

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

Signature des surveillants
.....
.....



Épreuve : **INFORMATIQUE**
Sections: **Mathématiques, Sciences expérimentales et Sciences techniques**
Session 2019

*Le sujet comporte 4 pages numérotées de 1/4 à 4/4.
Les pages 1/4 et 2/4 sont à remplir par le candidat et à rendre avec sa copie.*

Exercice 1 (4 points)

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

1) Soit l'algorithme suivant :

```
0) Début Inconnu
1) Lire (C1)
2) Lire (C2)
3) C3 ← 0
   Pour i de 1 à Long(C2) faire
     Si (Majus(C2[i]) = Majus(C1))
       Alors C3 ← C3 + 1
     Fin si
   Fin pour
4) Ecrire (C3)
5) Fin Inconnu
```

a) Ci-dessous des extraits de propositions de tableaux de déclaration des objets utilisés. La déclaration correspondante à l'algorithme **Inconnu** est :

| Objet | Type/nature |
|-------|-------------|
| C1 | Chaîne |
| C2 | Chaîne |
| C3 | Entier |

| Objet | Type/nature |
|-------|-------------|
| C1 | Caractère |
| C2 | Caractère |
| C3 | Entier |

| Objet | Type/nature |
|-------|-------------|
| C1 | Chaîne |
| C2 | Caractère |
| C3 | Réel |

| Objet | Type/nature |
|-------|-------------|
| C1 | Caractère |
| C2 | Chaîne |
| C3 | Entier |

b) Afin d'améliorer le message d'affichage du résultat de l'algorithme précédent et de le rendre significatif relativement au traitement effectué, l'instruction numéro 4 sera remplacée par l'instruction suivante :

- Ecrire ("Le nombre de caractères majuscules de ",C1," et ",C2," est : ",C3)
- Ecrire ("Le nombre d'occurrences de ",C1," dans ",C2," est : ",C3)
- Ecrire ("Le nombre de chiffres dans ",C2," est : ",C3)
- Ecrire ("Le nombre de caractères communs entre ",C1," et ",C2," est : ",C3)

2) Soit la suite U définie par
$$\begin{cases} U_0 = 1 \\ U_n = 1 + 1/U_{n-1} \text{ pour tout } n > 0 \end{cases}$$

a) La séquence algorithmique qui permet de déterminer le terme U_n avec $n \geq 0$ est :

| | | | |
|---|---|---|---|
| $T[1] \leftarrow 1$ Pour i de 2 à $n+1$ faire $T[i] \leftarrow 1 + 1/T[i-1]$ Fin pour $U_n \leftarrow T[n+1]$ <input type="checkbox"/> | $U_0 \leftarrow 1$ Pour i de 1 à n faire $U_n \leftarrow 1 + 1/U_0$ Fin pour <input type="checkbox"/> | $U_n \leftarrow 1$ Pour i de 1 à n faire $U_n \leftarrow 1 + 1/U_n$ Fin pour <input type="checkbox"/> | $U_0 \leftarrow 1$ Pour i de 1 à n faire $U_n \leftarrow 1 + 1/U_0$ $U_0 \leftarrow U_n$ Fin pour $U_n \leftarrow U_0$ <input type="checkbox"/> |
|---|---|---|---|

b) L'entête de la fonction qui permet de déterminer le terme U_n avec $n \geq 0$ est :

| | |
|--|--|
| DEF FN Suite (n : Réel) : Réel <input type="checkbox"/> | DEF FN Suite (n : Entier) : Entier <input type="checkbox"/> |
| DEF FN Suite (n : Entier) : Réel <input type="checkbox"/> | DEF FN Suite (n : Réel) : Entier <input type="checkbox"/> |

Exercice 2 (3 points)

Soit la séquence algorithmique ci-dessous, où x est un entier naturel :

```


nb ← 1
Tant que (x Div 10 ≠ 0) faire
    nb ← nb + 1
    x ← x Div 10
Fin tant que
  
```

1) Compléter le tableau ci-contre par la valeur finale de nb suite à l'exécution de cette séquence pour chacune des valeurs de x .

| x | nb |
|------|------|
| 5403 | |
| 176 | |
| 3 | |

2) Donner le rôle de cette séquence.

3) Ecrire une séquence algorithmique équivalente à celle donnée précédemment sans utiliser une structure itérative.

| | | |
|--|---|---|
| RÉPUBLIQUE TUNISIENNE MINISTÈRE DE L'ÉDUCATION EXAMEN DU BACCALAURÉAT | Session 2019 | |
| | Épreuve : INFORMATIQUE | Sections : Mathématiques, Sciences expérimentales et Sciences techniques |
| |  Durée : 1h 30 | Coefficient de l'épreuve : 0.5 |

QR QR QR QR QR QR

*Le sujet comporte 4 pages numérotées de 1/4 à 4/4.
Les pages 1/4 et 2/4 sont à remplir par le candidat et à rendre avec sa copie.*

Problème (13 points)

Une société commerciale cherche à automatiser l'accès à ses services via des cartes à puces. Chaque carte est caractérisée par un identifiant unique, une date de création et un code d'accès. Pour cela, on propose d'écrire un programme permettant de :

1. Stocker les identifiants de N cartes dans un tableau **IDENT** (avec $3 \leq N \leq 50$), sachant que l'identifiant est une chaîne formée de **8** chiffres,
2. Stocker les dates de création de ces N cartes d'accès dans un tableau **DATE**, sachant qu'une date de création est une chaîne de caractères formée de deux parties : le jour **J** et le mois **M** séparés par le caractère "/" (On ne tiendra compte que des contrôles suivants : **J** est un entier compris entre 1 et 31 et **M** est un entier compris entre 1 et 12).
3. Déterminer dans un tableau **CODE** les N codes à partir du tableau **DATE** en utilisant le principe ci-après, sachant qu'un code est une chaîne de **4** chiffres :
 - Concaténer **J** et **M** pour former un nombre de **2, 3** ou **4** chiffres.
 - Multiplier ce nombre par un entier aléatoire **X** compris entre **5** et **64** afin d'obtenir un nouveau nombre **Y**.
 - Former le code de quatre chiffres comme suit :
 - Si **Y** est un entier de **4** chiffres, déplacer le chiffre des milliers vers la droite de ce nombre.
 - Si **Y** est inférieur à 1000, ajouter des zéros (**0**) à droite de ce nombre pour que sa longueur soit égale à **4**.
 - Si **Y** est supérieur à 9999, additionner le nombre formé des 4 chiffres de gauche au nombre formé des chiffres restants à droite jusqu'à ce que sa longueur soit égale à **4**.
4. Afficher les informations relatives à chacune des cartes sous la forme suivante :

Identifiant de la carte : Date de sa création correspondante : Code correspondant

NB : Le candidat n'est pas appelé à vérifier l'unicité des identifiants dans le tableau **IDENT**.

Exemple :

Pour $N=3$ et les deux tableaux **IDENT** et **DATE** suivants :

| | | | |
|--------------|----------|----------|----------|
| IDENT | 12345678 | 23456789 | 34567891 |
| | 1 | 2 | 3 |
| DATE | 8/11 | 2/1 | 24/12 |
| | 1 | 2 | 3 |

❖ On obtient le tableau **CODE** suivant :

| | | | |
|-------------|------|------|------|
| CODE | 1108 | 6720 | 1314 |
| | 1 | 2 | 3 |

En effet, les codes des cartes sont obtenus comme suit :

- Pour la carte N°1, la concaténation de **J** et de **M** donne le nombre **811**. En supposant que l'entier aléatoire $X = 10$, le nombre obtenu est égal à **8110** ($811 * 10$) qui est composé de 4 chiffres. En déplaçant le chiffre des milliers (**8**) vers la droite, on obtient le code **1108**.
- Pour la carte N°2, la concaténation de **J** et de **M** donne le nombre **21**. En supposant que l'entier aléatoire $X = 32$, le nombre obtenu est égal à **672** ($21 * 32$) qui est inférieur à 1000. En ajoutant un zéro à droite pour que sa longueur soit égale à 4, on obtient le code **6720**.
- Pour la carte N°3, la concaténation de **J** et de **M** donne le nombre **2412**. En supposant que l'entier aléatoire $X = 53$, le nombre obtenu est égal à **127836** ($2412 * 53$) qui est supérieur à 9999. Le nombre formé des 4 chiffres de gauche (**1278**) sera additionné au nombre formé des chiffres restants à droite (**36**), on obtient le code **1314** ($1278 + 36$).

❖ Le programme affiche :

12345678 : 8/11 : 1108
23456789 : 2/1 : 6720
34567891 : 24/12 : 1314

Travail demandé :

1. Analyser le problème en le décomposant en modules.
2. Ecrire les algorithmes et les tableaux de déclaration des objets relatifs aux modules envisagés.

| | | |
|--|----------------------------------|---|
| <p>RÉPUBLIQUE TUNISIENNE MINISTÈRE DE L'ÉDUCATION</p> <p>EXAMEN DU BACCALAURÉAT</p> | Session 2019 | |
| | Épreuve : INFORMATIQUE | Sections : Mathématiques, Sciences expérimentales et Sciences techniques |
| | Durée : 1h 30 | Coefficient de l'épreuve : 0.5 |

Recommandations, explications, corrigé et barème de notation

Exercice 1 (4 points = 0.25 × 16)

(Recommandations et explications)

1)

a)

- Il est fortement recommandé de lire rapidement l'algorithme proposé et de voir ce qui est demandé de faire dans les questions.
- Il s'agit de répondre si des propositions de types de variables sont correctes ou fausses. il faut à ce moment après avoir lu les questions entrer dans les détails de l'algorithme proposé et remarquer les différentes manipulations sur les variables :

Remarquer que :

La variable C2 est parcourue via des indices dans Majus(C2[i]), ne peut pas être de type tableau sinon on ne peut pas faire lire(C2) et donc le type possible pour cette variable est chaîne de caractère.

- Ce qui élimine la proposition 2

Majus(C1) témoigne que la fonction Majus est appliquée sur une variable de type caractère (C1) → Ce qui élimine les propositions 1 et 3

La variable C3 est initialisée à 0 elle ne peut être qu'entière ou réelle.

La proposition 3 est correcte d'après les remarques citées supra.

b)

Remarquer d'après les propositions qu'on s'intéresse à la variable C3, la valeur de cette variable est déterminée par l'expression $C3 \leftarrow C3 + 1$. Cette expression est exécuté dans la boucle chaque fois qu'on trouve le caractère donné par la variable C1 dans une position dans la variable C2 (Majus(C2[i]) = Majus(C1)) → ce qui permet de compter alors le nombre de fois ou le caractère défini par C1 apparait dans C2.

On n'a pas le problème de majuscule/minuscule puisque tout est mis en majuscule. → La deuxième proposition est vraie

Les autres propositions ne sont pas correctes :

Le nombre de chiffres... : aucune indication sur les chiffres

Caractères en communs... : C1 présente un seul caractère

Le nombre de caractères majuscules de... : Ne peut être vraie puisque Majus() transforme en majuscule déjà après quoi on effectue le compte.

2)

a)

Remarquer que la suite est définie par son premier terme U_0 et le Terme général U_n . Ce dernier est défini par le terme qui le précède U_{n-1} .

1)

$T[1] \leftarrow 1$

Pour i de 2 à $n+1$ faire

$T[i] \leftarrow 1+1/T[i-1]$

Fin pour

$U_n \leftarrow T[n+1]$

→ remarquer qu'on a mis le premier élément U_0 de la suite dans la première case du tableau $T[1]$, l'élément U_1 sera mis dans la case $T[2]$...l'élément U_n sera mis dans la case $T[n+1]$

Donc cette suite d'instruction donne bien le résultat et calcule le terme U_n

2)

$U_0 \leftarrow 1$

Pour i de 1 à n Faire

$U_n \leftarrow 1+1/U_0$

Fin pour

→ Cette suite d'instruction ne peut pas donner le résultat puisque le terme (U_n) sera toujours égal à 2 du début jusqu'à la fin..

3)

$U_n \leftarrow 1$

Pour i de 1 à n Faire

$U_n \leftarrow 1+1/U_n$

Fin pour

→ Cette suite d'instruction donne bien le résultat puisque le terme (U_n) change à chaque fois et le nouveau terme est bien donné en fonction du terme précédent. Exemple : pour $i=1$; la première exécution de la boucle donne $U_n = 1+1/1 = 2$; la deuxième exécution donne $U_n=1+1/2 \dots$ etc. Ce qui donne bien le résultat.

4)

$U_0 \leftarrow 1$

Pour i de 1 à n Faire

$U_n \leftarrow 1+1/U_0$

$U_0 \leftarrow U_n$

Fin pour

$U_n \leftarrow U_0$

→ Cette suite d'instruction donne bien le résultat puisque à chaque exécution de la boucle le terme U_0 prend la valeur du terme précédent et (U_n) est calculé en fonction du terme précédent.

b)

La seule entête qui donne le résultat est bien DEF FN Suite (n : Entier) : Réel

Puisque n est entier et le résultat de calcul du nième terme de la suite est bien réel.

CORRIGÉ

Exercice 1 (4 points)

1)

a)

| Objet | Type/nature |
|-------|-------------|
| C1 | Chaîne |
| C2 | Chaîne |
| C3 | Entier |

F

| Objet | Type/nature |
|-------|-------------|
| C1 | Caractère |
| C2 | Caractère |
| C3 | Entier |

F

| Objet | Type/nature |
|-------|-------------|
| C1 | Chaîne |
| C2 | Caractère |
| C3 | Réel |

F

| Objet | Type/nature |
|-------|-------------|
| C1 | Caractère |
| C2 | Chaîne |
| C3 | Entier |

V

b)

F Ecrire ("Le nombre de caractères majuscules de ",C1," et ",C2," est : ",C3)

V Ecrire ("Le nombre d'occurrences de ",C1," dans ",C2," est : ",C3)

F Ecrire ("Le nombre de chiffres dans ",C2," est : ",C3)

F Ecrire ("Le nombre de caractères communs entre ",C1," et ",C2," est : ",C3)

2)

a) La séquence algorithmique qui permet de déterminer le terme U_n avec $n \geq 0$ est :

```
T[1] ← 1
Pour i de 2 à n+1 faire
  T[i] ← 1+1/T[i-1]
Fin pour
Un ← T[n+1]
```

V

```
U0 ← 1
Pour i de 1 à n Faire
  Un ← 1+1/U0
Fin pour
```

F

```
Un ← 1
Pour i de 1 à n Faire
  Un ← 1+1/Un
Fin pour
```

V

```
U0 ← 1
Pour i de 1 à n Faire
  Un ← 1+1/U0
  U0 ← Un
Fin pour
Un ← U0
```

V

b) L'entête de la fonction qui permet de déterminer le terme U_n avec $n \geq 0$ est :

DEF FN Suite (n : Réel) : Réel **F**

DEF FN Suite (n : Entier) : Entier **F**

DEF FN Suite (n : Entier) : Réel **V**

DEF FN Suite (n : Réel) : Entier **F**

Exercice 2 (3 points = 0.5 × 3 + 0.5 + 0,5 × 2)

(Recommandations et explications)

1)

| x | nb |
|------|----|
| 5403 | 4 |
| 176 | 3 |
| 3 | 1 |

2)

Chaque fois que le quotient de la division de x par 10 est différent de zéro on ajoute 1 à nb et on aura un nouveau quotient $x \leftarrow x \text{ Div } 10$ cela se termine lorsque ce dernier (le quotient) sera égal à zéro

Exemple $8 \text{ div } 10 = 0$

Donc cela calcule bien le nombre de chiffres(ou digit) qui composent un nombre.

3)

On convertit le nombre en une chaîne et puis on compte le nombre de caractères qui sera la longueur de la chaîne..

CORRIGÉ

1)

| x | nb |
|------|----|
| 5403 | 4 |
| 176 | 3 |
| 3 | 1 |

2) Cette séquence permet de déterminer **le nombre de chiffres** d'un **entier x** donné.

3) **Convch(x, xch)**

nb \leftarrow **Long(xch)**

NB. -0,25 par erreur

Problème :

(Recommandations et explications)

Lire attentivement l'énoncé du problème et dégager le résultat final demandé →

Il s'agit d'afficher les informations relatives à chacune des cartes comme c'est demandé dans l'étape 4

On pourra créer un module pour l'affichage des : Identifiant de la carte, de la Date de sa création correspondante et le Code correspondant. Puisque d'après l'énoncé toutes ces informations sont à extraire à partir des tableaux. Appelons ce module **Afficher**

Remarque : Il faut voir le problème dans son globalité et remarquer qu'il s'agit bien en fin de compte de trois tableaux : Un pour stocker les identifiants, un pour stocker les dates et un pour stocker les codes.

Le parcours de ces tableaux en même temps donne le résultat voulu : Pour un indice i qui va de 1 à n on parcourt simultanément les trois tableaux.

Si on cherche dans l'énoncé on trouve bien les façons pour remplir ces 3 tableaux

Le tableau des identifiants c'est à remplir comme indiqué dans l'étape 1 dans l'énoncé

Le tableau des dates c'est à remplir comme indiqué dans l'étape 2 de l'énoncé

Le tableau des codes c'est à remplir comme indiqué dans l'étape 3 de l'énoncé

Pour les deux premiers tableaux on pourra se servir d'un seul module pour le remplissage. Appelons le **Remplir**

Pour le troisième tableau on pourra se servir d'un module pour le remplir. Appelons le **Coder**

Remarques :

- Des contrôles de saisie sont à prévoir dans le module Remplir pour le remplissage des tableaux des identifiants et le tableau des dates.

(On pourra aussi créer des modules de remplissage pour chaque tableau mais cette solution est à éviter pour gagner du temps, un seul module est suffisant)

- Il est à dire que les contrôles de saisie empêchent les utilisateurs des programmes de saisir des données en dehors de ce qui est permis ou des valeurs permises comme par exemple saisir le mois numéro 15 qui n'existe pas bien sûr.

- Remarquer aussi que dans le cadre des contrôles de saisie que dans la première étape de l'énoncé il y a la phrase : **sachant que l'identifiant est une chaîne formée de 8 chiffres** donc on doit vérifier après chaque saisie si la chaîne saisie par l'utilisateur est formée seulement par des chiffres. On pourra prévoir un module pour la vérification si le numéro saisi contient seulement des chiffres, appelons ce module Verifnum, ce dernier va être utilisé dans le module Remplir.

Maintenant qu'on a compris de quoi il s'agit il est simple de passer à l'élaboration des différents modules.

Commençons par le programme principal puis élaborons les différents modules :

CORRIGÉ

1) Analyse du programme principal :

Nom : Validation

Résultat = PROC Afficher (IDENT, DATE, CODE, n)
 (IDENT, DATE, CODE, n) = PROC Remplir (IDENT, DATE, n)
 PROC Coder (CODE, DATE, n)

Fin Validation

TDNT

| Type |
|----------------------------|
| tab= tableau de 50 chaînes |

TDOG

| Objet | Type/Nature | Rôle |
|----------|-------------|--|
| IDENT | tab | Contient les identifiants des cartes |
| DATE | tab | Contient les dates de création des cartes |
| CODE | tab | Contient les codes des cartes |
| n | Entier | Le nombre de cartes |
| Remplir | Procédure | Saisir n et remplir les tableaux Ident et Date |
| Coder | Procédure | Déterminer les codes des différentes cartes |
| Afficher | Procédure | Afficher les éléments des trois tableaux |

2) Les algorithmes des modules :

0) DEF PROC Remplir (var Id,D:tab ; var n:entier)

1) Répéter

Ecrire ("Donner le nombre de cartes : ")

Lire(n)

Jusqu'à (n dans [3..50])

2) Pour i de 1 à n faire

Répéter

Ecrire ("Donner l'identifiant de la carte n° ",i, " : ")

Lire(Id[i])

Jusqu'à (Long(Id[i]=8)) et (FN Verifnum(Id[i]))

Ecrire ("Donner la date en jour et en mois de la carte n° ",i, " : ")

Répéter

Ecrire ("Donner J : ")

Lire(j)

Jusqu'à (j dans [1..31])

Répéter

Ecrire ("Donner M : ")

Lire(m)

Jusqu'à (m dans [1..12])

Convch(j,jch)

Convch(m,mch)

D[i] ← jch+"/"+mch

Fin pour

3) Fin Remplir

TDOL

| Objet | Type/Nature | Rôle |
|----------|-------------|---|
| i | Entier | Compteur |
| m | Entier | Contient le mois |
| j | Entier | Contient le jour |
| jch | chaîne | Résultat de la conversion du jour en chaîne |
| mch | chaîne | Résultat de la conversion du mois en chaîne |
| Verifnum | Fonction | Vérifier si une chaîne est numérique |

DEF FN Verifnum (ch:chaîne) : booléen

- 0) $k \leftarrow 0$
 Répéter
 $k \leftarrow k+1$
 Jusqu'à (non ch[k] dans ["0". ."9"]) ou (k=Long(ch))
 1) Verifnum \leftarrow ch[k] dans ["0". ."9"]
 2) **Fin Verifnum**

TDOL

| Objet | Type/Nature | Rôle |
|-------|-------------|----------|
| k | Entier | Compteur |

0) DEF PROC Coder (var C:tab ; D:tab ; n:entier)

- 1) Pour i de 1 à n faire
 ch \leftarrow D[i]
 Efface (ch,Pos("/",ch),1)
 Valeur (ch,r,e)
 $x \leftarrow 5 + \text{Aléa}(60)$
 $nb \leftarrow r * x$
 Convch(nb,nbch)
 Si (Long (nbch)=4)
 Alors C[i] \leftarrow Sous_chaine(nbch,2,4) + nbch[1]
 Sinon
 Si (Long(nbch) < 4)
 Alors
 Répéter
 $nbch \leftarrow nbch + "0"$
 Jusqu'à (Long(nbch) = 4)
 C[i] \leftarrow nbch
 Sinon
 Valeur(Sous_chaine(nbch,1,4),nb1)
 Valeur(Sous_chaine(nbch,5,Long(nbch)),nb2)
 Convch(nb1+nb2, C[i])
 Fin si
 Fin si
 Fin pour
 2) **Fin Coder**

TDOL

| Objet | Type/Nature | Rôle |
|-------|-------------|---|
| i | Entier | Compteur |
| r | Entier | Contient les chiffres relatifs à la date |
| x | Entier | Entier aléatoire |
| e | Entier | Position de l'erreur |
| nb | Entier long | Résultat de la multiplication de r par x |
| nb1 | Entier | Les quatre chiffres de gauche de nb |
| nb2 | Entier | Le reste des chiffres de nb |
| ch | Chaîne | Une date du tableau D |
| nbch | Chaîne | Résultat de la conversion de nb en chaîne |

0) DEF PROC Afficher (Id,D,C:tab ; n:entier)

- 1) Pour i de 1 à n faire
 Ecrire (Id[i], " : ", D[i], " : ", C[i])
 Fin pour
 2) **Fin Afficher**

TDOL

| Objet | Type/Nature | Rôle |
|-------|-------------|----------|
| i | Entier | Compteur |

Barème détaillé :

| | |
|---|---|
| Programme principal : - Modularité - Cohérence (appels + conformité des paramètres) | 1.5 points = 0.5 1 = (0.5+0.5) |
| Saisie du nombre de cartes N : - Lecture - Contrôle de la contrainte | 0.5 point = 0.25 0.25 |
| Remplissage du tableau IDENT : - Parcours du tableau - Lecture de IDENT[i] - Contrôle des contraintes : 8 chiffres (chiffres + longueur) | 1.75 points = 0.25 0.25 1.25 = (1+0.25) |
| Remplissage du tableau DATE : - Parcours du tableau - Lecture de J + contrôle des contraintes - Lecture de M + contrôle des contraintes - Remplissage d'un élément date (conversion + concaténation + affectation) | 2 points = 0.25 0.25+0.25 0.25+0.25 0.75 = (0.25*3) |
| Formation du tableau CODE : - Parcours du tableau - Concaténation de J et M - Lecture aléatoire de X - Détermination du nombre Y - Formation du code (cas 1 + cas 2 + cas 3) - Remplissage d'un élément Code | 4.25 points = 0.25 0.5 0.5 0.25 2.5 = (0.5+0.75+1.25) 0.25 |
| Affichage des éléments des tableaux : - Parcours des tableaux - Affichage des éléments + respect de la forme | 1.25 points = 0.25 1 = (0.25*3 + 0.25) |
| TDNT + TDOG | 0.75 point = (0.25+0.5) |
| TDOL | 1 point |

-0,25 par erreur.