

Algorithmique et programmation

Session principale (2018)

Correction

N.B. :

- On n'acceptera pas toute solution sous forme d'analyse ou de traduction Pascal si on demande un algorithme.
- - 0.25 par erreur
- - 0.25 de la note attribuée à tous les TDO si la colonne Rôle est omise ou erronée.

Exercice 1 (3 points = 0.25 * 12)

1. L'opération de décalage est utilisée dans :

le tri rapide

le tri insertion

le tri Shell

2. Le tri insertion est un cas particulier :

du tri sélection

du tri à bulle

du tri Shell

3. Le pas du tri Shell noté P est déterminé en utilisant la suite :

$\begin{cases} P_0=0 \\ P_n=3+P_{n-1} \end{cases}$

$\begin{cases} P_0=1 \\ P_n=2*P_{n-1} \end{cases}$

$\begin{cases} P_0=1 \\ P_n=3*P_{n-1}+1 \end{cases}$

4. La fonction **Verif** permet de vérifier si les N entiers d'un tableau T sont triés en ordre croissant :

0) Def FN Verif (T:Tab ; N:entier) : Booléen
 1) Si (N=1) Alors Verif ← Vrai
 Sinon
 Si (T[N]<T[N-1]) Alors
 Verif ← Faux
 Sinon
 Verif ← Fn Verif (T,N-1)
 Fin Si
 2) Fin Verif

0) Def FN Verif (T:Tab ; N:entier) : Booléen
 1) Si (N=1) Alors
 Verif ← Vrai
 Sinon
 Verif ← (T[N] ≥ T[N-1])
 ET Fn Verif(T,N-1)
 Fin Si
 2) Fin Verif

0) Def FN Verif (T:Tab ; N:entier) : Booléen
 1) Si (N=1) Alors Verif ← Faux
 Sinon
 Si (T[N]<T[N-1]) Alors
 Verif ← Vrai
 Sinon
 Verif ← Fn Verif(T,N-1)
 Fin Si
 2) Fin Verif

Ex n°2 (3 points)

a- L'algorithme de la fonction Fibo :

0) Def Fn Fibo(K:Entier) : Entier

1) Si K < 2 Alors Fibo ← 1
 Sinon Si k mod 2 = 0 Alors

$$\text{Fibo} \leftarrow \text{Carré}(\text{Fibo}(\text{K div } 2 - 1)) + \text{Carré}(\text{Fibo}(\text{K div } 2))$$

$$\text{Sinon Fibo} \leftarrow (2 * \text{Fibo}(\text{K div } 2 + 1) - \text{Fibo}(\text{K div } 2)) * \text{Fibo}(\text{K div } 2)$$

Fin Si

2) **Fin Fibo**

b- L'algorithmme du module Fibo Som :

0) **Def Fn Fibo_Som(n:Entier) : Entier**

1) $\text{Fibo_Som} \leftarrow \text{Fn Fibo}(n+2) - 1$

2) **Fin Fibo_Som**

Le tableau de déclarations des objets locaux

Objet	Type/Nature	Rôle
Fibo	Fonction	Calculer un terme de la suite de Fibonacci

Exercice 3 (4 points)

1. $\text{Inconnue}(5,2) = \text{Vrai}$

$\text{Inconnue}(6,2) = \text{Faux}$

$\text{Inconnue}(7,2) = \text{Vrai}$

$\text{Inconnue}(9,2) = \text{Faux}$

2. La fonction Inconnue **retourne la valeur Vrai si l'entier E (avec $E > 2$) est premier, ou la valeur Faux dans le cas contraire.**

3. L'algorithmme de la fonction Calcul :

0) **Def Fn Calcul(Epsilon:Réel) : Réel**

1) $R \leftarrow 4/3, i \leftarrow 1$

Répéter

$i \leftarrow i+2$

Si Fn Inconnue(i,2) Alors $A_r \leftarrow R$

$R \leftarrow R * \text{Carré}(i) / (\text{Carré}(i) - 1)$

Fin Si

Jusqu'à $(\text{Abs}(\text{RacineCarré}(6 * R) - \text{RacineCarré}(6 * A_r)) < \text{Epsilon})$

2) $\text{Calcul} \leftarrow \text{RacineCarré}(6 * R)$

3) **Fin Calcul**

Le tableau de déclarations des objets locaux

Objet	Type/Nature	Rôle
i	Entier	Compteur
R	Réel	Calculer $\pi^2/6$
Ar	Réel	Sauvegarder l'ancienne valeur de $\pi^2/6$
Inconnue	Fonction	Vérifier si i est premier

Problème (10 points)

1) Le tableau de déclarations des nouveaux types :

Type
Matrice = Tableau de 5 x 5 Octets
Tab = Tableau de 10 Octets

2) L'algorithme de la procédure **MiseAjour** :

```
0. Def Proc MiseAjour(Var R:Tab; N: Octet)
1. Pour i de 2*N à 2 (pas = -1) Faire
    Si R[i] > 9 Alors R[i-1]←R[i-1] + R[i] Div 10
        R[i] ← R[i] mod 10
    Fin Si
Fin Pour
2. Fin MiseAjour
```

Le tableau de déclarations des objets locaux

Objet	Type/Nature	Rôle
I	Entier	Compteur

3) Analyse du programme principal :

```
⑦ Résultat = Ecrire( A, " * ", B, " = " , P)
① A = Proc Saisie(A)
② B = Proc Saisie(B)
⑥ P←Fn Result(R,2*Long(CA))
⑤ R= Proc Diag(M, Long(CA) , R)
④ M = Remplir(M,CA,CB)
③ (CA,CB) = [Convch(A, CA), Convch(B, CB) ]
    Tantque (Long(CA) ≠ long(CB)) Faire
        Si Long(CA)< long(CB) Alors CA ← "0" + CA
        Sinon CB ← "0" + CB
    Fin Si
Fin Tantque
```

Le tableau de déclarations des objets globaux

Objet	Type/Nature	Rôle
A	Entier	Contenir l'entier A
B	Entier	Contenir l'entier B
CA	Chaine[5]	Une chaîne contenant les chiffres de l'entier A
CB	Chaine[5]	Une chaîne contenant les chiffres de l'entier B
P	Chaine[10]	Contenir l'équivalent du produit de A par B

M	Matrice	Contenir les résultats des produits des chiffres de A et de B
R	Tab	Contenir les sommes des chiffres des diagonales
Saisie	Procédure	Saisir un entier
Result	Fonction	Concaténer les chiffres du tableau R
Diag	Procédure	Remplir et mettre à jour le tableau R à partir des diagonales
Remplir	Procédure	Remplir M par les résultats des produits des chiffres de A et de B

4) Les algorithmes des modules envisagés :

a) L'algorithme de la procédure Saisie :

0) Def Proc Saisie(Var K:Entier)

1) Répéter

Ecrire("Saisir un entier ")

Lire(K)

Jusqu'à (K ≥ 10) et (K ≤ 10000)

2) Fin Saisie

b) L'algorithme de la procédure Remplir :

0) Def Proc Remplir(Var M:Matrice ; Cha,Chb:Chaîne)

1) Pour i de 1 à Long(Cha) Faire

Pour j de 1 à Long(Chb) Faire

$M[i,j] \leftarrow (\text{Ord}(\text{Cha}[i]) - 48) * (\text{Ord}(\text{Chb}[j]) - 48)$

Fin Pour

Fin Pour

2) Fin Remplir

Le tableau de déclarations des objets locaux

Objet	Type/Nature	Rôle
i	Octet	Compteur
j	Octet	Compteur

c) L'algorithme de la fonction Result

0) Def Fn Result (R:Tab ; K:Octet) : Chaîne

1) Res ← ""

Pour i de K à 1 Faire

Res ← Chr(R[i] + 48) + Res

Fin Pour

2) Result ← Res

3) Fin Résultat

Le tableau de déclarations des objets locaux

Objet	Type/Nature	Rôle
i	Octet	Compteur
Res	Chaîne	Former le résultat du produit de A et B

Barème :

Exercice 1 : (3 points = 12 * 0.25)

On accepte les réponses V, F, Vrai, Faux

1. F-V-V

2. F-F-V

3. F-F-V

4. V-V-F

Exercice n°2 : (3 points)

a) Fonction récursive Fibo (2.25 points)

Tâches	Points
Entête (Paramètres + type de la fonction)	0.5 (=0.25+0.25)
Condition + traitement d'arrêt	0.25+0.25
Test si n est pair	0.25
Appel récursif cas où n est pair	0.5
Appel récursif cas où n est impair	0.5

b) Fonction Fibo_som (0.75 points)

Tâches	Points
Entête (Paramètres + type de la fonction)	0.25
Appel de la fonction fibo et affectation	0.25
TDO	0.25

Exercice n°3 : (4 points)

- Inconnue(5,2)=**Vrai** Inconnue(6,2)=**Faux** Inconnue(7,2)=**Vrai** Inconnue(9,2)=**Faux**
(1 point = 4 *0.25)
- Vérifier** si un entier E est **premier** : (0.5 point= 0.25+0.25)
- Calcul approché de π : Fonction calcul (epsilon) : (2.5 points)

Tâches	Points
Entête	0.25
Initialisations	0.25
Boucle + condition d'arrêt	0.5=0.25+0.25
Test premier avec la fonction Inconnue	0.25
Calcul du nouveau terme + affectation	0.5=0.25+0.25
Incrémenter le compteur	0.25
Affectation du résultat au nom de la fonction	0.25
TDO	0.25

Problème : (10 points)

- TDNT Matrice et Tab: (0.5 point = 0.25+0.25)
- Algorithme PROC MiseAJour (R,N) : (1.50 point)

Tâches	Points
Entête	0.25
Boucle	0.25
Test	0.25

Affectations de mise à jour	0.5
TDO	0.25

3. Analyse du Problème : (2 points)

Tâches	Points
➤ Modularité	0.5
➤ Cohérence (appels + conformité des paramètres)	1 = 0.5+0.5
➤ TDOG	0.5

4. Algorithmes des modules envisagés : (6 points)

Tâches	Points
➤ TDOL	1
➤ Saisie de A et de B : - Lecture - Contrainte	1 = 0.5 = 0.25 * 2 0.5 = 0.25 * 2
➤ Conversion de A et B en chaînes et ajustement des longueurs - Conversion de A et B en chaînes - Ajustement des longueurs (boucle + test + affectations)	1.25 = 0.5 = 0.25 * 2 1 = 0.25 +0.25+0.25
➤ Remplissage de la matrice (Boucles + affectation)	1 = 0.5 * 2
➤ Appel du module Diag et détermination du produit à partir de R - Appel du module Diag - Détermination du produit ▪ Initialisation ▪ Boucle ▪ Affectation	1.25 = 0.25 0.25 0.5 0.25
➤ Affichage du produit	0.5