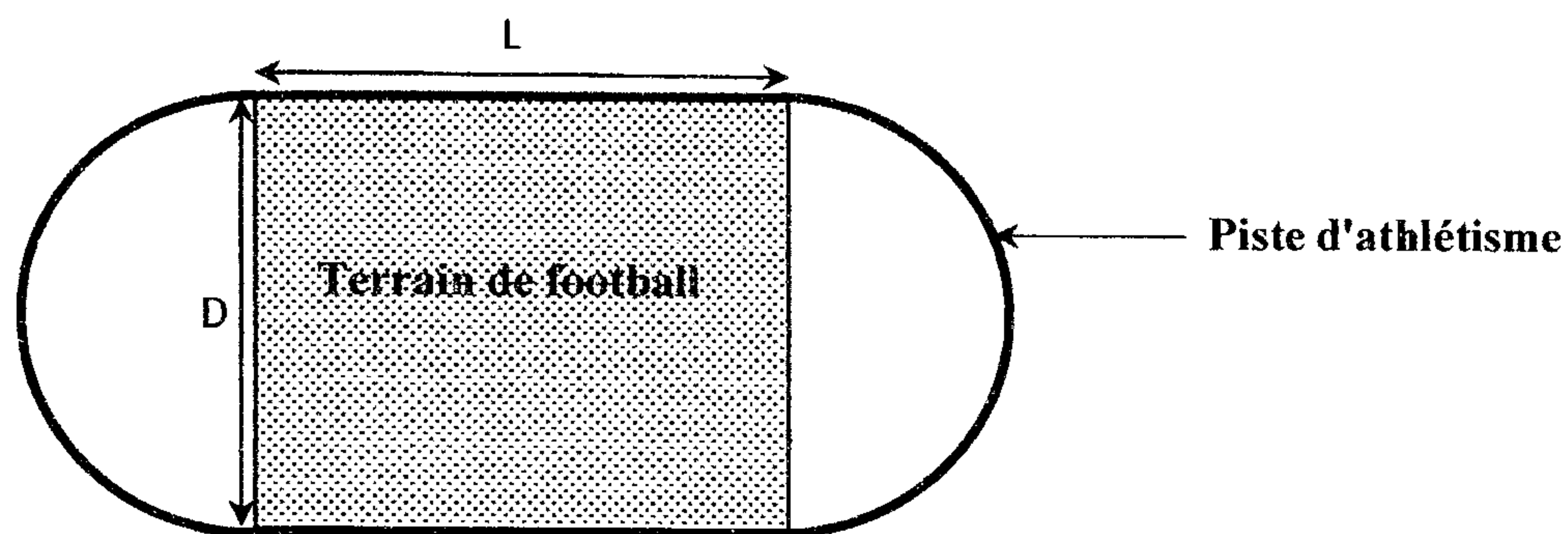
Ecrire une analyse d'un module intitulé **Verif_auto_nombre**, permettant de vérifier si un entier naturel N strictement positif est un **auto-nombre**, sachant que N est déjà saisi dans l'analyse du programme principal.

Exercice 3 (3 points)

La direction d'une association sportive veut construire un stade formé par une piste d'athlétisme et un terrain de football, tout en cherchant à **maximiser la surface** de ce dernier.

Le terrain de football est un rectangle de longueur L , de largeur D et de surface S .

La piste d'athlétisme est de longueur P et formée par les deux arrêtes parallèles du terrain de football (de longueur $2 * L$) et les deux demi-cercles de diamètre D (de longueur $\pi * D$), comme le montre le schéma suivant :



Puisque $S = L * D$ et $P = 2 * L + \pi * D$ alors $S = L * (P - 2 * L) / \pi$

Travail demandé :

Etant donné que L varie de 0 à $P/2$, écrire un **algorithme** d'une fonction qui permet de déterminer, à ϵ près, la longueur optimale L_{opt} correspondante à la surface maximale S_{max} du terrain, sachant que ϵ et P sont déjà saisis dans l'algorithme du programme principal.

Problème (8,75 points)

Un administrateur d'un site web veut assurer un maximum de sécurité pour les utilisateurs du site. Pour ceci il décide de réaliser une application qui évalue la force des mots de passe des différents utilisateurs du site, sachant qu'un mot de passe est une chaîne de caractères qui ne comporte pas d'espaces et de lettres accentuées.

La force d'un mot de passe varie, selon la valeur d'un score calculé, de "**Très faible**" jusqu'à "**Très fort**" :

- Si le score < 20 , la force du mot de passe est "**Très faible**"
- Sinon si le score < 40 , la force du mot de passe est "**Faible**"
- Sinon si le score < 60 , la force du mot de passe est "**Moyen**"
- Sinon si le score < 80 , la force du mot de passe est "**Fort**"
- Sinon la force du mot de passe est "**Très fort**"

Le score se calcule en additionnant des bonus et en retranchant des pénalités.



Les bonus attribués sont :

- Nombre total de caractères * 4
- (Nombre total de caractères – nombre de lettres majuscules) * 2
- (Nombre total de caractères – nombre de lettres minuscules) * 3
- Nombre de caractères non alphabétiques * 5

Les pénalités imposées sont :

- La longueur de la plus longue séquence de lettres minuscules * 2
- La longueur de la plus longue séquence de lettres majuscules * 2

Exemple :

Pour le mot de passe "**B@cSI_juin2015**", le score se calcule comme suit :

La somme des bonus = $14*4 + (14-3)*2 + (14-5)*3 + 6*5 = 135$

Car $\left\{ \begin{array}{l} \text{le nombre total de caractères} = 14 \\ \text{le nombre de lettres majuscules} = 3 \\ \text{le nombre de lettres minuscules} = 5 \\ \text{le nombre de caractères non alphabétiques} = 6 \end{array} \right.$

La somme des pénalités = $4*2 + 2*2 = 12$

Car $\left\{ \begin{array}{l} \text{la longueur de la plus longue séquence de lettres minuscules ("juin")} = 4 \\ \text{la longueur de la plus longue séquence de lettres majuscules ("SI")} = 2 \end{array} \right.$

Le score final = $135 - 12 = 123$; puisque $123 \geq 80$ alors le mot de passe est considéré comme "**Très fort**".

En disposant d'un fichier texte "**Motspass.txt**", situé sur la racine du disque **C**, dont chaque ligne contient un mot de passe, on se propose de :

- Générer un fichier d'enregistrements "**ForceMDP.dat**" où chaque enregistrement comporte le mot de passe lui-même, son score et sa force.
- Générer un fichier texte "**MDPfort.txt**" par la liste les mots de passe ayant une force égale à "**Très fort**" suivis de la liste des mots de passe ayant une force égale à "**Fort**" à raison d'un mot de passe par ligne, tout en séparant les deux listes par une ligne vide.

N.B : L'élève n'est pas appelé à remplir le fichier "**Motspass.txt**".

Travail demandé :

- 1- Analyser le problème en le décomposant en modules.
- 2- Analyser chacun des modules envisagés.