

## Examen Médian

Corrigé

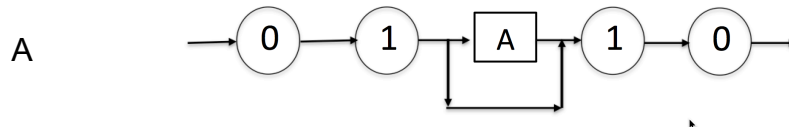
(Durée : 1 heure 30, documents non autorisés)

Attention : chaque partie doit être rédigée sur une copie séparée

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

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

a) Définir par une formule le langage exprimé par le diagramme de Conway suivant :



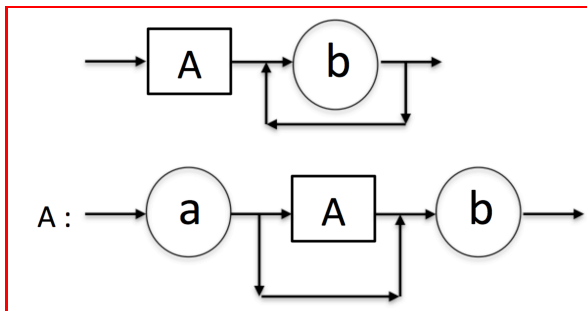
$(01)^n (10)^n$  avec  $n > 0$

Donner 3 exemples de mots générés par ce langage.

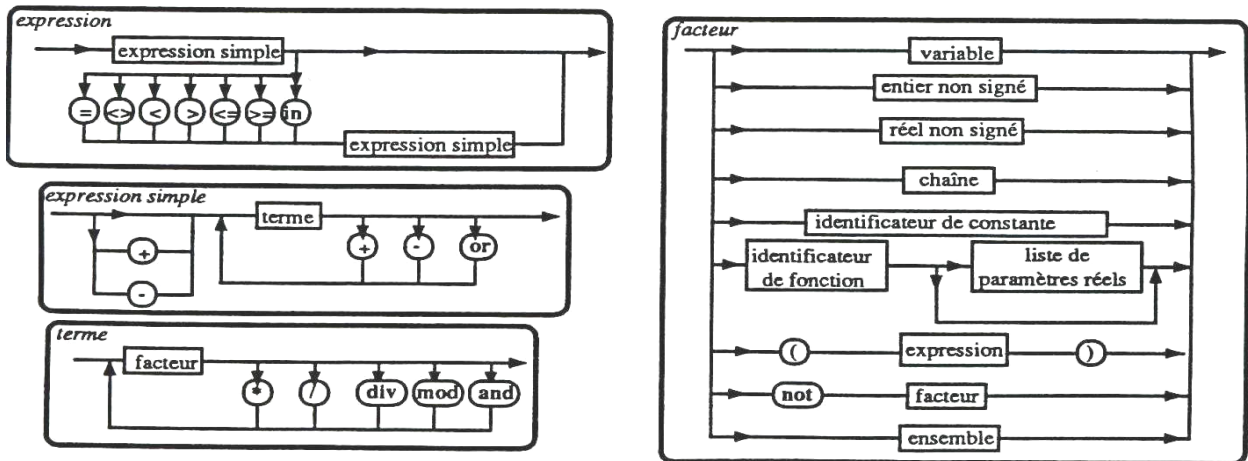
0110, 01011010, 010101101010

b) Ecrire le diagramme de Conway associé au langage défini par la formule syntaxique suivante :

$a^n b^m$  avec  $0 < n < m$  (on pourra utiliser le fait que  $\exists p > 0$  tel que  $b^m = b^{n+p}$ )



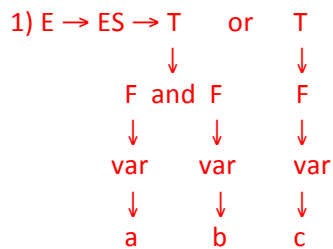
## 2) Expressions (2 points)



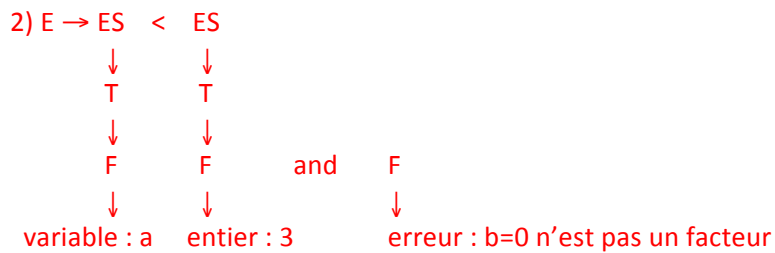
En utilisant les diagrammes de décomposition syntaxique du Pascal ci-dessus, déterminer si les expressions suivantes sont correctes :

- 1) a and b or c
- 2) a < 3 and b = 0

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



Correct. Priorité : (a and b) or c



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

## 2<sup>ème</sup> partie : tests (6 points)

### 1) Tri de 3 nombres (1 point)

Ecrire un programme Pascal qui classe trois nombres du plus petit au plus grand.

```

program tri ;
var
  d1, d2, d3 : integer ;
  
```

```

begin
  if d1 > d2 then
    begin
      temp := d1; d1 := d2; d2 := temp;
    end;
  if d1 > d3 then
    begin
      temp := d1; d1 := d3; d3 := temp;
    end;
  if d2 > d3 then
    begin
      temp := d2; d2 := d3; d3 := temp;
    end;
end.

```

## 2) Mini Yams (2 points)

Ecrire un programme Pascal simulant le lancement de trois dés et affichant selon le résultat obtenu :

- « brelan » si la valeur des trois dés est la même ;
- « paire » si la valeur de deux des trois dés est identique ;
- « suite » si les trois dés se suivent.

Pour obtenir des valeurs aléatoires, on pourra utiliser la fonction `random(n)` qui retourne un nombre aléatoire compris entre 0 et n-1.

```

program miniYams;
var
  d1, d2, d3 : integer;
  temp : integer;
  r : char;
begin
  Randomize;
  writeln('===Yams===');
  repeat
    writeln('Nouvelle Partie');
    d1 := 1 + random(6);
    d2 := 1 + random(6);
    d3 := 1 + random(6);
    write('d1 : '); readln(d1);
    write('d2 : '); readln(d2);
    write('d3 : '); readln(d3);
    writeln('Valeurs des dés : ', d1, ', ', d2, ', ', d3);
    if ((d1 = d2) and (d1 = d3)) then
      writeln('Brelan')
    else if ((d1 = d2) or (d1 = d3) or (d2 = d3)) then
      writeln('Carre')
    else
      begin
        //classement dans l'ordre croissant : cf question 1
        if d1 > d2 then
          begin
            temp := d1; d1 := d2; d2 := temp;
          end;
        if d1 > d3 then
          begin
            temp := d1; d1 := d3; d3 := temp;
          end;
        if d2 > d3 then
          begin

```

```

        temp := d2; d2 := d3; d3 := temp;
    end;

    //Vérification qu'ils se suivent

    if ((d2 = d1+1) and (d3 = d2+1)) then
        writeln('Suite')
    else
        writeln('Rien');
    end;
    writeln('Rejouez o-n');

    readln(r);
until r = 'n';
readln;
end.

```

### 3) Test branche (3 points)

Vous devez écrire un programme Pascal de type « test d'orientation » pour savoir si un étudiant est fait ou pas pour aller en GI.

Pour simplifier, on considérera seulement les trois questions (un peu caricaturales) suivantes :

- Q1 : Aimez-vous la logique ? (a : pas du tout, b : un peu, c : beaucoup)
- Q2 : Accordez-vous de l'importance à la sociabilité de vos camarades ? (a : pas du tout, b : un peu, c : beaucoup)
- Q3 : L'importance de votre salaire à la première embauche vous importe t-elle ? (a : pas du tout, b : un peu, c : beaucoup)

Le programme proposera, pour un maximum de :

- a : une réorientation,
- b : n'importe quelle branche,
- c : une orientation en GI.

Si l'étudiant, tape un autre choix que a, b ou c, vous choisirez une des options suivantes :

- ne pas lui permettre de rentrer un autre choix
- le mettre automatiquement dans le cas a (i.e. s'il ne sait même pas appuyer sur a, b, ou c, que faire de lui ?)

*N.B. : Pour gagner du temps, vous pouvez écrire Q1, Q2, Q3 sans le détail exact des questions.*

```

program test;
var
    a, b, c : integer;
    choix : char;
begin
    a := 0; b := 1 ; c := 1;
    writeln('Test - êtes vous fait pour aller en GI -') ;
    writeln('Accordez-vous de l'importance à la sociabilité de vos
camarades? a-pas du tout b-un peu c-beaucoup :');
    readln(choix);
    case choix of
        'a' : a := a+1;
        'b' : b := b+1;
        otherwise c := c+1;
    end;
end;

```

```

        writeln('L'importance de votre salaire à la première embauche
vous importe t-elle? a-pas du tout b-un peu c-beaucoup :');
        readln(choix);
        case choix of
            'a' : a := a+1;
            'b' : b := b+1;
            otherwise c := c+1;
        end;

        writeln('Priorisez-vous un apprentissage déductif? a-pas du tout,
je prefere le par coeur b-un peu c-beaucoup :');
        readln(choix);
        case choix of
            'a' : a := a+1;
            'b' : b := b+1;
            otherwise c := c+1;
        end;

        if (a > b) and (a > c) then
            writeln('Pensez à vous réorienter')
        else if (b > a) and (b > c) then
            writeln('Vous pouvez faire une autre branche mais c'est
moins bien')
        else
            write('Bienvenu au GI :) ');
        readln;
    end.

```

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

### 3<sup>ème</sup> partie : que fait ce programme ? (4 points)

Soit l'algorithme ci-dessous :

```

Début
lire(n)
Pour i allant de 1 à n faire
    j ← 1
    k ← 0
    Tant que k < i*i faire
        afficher j sur la ligne courante
        j ← j+1
        k ← k+j
    Fin Tant que
    Changer de ligne
Fin Pour
Fin

```

1) Ecrire le programme pascal correspondant

```

program partie3;
var
    i , j, n, k : integer;

begin

```

```
writeln('Veuillez entrer le nombre de lignes :');
readln(n);
for i := 1 to n do begin
  j := 1;
  k := 0;
  while k < i*i do begin
    write(j, ' ');
    j := j + 1;
    k := k + j;
  end;
  writeln;
end;
end.
```

2) Donner un exemple de sortie pour n = 5

```
1
1 2
1 2 3
1 2 3 4 5
1 2 3 4 5 6
```

3) Que fait ce programme ?

Il affiche n lignes à l'écran, une ligne i étant composée des nombres allant de 1 à j, tel que  $\sum_{k=1}^j k < i^2$

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

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

On souhaite afficher le carré suivant à n lignes (ici : n = 5) :

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

L'utilisateur doit pouvoir choisir le nombre n de lignes. Votre programme doit vérifier que n ne dépasse pas la taille maximum NMAX autorisée.

1. Ecrire le programme Pascal correspondant avec NMAX < 10.  
N.B. : il y a un seul espace entre deux chiffres.
2. Ecrire le programme Pascal avec (NMAX < 100). L'affichage doit respecter la présentation en colonne comme dans le cas n=5. Mais pour n = 10 l'affichage devra être le suivant :

```
1 2 3 4 5 6 7 8 9 10
2 3 4 5 6 7 8 9 10 1
3 4 5 6 7 8 9 10 1 2
...
10 1 2 3 4 5 6 7 8 9
```

*N.B. : vous devez trouver le nombre de chiffres de  $n$  afin de pouvoir aligner correctement les colonnes. Vous pouvez utiliser pour cela les fonctions `div` et `mod`.*

```
program iterations;
const
    NMAX =100 ;
var
    n : integer ;
    i,j,k,temp:integer;

begin
    repeat
        writeln('Nombre de lignes ?');
        readln(n);
    until n <= NMAX;

    temp:=n;
    i:=0;
    repeat
        temp:=temp div(10);
        i:=i+1;
    until (temp=0);
    writeln ('nombre de chiffres : ',i);

    for j:=0 to n-1 do
        begin
            for k:=j+1 to n do write(k:i+1);
            for k:= 1 to j do write(k:i+1);
            writeln;
        end;
    end.
end.
```