RÉPUBLIQUE TUNISIENNE MINISTÈRE DE L'ÉDUCATION ●●●●● **EXAMEN DU BACCALAURÉAT** SESSION **2018**	**Session de contrôle**	
	Épreuve : **Algorithmique et Programmation**	Section : **Sciences de l'informatique**
	Durée : **3h**	Coefficient de l'épreuve : **2.25**

Section : .... **N° d'inscription :** .... **Série :** ....

Nom et prénom : ....

Date et lieu de naissance : ....

Signatures des surveillants

....

....

Le sujet comporte 5 pages numérotées de 1/5 à 5/5.
Cette feuille doit être remise à la fin de l'épreuve.

Exercice 1 : (3 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.

a) Un module est dit récursif, s'il comporte dans son corps :

- ☐ au plus un appel à lui même
- ☐ au moins deux appels à lui même
- ☐ un ou plusieurs appels à lui-même

b) Une fonction récursive doit comporter :

- ☐ un appel récursif en changeant au moins la valeur d'un paramètre
- ☐ une condition d'arrêt de l'appel récursif
- ☐ des variables locales

c) Lors de l'exécution d'un traitement récursif :

- ☐ le dernier appel doit être traité en premier
- ☐ le premier appel doit être traité en premier
- ☐ le dernier appel doit être traité en dernier

d) Un traitement récurrent dépend :

- ☐ toujours d'un seul traitement précédent
- ☐ de zéro traitement précédent
- ☐ d'un ou de plusieurs traitement(s) précédent(s)

RÉPUBLIQUE TUNISIENNE MINISTÈRE DE L'ÉDUCATION ●●●●● **EXAMEN DU BACCALAURÉAT** SESSION **2018**	**Session de contrôle**	
	Épreuve : **Algorithmique et Programmation**	Section : **Sciences de l'informatique**
	Durée : **3h** ◆	Coefficient de l'épreuve : **2.25**

Important :

Dans tout ce qui suit, chaque solution sous forme d'un algorithme doit être accompagnée d'un tableau de déclaration des objets ayant la forme suivante :

Objet	*Type/Nature*	*Rôle*

Exercice 2 : (4,5 points)

Soient les deux fonctions **f** et **g** définies comme suit :

- $f(x) = x \quad avec\, x \in \mathbb{R}$
- $g(x) = \cos(x) \quad avec\, x \in \mathbb{R}$

1- Ecrire un algorithme d'une fonction **Calcul (epsilon)** permettant de calculer une valeur approchée, à epsilon près, de **p** tel que **cos(p) = p**.

2- Soit le graphique suivant représentant les courbes des deux fonctions f et g et de la droite $x = p$.

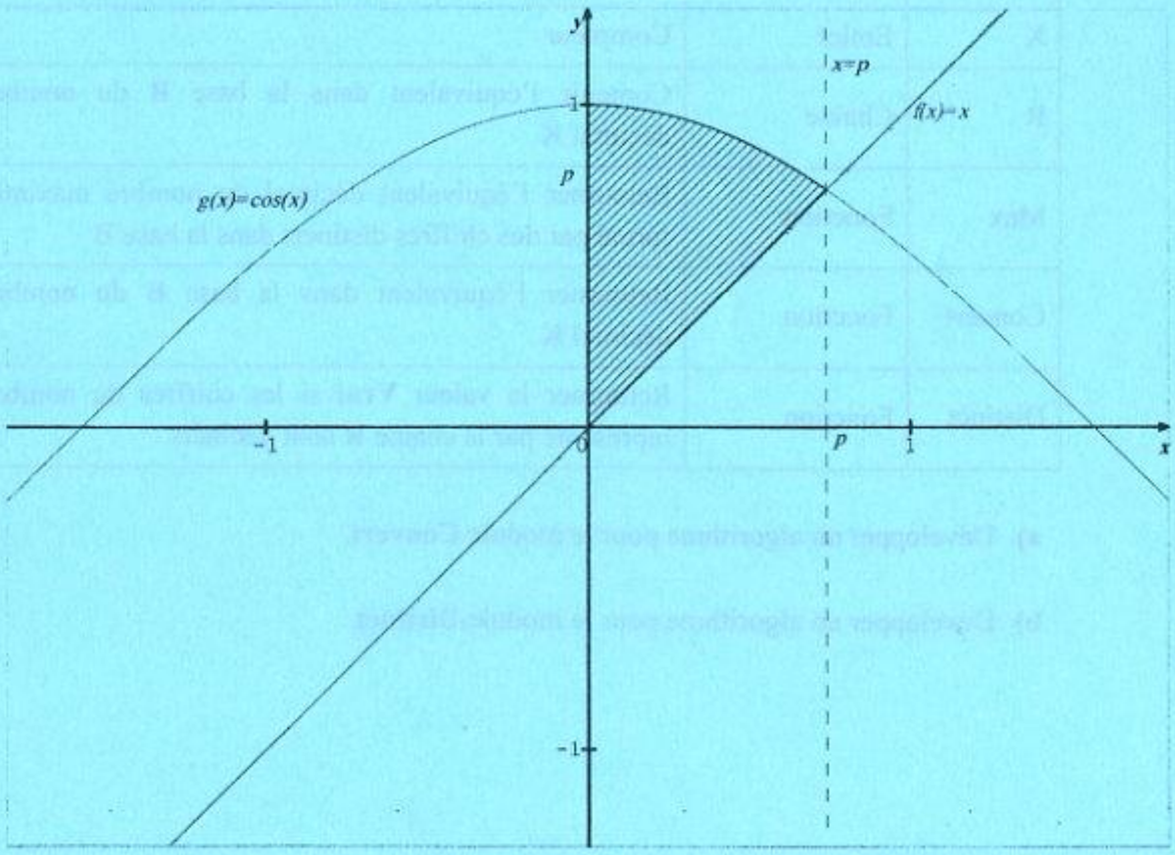

Figure 1

Ecrire un algorithme d'une fonction **Surface (epsilon)** qui permet de calculer une valeur approchée, à epsilon près, de l'aire délimitée par les deux courbes des deux fonctions **f** et **g**, l'axe des ordonnés et la droite **x = p** (l'aire hachurée dans *Figure 1*).

Exercice 3 : (4 points)

On se propose de trier un tableau **T** de **n** entiers (**avec n dans [5..20]**), en utilisant le **tri par insertion dichotomique** qui repose sur le même principe du tri par insertion, mais utilise la recherche dichotomique pour déterminer la position d'insertion.

On rappelle que le principe de tri par insertion consiste à rechercher, séquentiellement pour chaque élément d'un tableau, sa position d'insertion dans la portion du tableau qui le précède, de décaler si c'est nécessaire et de l'insérer pour que cette portion du tableau reste triée et ce en commençant de l'élément numéro **2** jusqu'au dernier.

1- Soit l'algorithme incomplet de la fonction récursive **Dichotomie** ci-dessous permettant de retourner la position d'insertion d'un entier **k** dans la portion triée dans l'ordre croissant et délimitée par les indices **g** et **d** d'un tableau **T** d'entiers :

0) DEF FN Dichotomie (T : Tab ; g, d, k : Octet) : Octet
1) ....
    Si g>d Alors Dichotomie ← g
    Sinon Si T[mil] = k Alors Dichotomie ← mil
    Sinon Si T[mil] > k Alors ....
    Sinon Si T[mil] < k Alors ....
    FinSi
2) Fin Dichotomie

a) Réécrire l'algorithme de la fonction **Dichotomie** et le compléter en plaçant les instructions ci-après aux bons endroits.

- ***Dichotomie ← Dichotomie (T, g, mil-1, k)***
- ***mil ← (g+d) DIV 2***
- ***Dichotomie ← Dichotomie (T, mil+1, d, k)***

b) Dresser le tableau de déclaration du nouveau type **Tab**.

c) Dresser le tableau de déclaration des variables locales de la fonction **Dichotomie**.

2- En utilisant la fonction **Dichotomie** définie précédemment, écrire un algorithme d'un module qui permet de trier un tableau **T** de **n** entiers dans l'ordre croissant en appliquant la méthode de **tri par insertion dichotomique**.

Exercice 4 : (4 points)

Soit un circuit série formé par un générateur de tension constante, un interrupteur **K**, une bobine d'inductance **L** et de résistance interne **r** et un résistor de résistance $\mathbf{R_0}$.

Afin de vérifier l'expression $i = I_0\left(1 - e^{-\frac{t}{\tau}}\right)$ qui détermine la valeur de l'intensité **i** à un instant **t**, un expérimentateur a stocké dans un fichier d'enregistrements **F_intens.dat** situé sur la racine du disque **C**, les mesures effectuées de l'intensité du courant **i** à plusieurs instants **t**. Chaque enregistrement est formé de deux champs, le premier comporte la valeur de **t** et le deuxième comporte la valeur de **i**.

On se propose d'écrire un programme qui permet de vérifier le degré de réussite de l'expérience et ce en comparant les valeurs expérimentales de **i** avec celles calculées théoriquement en utilisant la formule précédente. L'expérience sera dite réussie si la différence entre la valeur expérimentale et la valeur théorique ne dépasse pas $\mathbf{10^{-3}}$ dans **90%** des mesures.

Travail demandé :

1- Ecrire une instruction d'assignation du fichier **F_intens.dat** à une variable logique **F**.

2- Donner une déclaration pour le type du fichier **F** ainsi que pour tout nouveau type nécessaire à sa déclaration.

3- Ecrire l'algorithme d'un module qui permet de remplir le fichier **F_intens.dat** par les valeurs expérimentales. La saisie se termine en répondant par **"N"** à la question " **Voulez vous saisir les valeurs d'une expérience (O/N) ?** ".

4- Ecrire l'algorithme d'un module qui permet de vérifier si l'expérience est réussie ou non.
On donne E=6V, L=300mH, r=10Ω et R0=140Ω avec R=R0 + r, $\tau = \frac{L}{R}$ et $I_0 = \frac{E}{R}$

NB :

- En remplaçant **E**, **L**, $\mathbf{R_0}$ et **r** par leurs valeurs, l'expression de l'intensité devient :
$$i = \left(1 - e^{-\frac{t}{2}}\right)/25\,.$$
- Pour calculer l'exponentiel $\mathbf{e^x}$, on utilise la fonction prédéfinie **exp(x)**.

Exercice 5 : (4.5 points)

Dans une base **B**, un nombre est dit distinct s'il est composé par des chiffres distincts.

Exemple :

Dans la base **B = 2**, il y a trois nombres distincts qui sont : **0**, **1** et **10**

1- Donner tous les nombres distincts dans la base **3**.

2- On présente ci-dessous l'algorithme d'une procédure **Nbre_Distincts** qui permet d'afficher tous les nombres distincts d'une base **B**.

0) **Def Proc Nbre_Distincts (B : Octet)**

1) *Pour K de 0 à **Fn Max(B)** Faire*

   *R ← **Fn Convert(K, B)***

   *Si **Fn Distinct(R)** Alors Ecrire(R)*

   Fin Si

   Fin Pour

2) **Fin Nbre_Distincts**

Tableau de déclaration des objets locaux

Objet	Type/Nature	Rôle
K	Entier	Compteur
R	Chaîne	Contenir l'équivalent dans la base **B** du nombre décimal **K**
Max	Fonction	Retourner l'équivalent décimal du nombre maximal formé par des chiffres distincts dans la base B
Convert	Fonction	Retourner l'équivalent dans la base **B** du nombre décimal **K**
Distinct	Fonction	Retourner la valeur **Vrai** si les chiffres du nombre représenté par la chaîne **R** sont distincts

a) Développer un algorithme pour le module **Convert**.

b) Développer un algorithme pour le module **Distinct**.