

Concours d'accès en deuxième année du Cycle  
d'Ingénieurs d'Etat  
INPT – Rabat – 10 Juillet 2003

NOM : .....	PRENOM : .....
N° CIN : .....	N° PLACE : .....

QCM d'Informatique  
DUREE : 2 Heures

Ce fascicule comporte 15 questions

DIRECTIVES

- Aucun feuillet ne doit être séparé du fascicule de questions qui sera rendu impérativement avec les réponses pour valider le QCM
- Un seul fascicule est distribué par candidat
- L'usage de la calculatrice est interdit
- Pour chaque question, il y a 4 propositions numérotées 1, 2, 3 et 4
- Il y a toujours une réponse et une seule parmi les 4 proposées
- Répondez au QCM en utilisant le « Document de réponse » qui se trouve à la fin du fascicule
- Pour chaque question de la grille de réponses du « Document de réponse », remplissez la case correspondante par le numéro de la proposition que vous jugez bonne

BAREMES ET EVALUATION

- Réponse exacte : 1 point
- Réponse inexacte ou réponse multiple : 0 point
- Absence de réponse : 0 point

## INFORMATIQUE, SERIE 4

### Notes

- $v : T \Rightarrow v$  est une variable de type  $T$
- $u, v, x : T \Rightarrow u, v$  et  $x$  sont de type  $T$
- *Integer* est le type des entiers
- *Char* est le type des caractères
- *Boolean* est le type des expressions booléennes dont les valeurs possibles sont « True » ou « False »
- *Real* est le type des nombres réels
- $v := \text{val} \Rightarrow v$  reçoit la valeur « val »
- $v = \text{val} \Rightarrow v$  est égal à « val »
- $\text{Afficher}(\text{Expr}) \Rightarrow$  affichage à l'écran de l'expression « Expr »
- $\text{Lire}(v) \Rightarrow$  Lecture d'une valeur au clavier et son affectation à la variable  $v$
- $\text{Tab} : \text{Tableau}[1..N]$  of  $T \Rightarrow$  déclaration d'un tableau de  $N$  éléments de type  $T$ .  
 $\text{Tab}[i]$  désigne l'élément du tableau  $\text{Tab}$  qui se trouve à la position  $i$ .

**Q 1 : Soit la boucle while (tantque) suivante :**

```
I : Integer ; J : Integer ;  
I := 0 ; J := 10
```

```
while (I < 10) and (J > 0)  
début  
  Afficher ("Bonjour");  
  I := I + 1;  
  J := J - 2 ;  
fin du while.
```

L'exécution de cette boucle affichera « Bonjour »:

1	5 fois
2	9 fois
3	10 fois
4	6 fois

## Q2 :Recherche d'un élément « elem » dans un tableau T.

1

Algorithme 1 :

```
T:Tableau [1..10] of Integer ; Trouvé:Boolean ; I:Integer;
début
  I :=1 ; Trouvé := False ;
  while (Trouvé=False) or (I ≤ 10)
    début
    Si T[I]=elem alors
      Trouvé := True ;
    Sinon
      Trouvé := False ;
    finSi
    I:=I+1;
  fin du while
  Si Trouvé=True alors
    Afficher('Elément trouvé');
  Sinon
    Afficher('Elément non trouvé');
  finSi
fin
```

2

Algorithme 2 :

```
T:Tableau [1..10] of Integer ; Trouvé:Boolean ; I:Integer;
début
  I :=1 ; Trouvé := False ;
  while I ≤ 10
    début
    Si T[I]=elem alors
      Trouvé := True ;
    Sinon
      I:=I+1;
    finSi
  fin du while
  Si Trouvé=True alors
    Afficher('Elément trouvé');
  Sinon
    Afficher('Elément non trouvé');
  finSi
fin
```

3

Algorithme 3 :

```
T:Tableau [1..10] of Integer ; Trouvé:Boolean ; I:Integer;
début
  I :=1 ; Trouvé := False ;
  while I ≤ 10
    début
    Si T[I]=elem alors
      Trouvé := True ;
    Sinon
      Trouvé := False ;
    finSi
    I := I+1 ;
  fin du while
  Si Trouvé=True alors
    Afficher('Elément trouvé');
  Sinon
    Afficher('Elément non trouvé');
  finSi
fin
```

4

```
Algorithme 4 :
T:Tableau [1..10] of Integer ;
Trouvé:Boolean ;
I:Integer;
début
  I :=1 ; Trouvé := False ;
  while (Trouvé=False) and (I ≤ 10)
    début
      Si T[I]=elem alors
        Trouvé := True ;
      Sinon
        I:=I+1;
      finSi
    fin du while
  Si Trouvé=True alors
    Afficher('Elément trouvé');
  Sinon
    Afficher('Elément non trouvé');
  finSi
fin
```

### Q 3 : UNIX est un système d'exploitation :

- |   |                               |
|---|-------------------------------|
| 1 | multi tâche/mono utilisateur  |
| 2 | multi tâche/multi utilisateur |
| 3 | mono tâche/mono utilisateur   |
| 4 | mono tâche/multi utilisateur  |

### Q4 : Affichage du message « Bonjour » au moins une fois et jusqu'à ce que l'utilisateur décide d'arrêter l'affichage en tapant le caractère 'y'.

1

```
Algorithme 1:
stop:char;
début
stop := 'y';
while stop ≠ 'y'
  début
    Afficher('Bonjour' );
    Afficher ('Voulez vous arrêter l'affichage ?');
    Lire (stop);
  fin du while
fin
```

2

```
Algorithme 2 :
stop:char;
début
stop := 'n';
while stop = 'n'
  début
    Afficher('Bonjour' );
    Afficher ('Voulez vous arrêter l'affichage ?');
    Lire (stop);
  fin du while
fin
```

3

Algorithme 3 :

stop:char;

**début**

stop := 'n';

**while** stop ≠ 'y'

**début**

Afficher('Bonjour' );

Afficher ('Voulez vous arrêter l'affichage ?');

Lire (stop);

**fin du while**

**fin**

4

Algorithme 4 :

stop:char;

**début**

stop := 'n';

**while** stop ≠ 'y'

**début**

Afficher('Bonjour' );

**fin du while**

**fin**

**Q 5 : Soit la boucle while suivante :**

I : Integer ; J : Integer ;

I :=0 ; J:=10

**while** (I < 10) or (J >0)

**début**

Afficher ("Bonjour");

I:=I+1;

J :=J-2 ;

**fin du while.**

L'exécution de cette boucle affichera « Bonjour »:

- |   |         |
|---|---------|
| 1 | 6 fois  |
| 2 | 9 fois  |
| 3 | 5 fois  |
| 4 | 10 fois |

**Q 6: Le rôle de la mémoire centrale d'un ordinateur est :**

- |   |   |
|---|---|
| 1 | Le traitement des instructions                |
| 2 | L'affichage des données                       |
| 3 | L'enregistrement des données                  |
| 4 | Le calcul arithmétique et logique des données |

**Q7: Calculer le nombre d'occurrences d'un élément « elem » dans un tableau T. Si « elem » se trouve trois fois dans T, la variable « Compteur » vaudra 3.**

1

Algorithme 1 :

T:Tableau [1..10] of Integer ;

Compteur:Integer ;

I:Integer;

```

début
  I := 1 ; Compteur := 0 ;
  while I ≤ 10
    début
      Si T[I]=elem alors
        Compteur := Compteur + 1 ;
      Sinon
        I:=I+1;
      finSi
    fin du while
  fin

```

2

Algorithme 2 :

```

T:Tableau [1..10] of Integer ;
Compteur:Integer ;
I:Integer;
début
  I := 1 ; Compteur := 0 ;
  while I ≤ 10
    début
      Si T[I]=elem alors
        Compteur := Compteur + 1 ;
      finSi
      I:=I+1;
    fin du while
  fin

```

3

Algorithme 3 :

```

T:Tableau [1..10] of Integer ;
Compteur:Integer ;
I:Integer;
début
  I := 1 ; Compteur := 0 ;
  while I ≤ 10
    début
      Si T[I]=elem alors
        Compteur := Compteur + 1 ;
        I:=I+1 ;
      finSi
      I:=I+1;
    fin du while
  fin

```

4

Algorithme 4 :

```

T:Tableau [1..10] of Integer ;
Compteur:Integer ;
I:Integer;
début
  I := 1 ; Compteur := 0 ;
  while I ≤ 10
    début
      Si T[I]=elem alors
        Compteur := Compteur + 1 ;
        I:=I+1 ;
      finSi
    fin du while
  fin

```

**Q 8 : Le langage C++ est un :**

- 1 langage purement objet
- 2 langage purement procédural
- 3 langage hybride ( procédural et objet)
- 4 langage fonctionnel

**Q 9 : Afficher 5 fois le message « Bonjour » :**

1 Algorithme 1:

```
I : Integer
début
  I := 0 ;
  while I < 5
  début
    I := I+ 1 ;
    Afficher ("Bonjour") ;
  fin du while
fin
```

2 Algorithme 2 :

```
I : Integer
début
  I := 1 ;
  while I < 5
  début
    Afficher ("Bonjour") ;
    I := I+ 1 ;
  fin du while
fin
```

3 Algorithme 3:

```
I : Integer
début
  I :=0 ;
  while I ≤ 5
  début
    Afficher ("Bonjour") ;
    I := I+ 1 ;
  fin du while
fin
```

4 Algorithme 4 :

```
I : Integer
début
  I :=0 ;
  while I < 5
  début
    Afficher ("Bonjour") ;
    I := I+ 1 ;
  fin du while
fin
```

Q 10 : Le code ASCII qui attribue un chiffre (allant de 0 à 127) aux caractères est représenté par :

1	16 bits
2	8 bits
3	7 bits
4	32 bits

Q11: Recherche du plus petit élément Min dans un tableau T.

1

Algorithme 1 :

```

T:Tableau [1..10] of Integer ;
Min:Integer ;
I:Integer;
début
  I:=1 ; Min := 0 ;
  while I ≤ 10
    début
      Si T[I]<Min alors
        Min := T[I] ;
      finSi
      I:=I+1;
    fin du while
  fin

```

2

Algorithme 2 :

```

T:Tableau [1..10] of Integer ;
Min:Integer ;
I:Integer;
début
  I :=1 ; Min := T[I] ;
  while I ≤ 10
    début
      Si T[I]<Min alors
        Min := T[I] ;
      finSi
      I:=I+1;
    fin du while
  fin

```

3

Algorithme 3 :

```

T:Tableau [1..10] of Integer ;
Min:Integer ;
I:Integer;
début
  I :=1 ; Min := T[1] ;
  while I ≤ 10
    début
      Si T[I]<Min alors
        Min := T[I] ;
      Sinon
        I:=I+1;
      finSi
    fin du while
  fin

```

4

Algorithme 4 :

```

T:Tableau [1..10] of Integer ;
Min:Integer ;
I:Integer;
début
  I:=1 ; Min := T[1] ;
while I ≤ 10
  début
  Si T[I]>Min alors
    Min := T[I] ;
  Sinon
    I:=I+1;
  finSi
fin du while
fin

```

**Q12 : La moyenne des nombres strictement positifs contenus dans un tableau T.**

1

Algorithme 1 :

```

Som, compteur, I : Integer;
moyenne:real;
T:Tableau [1..10] of Integer ;
début
som := 0; compteur:= 0 ; moyenne :=0 ; I := 1;
while I ≤ 10)
  début
  Si T[I] > 0 alors
    som := som + T[I];
    compteur := compteur + 1;
  finSi
  I:=I+1;
fin du while
  Si compteur > 0 alors
    moyenne:=som/compteur;
  finSi
fin

```

2

Algorithme 2 :

```

Som, compteur, I : Integer;
moyenne:real;
T:Tableau [1..10] of Integer ;
début
som := 0; compteur:= 0 ; I := 1;
while I ≤ 10)
  début
  Si T[I] > 0 alors
    som := som + T[I];
    compteur := compteur + 1;
  finSi
  I:=I+1;
fin du while
  moyenne:=som/compteur;
fin

```

3

Algorithme 3 :

```
Som, compteur, I : Integer;
moyenne:real;
T:Tableau [1..10] of Integer ;
début
som := 0; compteur:= 0 ; moyenne :=0 ; I := 1;
while I ≤ 10)
  début
    Si T[I] > 0 alors
      som := som + T[I];
      compteur := compteur + 1;
    finSi
    I:=I+1;
  fin du while
Si compteur > 0 alors
  moyenne:=som/compteur;
finSi
fin
```

4

Algorithme 4:

```
Som, I:Integer;
moyenne:real;
T:Tableau [1..10] of Integer ;

début
som := 0 ; I := 1;
while I ≤ 10)
  début
    som := som + T[I];
    I:=I+1;
  fin du while
  moyenne:=som/10;
fin
```

**Q13 : Echange de deux valeurs. La variable a reçoit la valeur de b et b reçoit la valeur de a . Si a=2 et b=3, après l'échange , on veut que: a=3 et b=2 :**

1

Algorithme 1 :

```
a,b,auxiliare:Integer;
début
  auxiliare:=a;
  a:=b;
  b:= auxiliare ;
fin
```

2

Algorithme 2 :

```
a,b:Integer;
début
  a:=b; b:=a;
fin
```

3

Algorithme 3 :

```
a,b,auxiliare:Integer;
début
  a:=b;
  auxiliare:=a;
  b:= auxiliare ;
fin
```

4

Algorithme 4 :

```

a,b,auxiliaire:Integer;
début
  b:=a;
  auxiliaire:=b;
  a:= auxiliaire ;
fin

```

**Q 14 : Soit la boucle while suivante :**

```

I : Integer ; J : Integer ;
I :=0 ; J:=10
while (I < 10) and (J >0)
début
  Si I ≤ J alors
    Afficher ("Bonjour");
  Sinon
    Afficher ("Bonsoir");
  finSi
  I:=I+1;
  J:=J-1 ;
fin du while.

```

L'exécution de cette boucle affichera :

1	5 fois « Bonjour » et 5 fois « Bonsoir »
2	1 fois « Bonjour » et 1 fois « Bonsoir »
3	5 fois « Bonjour » et 4 fois « Bonsoir »
4	6 fois « Bonjour » et 4 fois « Bonsoir »

**Q15: Calculer et afficher les multiples de 2 inférieur ou égal à 10**

1

Algorithme 1 :

```

i:Integer;
multiple:Integer;

début
i:=0; multiple:=0;
Afficher("les multiples de 2 ≤ 10 sont:");
while multiple ≤ 10
  début
    Afficher(multiple);
    multiple := 2 × i;
    i:=i+1;
  fin du while
fin

```

2

Algorithme 2 :

```
i:Integer;
multiple:Integer;

début
i:=0; multiple:=0;
Afficher("les multiples de 2 ≤ 10 sont:");
while multiple ≤ 10
  début
    Afficher(multiple);
    i:=i+1;
    multiple := 2 × i;
  fin du while
fin
```

3

Algorithme 3 :

```
Les variables:
i:Integer;
multiple:Integer;

début
i:=0; multiple:=0;
Afficher("les multiples de 2 ≤ 10 sont:");
while multiple ≤ 10
  début
    multiple := 2 × i;
    Afficher(multiple);
    i:=i+1;
  fin du while
fin
```

4

Algorithme 4 :

```
Les variables:
i:Integer;
multiple:Integer;

début
i:=1; multiple:=0;
Afficher("les multiples de 2 ≤ 10 sont:");
while multiple ≤ 10
  début
    Afficher(multiple);
    i:=i+1;
    multiple := 2 × i;
  fin du while
fin
```

# Document de réponse

## Partie Informatique

### SERIE 4

	Réponse :1,2,3 ou 4	Note
Q1		
Q2		
Q3		
Q4		
Q5		
Q6		
Q7		
Q8		
Q9		
Q10		
Q11		
Q12		
Q13		
Q14		
Q15		