

**Examen du baccalauréat session juin 2013  
Correction du sujet théorique d'informatique**

**SECTIONS : Mathématiques + Sciences Expérimentales + Sciences Techniques**

**Exercice 1 : (3 points = 4 \* 3 \* 0.25)**

Pour chacune des instructions Pascal suivantes, valider chaque proposition en mettant dans la case correspondante la lettre **V** si elle est correcte ou **F** dans le cas contraire.

**a. C ← Sous\_chaine ("Baccalauréat",4,1)**

Elle permet d'affecter le caractère "c" à la variable C.  **V**

La variable C doit être déclarée de type caractère.  **F**

La variable C doit être déclarée de type chaîne.  **V**

**b. X ← Aléatoire (6) + 4**

Permet d'affecter à la variable X une valeur aléatoire de l'intervalle [4,6]  **F**

Permet d'affecter à la variable X une valeur aléatoire de l'intervalle [4,10]  **F**

Permet d'affecter à la variable X une valeur aléatoire de l'intervalle [4,9]  **V**

**c. R ← Arrondi (12.5)**

Permet d'affecter à la variable R l'entier 12  **F**

Permet d'affecter à la variable R l'entier 13  **V**

Permet d'affecter à la variable R le réel 13.0  **F**

**d. C ← Majus("??")**

Permet d'affecter à la variable C le caractère '?' en gras  **F**

Permet d'affecter à la variable C le caractère '?'  **V**

La variable C doit être de type Caractère  **V**

**Exercice 2:(2 points = 4\*0.5)**

Soit la partie déclarative suivante d'un programme Pascal :

**Program** Composer ;

**CONST** mot1 = 'informatique' ; mot2 = '3D' ;

**VAR** mot3, mot4 : string ; n, m : integer ;

En utilisant des fonctions et des procédures prédéfinies, donner les instructions Pascal permettant de réaliser les traitements suivants :

a- A partir de la constante **mot1**, mettre dans la variable **mot3** le terme "format".

.....**mot3 := copy (mot1,3,6) ;** .....

b- A partir de **mot3** et **mot2**, mettre dans la variable **mot4** le terme "format 3D".

.....**mot4 := concat ( mot3 , ' ', mot2 ) ;** .....

On n'acceptera pas la concaténation par "+"

c- Mettre dans **n** la longueur de la chaîne **mot4**

.....**n := length (mot4) ;** .....

d- A partir de la constante **mot2**, mettre dans **m** la valeur 3.

.....**Val (copy(mot2,1,1),m,n) ;** .....

Si le nom de la fonction ou de la procédure est incorrect alors on attribue 0

Si non 0,25 pour le nom de la fonction ou de la procédure en Pascal et 0,25 pour l'affectation et les paramètres

### Exercice 3 : (3 Points= 0.75+0.75+0.25\*6)

Soit le type **Examen** contenant les valeurs suivantes :

**Math, Anglais, Physique et Informatique**

1. Qu'appelle-t-on le type **Examen** décrit ci-dessus ?

Type scalaire énuméré (0,75)

Type utilisateur (0,5)

Type scalaire (0,5)

Type énuméré (0,5)

2. Proposer une déclaration Pascal du type **Examen** en respectant l'ordre des valeurs proposé ci-dessus.

Examen= (Math, Anglais, Physique, Informatique) ;

(0,75 : -0,25 par type d'erreur)

3. Compléter le tableau ci-dessous par les types et les valeurs des variables A, B et C après exécution des instructions suivantes :

A := PRED (Informatique) ;

B := ORD (Anglais) \* 8 DIV 4 ;

C := (Math < Physique) ;

Variable	Type	Valeur
A	Examen	Physique
B	Tout type numérique	2
C	Boolean / Booléen	True

0,25\*6

### Problème : (12 points)

Analyse du programme principal :

Résultat = Proc affiche(T<sub>f</sub>,n)

T<sub>f</sub> = Proc Tri (T<sub>i</sub>,n)

T<sub>i</sub> = Proc Remplissage (T<sub>i</sub>,n)

N = Proc saisie(n)

NB : T<sub>f</sub> représente l'état final du tableau T  
T<sub>i</sub> représente l'état initial du tableau T

### T.D.N.T

Type
Tab = tableau de 50 entiers

### T.D.O.G

Objet	Type /Nature	Rôle
T	Tab	Tableau à trier
N	Entier	Nombre d'éléments du tableau
Saisie	Procédure	Permet de saisir le nombre d'éléments du tableau T
Remplissage	Procédure	Permet de remplir le tableau T

Tri	Procédure	Permet de trier le tableau T
affiche	Procédure	Permet d'afficher le tableau T après tri

### Analyse de la procédure saisie

**DEF PROC** saisie (var n : entier)

Résultat= n

n=[ ]répéter

n= donnée( "saisir le nombre d'entiers : ")

jusqu'à ( n dans [6..50] )

**Fin** saisie

### Analyse de la procédure remplissage

**DEF PROC** remplissage (var T : tab ; n : entier)

Résultat= T

T=[ ]Pour i de 1 à n faire

T [ i ]= donnée ( " Donner T[" , i , " ] : ")

FinPour

**Fin** remplir

### T.D.O.L

Objet	T/N	Rôle
i	Entier	Compteur

### Analyse de la procédure tri

**DEF PROC** tri (var T: tab; n: entier)

Résultat= T

T=[ j ← 0 ] Répéter

Permut ← faux

j ← j+1

Pour i de j à n-1 faire

Si (T[i]>T[i+1]) alors

Permut ← vrai

aux ← T[i]

T[i] ← T[i+1]

T[i+1] ← aux

FinSi

FinPour

Si (permut = vrai) Alors

Permut ← faux

Pour i de n-1 à j+1 faire

Si (T[i]<T[i-1]) alors

Permut ← vrai

Aux ← T[i]

T[i] ← T[i-1]

T[i-1] ← aux

FinSi

FinPour

FinSi

n ← n-1

Jusqu'à (permut = faux) ou (j ≥ n)

**Fin** trier

### T.D.O.L

Objet	T/N	Rôle
i , j	Entier	Compteur
aux	Entier	Variable auxiliaire
permut	booléen	Test de permutation

**Analyse de la procédure affiche**  
**DEF PROC** affiche (T : tab ; n : entier)  
**Résultat**= [ ]Pour i de 1 à n faire  
                   Ecrire( T [ i ] )  
                   FinPour

**Fin affiche**

**T.D.O.L**

Objet	T/N	Rôle
I	Entier	Compteur

**Barème :**

*Toute solution équivalente sera acceptée*  
*-0,25 par type d'erreur*

Action	Nombre de points
<b>Programme principal :</b> <ul style="list-style-type: none"> <li>• Modularité</li> <li>• Cohérence (mode de passage, conformité entre nombre, ordre et type des paramètres)</li> </ul>	<b>2 points :</b> <ul style="list-style-type: none"> <li>• 1 point</li> <li>• 1 point</li> </ul>
<b>Saisie de n:</b> <ul style="list-style-type: none"> <li>• Lecture</li> <li>• Contrôle</li> </ul>	<b>1 point :</b> <ul style="list-style-type: none"> <li>• 0.5 point</li> <li>• 0,5 point</li> </ul>
<b>Remplissage de T:</b> <ul style="list-style-type: none"> <li>• Parcours</li> <li>• Lecture</li> </ul>	<b>1.5 points :</b> <ul style="list-style-type: none"> <li>• 1 point</li> <li>• 0.5 point</li> </ul>
<b>Tri de T:</b> <ul style="list-style-type: none"> <li>• Test de répétition des parcours               <ul style="list-style-type: none"> <li>○ initialisation</li> <li>○ incrémentation</li> <li>○ décrémentation</li> <li>○ parcours et condition d'arrêt</li> </ul> </li> <li>• Parcours de gauche à droite :               <ul style="list-style-type: none"> <li>○ Initialisation + boucle</li> <li>○ Test et permutation</li> </ul> </li> <li>• Parcours de droite à gauche               <ul style="list-style-type: none"> <li>○ test</li> <li>○ Initialisation + boucle</li> <li>○ Test et permutation</li> </ul> </li> </ul>	<b>5 points :</b> <ul style="list-style-type: none"> <li>• 0.25 point</li> <li>• 0.25 point</li> <li>• 0.25 point</li> <li>• 0.25 point</li> <li>• 0,75 point (0.25+0.5)</li> <li>• 1 point (0.5+0.5)</li> <li>• 0,5 point</li> <li>• 0,75 point (0.25+0.5)</li> <li>• 1 point (0.5+0.5)</li> </ul>
<b>Affichage de T après tri :</b> <ul style="list-style-type: none"> <li>• Parcours</li> <li>• Ecriture</li> </ul>	<b>1 point</b> <ul style="list-style-type: none"> <li>• 0,5 point</li> <li>• 0.5 point</li> </ul>
<b>TDNT</b>	<b>0, 5 point</b>
<b>TDOG</b>	<b>0, 5 point</b>
<b>TDOL</b>	<b>0, 5 point</b>