

Examen du baccalauréat session juin 2016
Correction du sujet théorique d'informatique

SECTIONS : Mathématiques + Sciences Expérimentales + Sciences Techniques

Exercice 1 : (3points= 0,25*3*4)

Valider chacune des propositions ci-dessous en mettant dans la case correspondante la lettre (V) si elle est correcte ou la lettre (F) dans le cas contraire.

1) L'identificateur d'une variable :

- V ne doit pas commencer par un chiffre.
- F peut contenir un espace.
- V peut contenir le caractère souligné (tiret bas "_").

2) L'instruction qui permet d'affecter à une variable X, une valeur aléatoire de l'intervalle [2,10] est :

- F $X \leftarrow 2 + \text{Aléa}(10)$
- V $X \leftarrow 2 + \text{Aléa}(9)$
- F $X \leftarrow \text{Aléa}(2 + 10)$

3) Une structure de données tableau peut :

- F contenir des éléments de types différents.
- F être déclarée avec une taille maximale variable.
- V avoir des indices de cases de type caractère.

4) Pour le typescalaire énuméré :

- F les valeurs énumérées peuvent appartenir à un type prédéfini.
- V une valeur énumérée peut être affectée à une variable du même type.
- V les opérateurs relationnels "<", ">" et "=" peuvent être appliqués.

Exercice 2 (5 points)

1) A partir des algorithmes donnés ci-dessus, remplir la 2^{ème} colonne du tableau suivant par un exemple de chaque élément cité dans la 1^{ère} colonne : (1points = 4 * 0.25)

Elément	Exemple
Expression booléenne	$(C = \text{" "})$ ou $(E \neq 0)$
	$E=0$
	$C = \text{" "}$
	$E \neq 0$
Procédure prédéfinie	Valeur(C[1],x,e)
	Efface(C,1,1)
Paramètre formel	C
Paramètre effectif	A

Remarque : On accepte aussi FN inconnue(A) comme un exemple d'expression booléenne

2) Compléter l'entête de la fonction **Inconnue** par les types appropriés : (0,5 point= 0.25*2)

Def Fn **Inconnue** (C : *Chaîne*) : *Booléen*

3) Compléter le tableau de déclaration ci-dessous par les types des objets locaux de la fonction **Inconnue**. (0,5 point = 0,25*2)

<i>Objet</i>	<i>Type / Nature</i>
X	<i>Entier / Réel</i>
E	<i>Entier</i>

4) Parmi les variables **A, C, X** et **E**, réécrire dans le tableau ci-dessous celles qui ne sont pas visibles par le programme principal. (0.5 point)
(-0.25 par erreur)

Variables non visibles par le programme principal
C, X, E

5) Donner le résultat affiché par le programme **Exercice** pour chacune des valeurs de **A** suivantes :
(2 points = 4 * 0.5)

- A = "523" → 523 Vérifie la propriété.
- A = "-523" → -523 Ne vérifie pas la propriété.
- A = "5,23" → 5,23 Ne vérifie pas la propriété.
- A = "A5B3" → A5B3 Ne vérifie pas la propriété.

6) En déduire le rôle de la fonction **Inconnue**. (0.5 point)

La fonction **Inconnue** permet de vérifier si une chaîne est formée uniquement par des chiffres.

Problème : (12 points)

1°) Analyse du programme principal

Résultat = PROC Affiche (N, Tx, Ty)

(Tx, Ty) = PROC Tri (N, Tx, Ty, Td)

Td = PROC Distance (N, Xd, Yd, Tx, Ty, Td)

(N, Xd, Yd, Tx, Ty) = PROC Saisie (N, Xd, Yd, Tx, Ty)

Tableau de déclaration des nouveaux types

Type
Tab = tableau de 50 réels

Tableau de déclaration des objets globaux

Objet	Type / Nature	Rôle
N	Entier	Nombre de bateaux les plus proches
Xd, Yd	Réel	Coordonnées du bateau en détresse
Tx	Tab	Tableau des abscisses des bateaux
Ty	Tab	Tableau des ordonnées des bateaux
Td	Tab	Tableau des distances

Affiche	Procédure	Procédure qui permet de faire l'affichage
Tri	Procédure	Procédure qui permet de faire le tri de Tx, Ty et Td
Distance	Procédure	Procédure qui permet de remplir le tableau des distances Td
Saisie	Procédure	Procédure qui permet de faire la saisie de N, Xd, Yd, Tx et Ty

2°) Analyses des modules envisagés

Analyse de la procédure Saisie

```

DEF PROC Saisie(Var N : entier ; VAR Xd,Yd : réel ; Var Tx,Ty : Tab)
Résultat = (Xd,Yd,N,Tx,Ty)
Xd,Yd = Donnée ("Introduire les coordonnées du bateau en détresse : ")
N=[]Répéter
    N=donnée ("Donner le nombre de bateaux les plus proches : ")
Jusqu'à N dans [1..50]
(Tx ,Ty) = [Répéter
    Tx[1] ,Ty[1] = Donnée("Introduire les coordonnées du bateau proche n° 1: ")
Jusqu'à ((Tx[1]<>Xd) OU (Ty[1]<>Yd))]
Pour i de 2 à N faire
    Répéter
        Tx[i] ,Ty[i] = Donnée("Introduire les coordonnées du bateau proche n° ",i," : ")
        Jusqu'à (Non (FNExiste(Tx[i],Ty[i],Tx,Ty,i)) ) ET((Tx[i]<>Xd) OU (Ty[i]<>Yd))
    Fin Pour
Fin Saisie

```

Tableau de déclaration des objets locaux de la procédure Saisie

Objet	Type / Nature	Rôle
i	Entier	Compteur
Existe	Fonction	Fonction qui permet de vérifier l'unicité des coordonnées.

Analyse de la fonction Existe:

```

DEF FnExiste (x, y: réel ; T1,T2 : tab ; p : entier): booléen
Résultat= existe ← test
test =[i←0] Répéter
    i←i+1
    test←(T1[i]=x) et (T2[i]=y)
    Jusqu'à (i=p-1) ou (test=Vrai)
Fin Existe

```

Tableau de déclaration des objets locaux

Objet	T/N
i	Entier
test	booléen

Analyse de la procédure Distance

```

DEF PROC Distance (N : entier ; Xd,Yd : réel ; Tx,Ty : Tab ; Var Td : Tab)
Résultat = Td
Td= [] Pour i de 1 à N Faire
    Td[i] ←Racine_carré (carrée (Tx[i]-Xd) + carrée (Ty[i]-Yd))
    Fin Pour
Fin Distance

```

Tableau de déclaration des objets locaux de la procédure Distance

Objet	Type / Nature	Rôle
i	Entier	Compteur

Analyse de la procédure Tri

```

DEF PROC Tri (N :entier ; VarTx,Ty,Td : Tab)
Résultat = (Tx,Ty,Td)
(Tx,Ty,Td)=[ ] Répéter
    Test ←Faux
    Pour i de 1 à N-1 Faire
        Si Td[i] >Td[i+1] Alors
            PROC Permut (Td[i],Td[i+1])
            PROC Permut (Tx[i],Tx[i+1])
            PROC Permut (Ty[i],Ty[i+1])
            Test ←Vrai
        Fin Si
    FinPour
    N← N-1
    Jusqu'à (Test = Faux) OU (N=1)
Fin Tri

```

Tableau de déclaration des objets locaux de la procédure Tri

Objet	Type / Nature	Rôle
i	Entier	Compteur
Test	Booléen	Tester s'il y a eu une permutation ou non
Permut	Procédure	Procédure qui permet de faire la permutation de deux cases.

Analyse de la procédure Permut

```

DEF PROC Permut(Varx,y : réel)
Résultat = (x,y)
(x,y) = aux ← x
    x ← y
    y← aux
Fin Permut

```

Tableau de déclaration des objets locaux de la procédure Permut

Objet	Type / Nature	Rôle
aux	réel	Variable auxiliaire

Analyse de la procédure Affiche

```

DEF PROC Affiche (N : entier ; Tx,Ty : Tab)
Résultat = [ ] Pour i de 1 à N faire
    Ecrire ("(",Tx[i],",", " ", Ty[i], ") " )
Fin Pour
Fin Affiche

```

Tableau de déclaration des objets de la procédure Affiche

Objet	Type / Nature	Rôle
i	Entier	Compteur

Barème : 12 points

Toute solution équivalente sera acceptée

-0.25 par type d'erreur

Action	Nombre de points
Programme principal : <ul style="list-style-type: none">• Modularité• Cohérence (mode de passage, conformité entre nombre, ordre et type des paramètres)	2 points : <ul style="list-style-type: none">• 1• $1 = 0.25 \times 4$
Saisie de n: <ul style="list-style-type: none">• Lecture• Contrôle	0.5 point : <ul style="list-style-type: none">• 0.25• 0.25
Saisie de Xd et Yd:	0.5 point = 0.25+0.25
Saisie de Tx et Ty: <ul style="list-style-type: none">• Initialisation (lecture du premier élément)• Parcours• lecture d'un élément de Tx et un élément de Ty• Contrôle de saisie<ul style="list-style-type: none">○ Test d'existence + Traitement de Existe○ Test de différence avec Xd et Yd	2.5 points : <ul style="list-style-type: none">• 0.25• 0.5• $0.25 + 0.25$• $0.25 + 0.75$• 0.25
Remplissage de Td: <ul style="list-style-type: none">• parcours• calcul	1 point : <ul style="list-style-type: none">• 0.5• 0.5
Tri : <ul style="list-style-type: none">• Parcours• Comparaison• permutation<ul style="list-style-type: none">○ dans Tx○ dans Ty○ dans Td	3 points : <ul style="list-style-type: none">• $1 = 0.5 + 0.5$• 0.5• 0.5• 0.5• 0.5
Affichage : <ul style="list-style-type: none">• Parcours• Ecriture	1.5 points <ul style="list-style-type: none">• 0.5• $1 = 0.5 + 0.5$
TDNT	0.25 point
TDOG	0.5 point
TDOL	0.25 point