

## Examen Médian

Durée : 1 heure 30

Polycopié papier autorisé, autres documents interdits.  
Calculatrices, téléphones, traducteurs et ordinateurs interdits.

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

### 1ère partie : choix (5 points)

#### 1.1. Quelle salle pour le médian ? (1,5 point)

Ecrire un programme en Pascal qui demande à un étudiant de NF01 la première lettre de son nom de famille et qui lui indique la salle dans laquelle a lieu le médian. On supposera que les étudiants dont la première lettre du nom est comprise entre 'A' et 'H' doivent aller dans la salle FA501, ceux dont la première lettre est comprise entre 'I' et 'P' dans la salle FA502 et les autres dans la salle FA503.

```
program salle;
var
  lettre : char ;
  nSalle : string;

begin
  repeat
    writeln('Quelle est la première lettre de votre Nom ');
    readln(lettre);
  until (lettre >= 'A') AND (lettre <= 'Z');
  case lettre of
    'A'..'H' : nSalle := 'FA501';
    'I'..'P' : nSalle := 'FA502';
    'Q'..'Z' : nSalle := 'FA503';
    // otherwise nsalle := 'inconnue'; // si pas de gestion avec
  repeat
    end;
  writeln('Votre médian aura lieu en salle ', nSalle) ;
end.
```

#### 1.2. Un programme mystérieux (0,5 point)

On considère le programme Pascal suivant :

```

program Mystere ;
var
    boule, de, gomme : integer ;
begin
    writeln('Saisir deux valeurs comprise entre 0 et 20');
    readln(boule, de);
    gomme := boule;
    if boule < 10 then
        gomme := -boule;
        writeln(gomme);
        if de < 5 then
            gomme := gomme - de;
    writeln('Resultat', gomme);
end.

```

- a) Quel est le résultat de ce programme si l'utilisateur saisit 2 et 3 ? -5  
b) Quel est le résultat de ce programme si l'utilisateur saisit 15 et 2 ? 13

### 1.3. Le jeu de Chifoumi (3 points)

Le jeu de Chifoumi, aussi appelé caillou-ciseaux-papier, se joue à deux joueurs avec les mains. Simultanément, les deux joueurs font un signe avec leur main qui représente soit un caillou, soit des ciseaux, soit un papier. Si on nomme les joueurs A et B, les règles sont les suivantes :

- Si A et B font le même signe, il y a égalité, aucun des deux joueurs ne marque de point.
- Si le joueur A joue Caillou et le joueur B Ciseaux, A marque un point, car « le caillou émousse les ciseaux », et réciproquement.
- Si A joue Papier et B joue Caillou, A marque un point, car « le papier enveloppe le caillou », et réciproquement.
- Si A joue Ciseaux et B joue Papier, A marque un point car « les ciseaux coupent le papier », et réciproquement.

- 1) Ecrire un programme Pascal qui permet à l'utilisateur de saisir les deux coups joués par A et B et qui affiche le nom du joueur qui marque un point. (1,5 point)
- 2) Modifier le programme précédent pour ajouter un score à A et B, et arrêter la partie quand l'un des joueurs atteint 5 en indiquant qui est le gagnant. (1,5 point)

```

program Chifoumi;
var
    jA, jB : string;
    pointsA, pointsB : integer;

begin
    pointsA := 0; pointsB := 0; //Q2

    repeat // Q2
        writeln('-----');
        repeat
            write('Que joue le joueur A (Papier, Ciseaux, Caillou) : ');
            readln(jA);
        until (jA = 'Papier') OR (jA = 'Caillou') OR (jA = 'Ciseaux');
        writeln(jA);

        repeat
            write('Que joue le joueur B (Papier, Ciseaux, Caillou) : ');

```

```

        readln(jB);

until (jB = 'Papier') OR (jB = 'Caillou') OR (jB = 'Ciseaux');

writeln(jB);

if (((jA = 'Caillou') AND (jB = 'Ciseaux')) OR
    ((jA = 'Papier') AND (jB = 'Caillou')) OR
    ((jA = 'Ciseaux') AND (jB = 'Papier'))) then
    pointsA := pointsA + 1
else if (((jB = 'Caillou') AND (jA = 'Ciseaux')) OR
        ((jB = 'Papier') AND (jA = 'Caillou')) OR
        ((jB = 'Ciseaux') AND (jA = 'Papier'))) then
    pointsB := pointsB + 1;
writeln('Points Joueur A : ', pointsA,
        ' - Points Joueur B : ', pointsB);
until (pointsA = 5) OR (pointsB = 5); // Q2

writeln('=====');

if (pointsA = 5) then
    writeln('Joueur A gagne')
else
    writeln('Joueur B gagne');
readln;
end.

```

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

## 2<sup>ème</sup> partie : sapin numérique (5 points)

On souhaite réaliser un programme en Pascal qui donne la possibilité à l'utilisateur d'afficher à l'écran une figure 'sapin numérique' (voir Figure 1 ci-dessous).

```

          1
         1 1
          2
        1 2 1
         2 2
          3
       1 2 3 1
        2 3 2
         3 3
          4
       1 2 3 4 1
        2 3 4 3 2
         3 4 3
          4 4
         5
       1 2 3 4 5 1
        2 3 4 3 2
         3 4 3
          4 4

```

5    5  
6

Figure 1. Sapin numérique

L'utilisateur doit pouvoir définir la taille du sapin (le nombre d'étages est 6 pour la Figure 1).

Pour écrire le programme, on procédera en 3 étapes :

1. Définir le nombre  $n$  ( $1 < n < 10$ ) d'étages du sapin. L'utilisateur peut se tromper dans sa demande ( $n$  en dehors du domaine de définition), mais pas plus de 5 fois.
2. Ecrire un programme Pascal permettant d'afficher à l'écran un étage de branche. Par exemple pour l'étage 4, le programme affichera :

1                    1  
  2                    2  
    3                  3  
     4

3. Afficher le sapin avec le nombre d'étages demandé par l'utilisateur en utilisant l'étape 2.

```

program sapin;
const
  NMAX=9;
var
  i,j,n,m,k,nb : integer;
begin
  nb:=0;
  repeat
    writeln('1<m<NMAX');
    read(m);
    nb:=nb+1;
  until (m in [2..NMAX]) or (nb > 5);

  if nb <= 5 then
  begin
    readln;
    for n:=1 to m do
    begin
      for i := 1 to n-1 do
      begin
        for k:=1 to m-n do write(' ');
        for j:= 1 to i-1 do write(' ');
        write(i);
        for j:= 1 to 2*(n-i)-1 do write(' ');
        writeln(i);
      end;
      for j:= 1 to m-1 do write(' ');
      writeln(n);
    end;
    readln;
  end
end

```

```

else
  writeln ('Trop d''essais');
readln;
end.

```

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

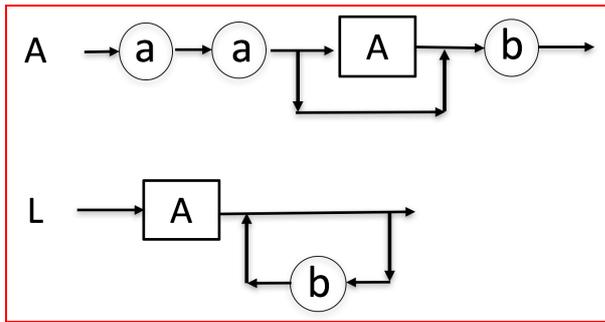
### 3<sup>ème</sup> partie : diagrammes de Conway, expressions (5 points)

#### 1) Diagrammes de Conway (3 points)

Ecrire le diagramme de Conway associé au langage défini par la formule syntaxique suivante :  $a^{2n} b^m$  avec  $0 < n \leq m$

On pourra utiliser le fait que :

$$\exists p \geq 0 \text{ tel que } b^m = b^{n+p}$$

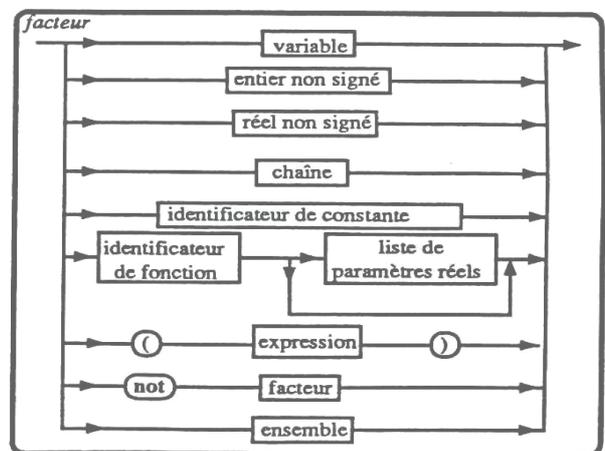
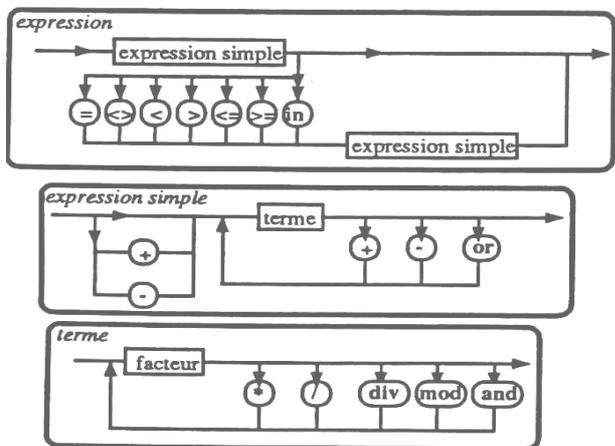


#### 2) Expressions (2 points)

En utilisant les diagrammes de décomposition syntaxique du Pascal ci-dessous, déterminer si les expressions suivantes sont correctes :

- $0 < x < 1$
- `not (x = y)`

Indiquer le chemin de décomposition et préciser, en cas de succès, l'ordre de priorité des opérations.



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

#### 4<sup>ème</sup> partie : Tableaux (5 points)

- a) Ecrire un algorithme permettant à l'utilisateur de saisir un tableau de 8 caractères, puis réalisant un décalage circulaire vers la gauche des éléments de ce tableau.

*Exemple :*

Tableau initial : D E C A L A G E

Tableau modifié (décalage à gauche) : E C A L A G E D

```
programme Decalage_gauche
variables tmp : caractere
          i : entier
          T : tableau de 10 caractères
```

Début

```
  Pour i allant de 1 à 10 faire
    Lire T[i]
  Fin pour
  tmp <- T[1]
  Pour i allant de 1 à 9 Faire
    T[i] <- T[i+1]
  Fin pour
  T[10] <- tmp
Fin
```

- b) Ecrire un algorithme permettant à l'utilisateur de saisir un tableau de N entiers, et testant ensuite si ce tableau est trié en ordre croissant.

```
programme Vérification_Tableau_Trie
variables i : entier
          T : tableau de 10 entiers
```

Début

```
  i <- 1
  Tant que i < 10 ET T[i] <= T[i+1] Faire
    i <- i + 1
  Ftq
  Si i = 10 alors
    écrire 'Le tableau est trié'
  Sinon
    écrire 'Le tableau est non trié'
  Fsi
Fin
```