

## Examen Médian Corrigé

(Durée : 1 heure 30, documents non autorisés)

*Attention : chaque partie doit être rédigée sur une copie séparée*

### 1<sup>ère</sup> partie : numération, diagrammes de Conway, algorithmes (7 points)

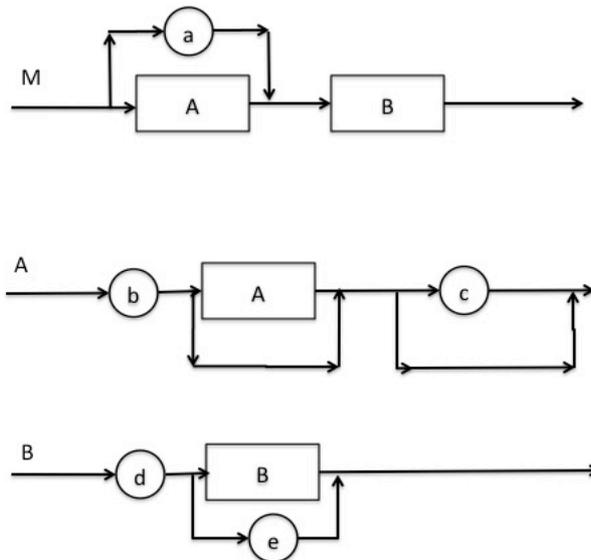
#### 1.1 Numération (2 points)

Compléter le tableau suivant en effectuant les différents changements de base :

Base 10	Base 2	Base 8	Base 16
513	100000001	1001	201
123	1111011	173	7B
479	111011111	737	1DF
3133	110000111101	6075	C3D

#### 1.2 Diagrammes de Conway (2 points)

a) Définir le langage exprimé par les diagrammes suivants, en donnant une formule :



$M=AB$  ou  $aB$

$A=b^i c^j$  avec  $i \in \mathbb{N}^*$  et  $j \in \mathbb{N}$  (et  $j \leq i$ )

$B=d^k e$  avec  $k \in \mathbb{N}^*$

$M = a^u (b^i c^j)^{1-u} d^k e$  avec  $i \in \mathbb{N}^*$ ,  $j \in \mathbb{N}$ ,  $j \leq i$ ,  $k \in \mathbb{N}^*$ ,  $u \in \{0,1\}$

b) Donner des exemples de phrases acceptées par ce langage et des exemples de phrases refusées.

Exemples acceptés :

- ad
- bcccdde
- bbccddde

Exemples refusés :

- abc
- aaab
- abdeeeee

### 1.3 Algorithmes (3 points)

a) Ecrire un algorithme qui convertit les degrés Fahrenheit en degrés centigrades. Dans votre algorithme vous devrez permettre à l'utilisateur de faire plusieurs conversions.

La formule à utiliser est :

$$\theta_C = \frac{5}{9}(\theta_F - 32)$$

Algorithme Fahrenheit

Variables  $\theta_c$ ,  $\theta_f$  de type réel  
nouvelleConversion de type booléen  
reponse de type caractère

Début

```
nouvelleConversion ← vrai
tant que nouvelleConversion faire
    Afficher « Entrer la température en degrés Fahrenheit »
    Lire  $\theta_f$ 
     $\theta_c \leftarrow (5/9) * (\theta_f - 32)$ 
    Afficher « La température en degrés centigrades vaut : »  $\theta_c$ 
    Afficher « Voulez-vous faire une nouvelle conversion : o/n »
    Lire réponse
    si reponse='o' alors nouvelleConversion ← vrai
    sinon nouvelleConversion ← faux
finsi
finTantque
```

Fin

b) Un magasin de reprographie facture 0,10 E les dix premières photocopies, 0,09 E les vingt suivantes et 0,08 E au-delà. Ecrivez un algorithme qui demande à l'utilisateur le nombre

de photocopies effectuées et qui affiche la facture correspondante. Attention à faire en sorte que le nombre de photocopies à faire soit bien positif.

### Algorithme Reprographie

Variables nbPhotocopies de type entier

facture de type réel

Début

répéter

Afficher « Nombre de photocopies effectuées »

Lire nbPhotocopies

jusqu'à ce que nbPhotocopies>0

si nbPhotocopies <= 10 alors

facture ← nbPhotocopies\*0.1

sinon si nbPhotocopies <= 30 alors

facture ← 10\*0.1+(nbPhotocopies-10)\*0.09

sinon

facture ← 10\*0.1+20\*0.09+(nbPhotocopies-30)\*0.08

finsi

finsi

Afficher « La facture vaut : » facture

Fin

----- { prendre une nouvelle copie } -----

## 2<sup>ème</sup> Partie : Itérations (7 points)

1. Qu'affiche le programme suivant : (1 point)

```
program Test;
var
  s, j : integer;
begin
  s := 0;
  for j:= 1 to 5 do
  begin
    write(j:3);
    s := s + j
  end;
  writeln(s:3);
end.
```

1 2 3 4 5 15

2. Quelles sont les valeurs de i et x à la sortie de la boucle suivante : (0,5 point)

```
x:=12 ; i:=0 ;
repeat
  x:=x*2;
  i:=i+1;
until x > 85;
```

```
i = 3
x = 96
```

3. Pourquoi la boucle suivante se termine-t-elle ? (1 point)  
Quelles sont les valeurs de  $i$  et  $x$  à la sortie ?

```
i:=0; x:=23;
repeat
  i:=i+1;
  x:=x-2;
until (x > 30) or (i >= 10);
```

Parce qu'il suffit qu'une des 2 conditions après until soit vérifiée. Comme  $i$  est incrémenté de 1 à chaque passage dans la boucle, la boucle s'arrêtera lorsque  $i$  sera égal à 10.

A la sortie :

```
i = 10
x = 3
```

4. Quelle est la valeur de  $x$  à la sortie de la boucle suivante ? Combien de fois la boucle s'exécute-t-elle ? (1 point)

```
x:=2;
while x < 100 do
  x:=x + x;
```

```
x = 128
La boucle s'exécute n fois avec  $2^{n+1} > 100$ 
donc n = 6
```

5. Qu'affiche l'extrait de programme suivant ? (1 point)

```
for i := 1 to 4 do
begin
  for j :=1 to i do write(j) ;
  for j :=i-1 downto 1 do write(j) ;
  writeln ;
end ;
```

```
1
1 2 1
1 2 3 2 1
1 2 3 4 3 2 1
```

6. On dispose des notes de 25 élèves. Chaque élève peut avoir une ou plusieurs notes (mais toujours au moins une). Ecrire un programme en Pascal permettant d'obtenir la moyenne des notes de chaque élève, ces notes étant saisies au clavier. On souhaite que les données et les résultats se présentent de la façon suivante : (2,5 points)

```
Notes de l'élève numéro 1
12
10
-1
Moyenne : 11
```

Notes de l'élève numéro 2

.

*N.B. : Les parties en italiques correspondent aux données saisies par l'utilisateur. La valeur -1 indique la fin de saisie des notes pour un élève.*

```
program Notes ;
const
    N=25 ;
var
    i : integer ;
    note, nbNotes, somme : integer ;

begin
    for i :=1 to N do
        begin
            writeln ('Notes de l''élève numéro ', i) ;
            somme := 0 ;
            nbNotes := 0 ;
            repeat
                readln(note) ;
                somme := somme + note ;
                nbNotes := nbNotes + 1 ;
            until note = -1 ;
            somme := somme + 1 ;
            nbNotes := nbNotes - 1 ;
            writeln('Moyenne : ', somme / nbNotes) ;
            writeln ;
        end ;
    end.
```

Ou avec une boucle while :

```
program Notes ;
const
    N=25 ;
var
    i : integer ;
    note, nbNotes : integer ;

begin
    for i :=1 to N do
        begin
            writeln ('Notes de l''élève numéro ', i) ;
            somme := 0 ;
            nbNotes := 0 ;
            readln(note) ;
            while (note <> -1) do
```

```

begin
    nbNotes := nbNotes + 1 ;
    somme := somme + note ;
    readln(note) ;
end ;
if (nbNotes > 0) then
    writeln('Moyenne : ', somme / nbNotes)
else
    writeln('Pas de notes') ;
writeln ;
end ;
end.

```

----- { prendre une nouvelle copie } -----

### 3<sup>ème</sup> partie : Algorithmes de conversion (6 points)

Un nombre dans une base  $n$  donnée s'écrit sous la forme d'addition des puissances successives de cette base. Le nombre  $c_n \dots c_2 c_1 c_0$  en base  $b$ , constitué des chiffres  $c_n, \dots, c_2, c_1, c_0$  peut aussi s'écrire sous la forme :  $c_n b^n + \dots + c_2 b^2 + c_1 b^1 + c_0 b^0$ , c'est-à-dire un polynôme dont les coefficients sont les chiffres et l'inconnue est la base.

*Exemple :*

Le nombre 122 en base 10, constitué des chiffres 1, 2, 2 peut s'écrire sous la forme :

$$1 * 10^2 + 2 * 10^1 + 2 * 10^0$$

Lorsqu'on veut passer d'une base à une autre, on utilisera deux méthodes suivant que l'on sait calculer dans la base de départ ou dans la base d'arrivée.

#### Méthode A

Si on sait calculer dans la base de **départ**, des **divisions entières** successives par la base donneront en **reste** les chiffres du résultat, en commençant par les unités.

*Exemple :*

Le nombre 13 en base 10 se traduira par 1101 en base 2 en suivant

$$13 \bmod 2 = 1 \text{ // quotient} = 6$$

$$6 \bmod 2 = 0 \text{ // quotient} = 3$$

$$3 \bmod 2 = 1 \text{ // quotient} = 1$$

$$1 \bmod 2 = 1 \text{ // quotient} = 0 \Rightarrow \text{STOP}$$

Donnez l'algorithme correspondant à cette méthode pour la conversion d'un nombre en base 10 dans une base d'arrivée comprise entre 2 et 9 (base qui sera demandée à l'utilisateur).

On supposera que la fonction puissance est connue.

Algo A

Variables res, n, i, base, nombre : entiers

Début

  Repete

    Afficher 'entrer la base d'arrivée'

  Jusqu'à  $2 \leq \text{base} \leq 9$

  Afficher 'entrez le nombre à convertir'

  Lire (nombre)

  Res <- 0

  i <- 0

  répéter

    res <- res + nombre mod base \* puissance (10, i)

    i <- i + 1

    nombre <- nombre div base

  jusqu'à nombre = 0

  Afficher (res)

Fin

## Méthode B

Si on sait calculer dans la base **d'arrivée**, on évalue le polynôme (en représentant les coefficients et la base de départ dans la base d'arrivée).

*Exemple :*

Le nombre 1101 en base 2 se traduira par 13 en base 10 de la façon suivante :

$$1 * 2^3 + 1 * 2^2 + 0 * 2^1 + 1 * 2^0 = 13$$

Donnez l'algorithme correspondant à cette méthode pour la conversion d'un nombre d'une base comprise entre 2 et 9 (demandée à l'utilisateur) en base 10. Pour des raisons de simplicité, on pourra demander à l'utilisateur de saisir le nombre de chiffres du nombre à convertir, puis les chiffres constituant ce nombre un par un.

Algo B

Variables res, n, i, base, taille : entiers

Début

  Repete

    Afficher 'entrer la base de départ'

  Jusqu'à  $2 \leq \text{base} \leq 9$

  Afficher 'entrez la taille du nombre en base 2'

  Lire (taille)

  Res <- 0

  Pour i allant de taille-1 à 0 faire

    Afficher 'entrer le chiffre en position de gauche à droite', taille-1-i

    Lire (n)

    Res <- res + n \* puissance(base, i)

  Fpour

  Afficher (res)

Fin

