

## 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 : prix de location d'un véhicule (5 points)

Une agence de location de véhicules prévoit d'installer des bornes libre-service pour optimiser l'accueil de ses clients. Le client commencera le processus de location sur la borne et le terminera au guichet pour la signature du contrat.

Vous avez la mission d'écrire le programme qui équipera ces bornes, et vous commencez par la partie dialogue avec le client. Il s'agit d'afficher à l'écran le prix final de location, qui sera calculé en fonction des caractéristiques et des choix du client. Le prix de base par journée de location est :

- 20 € pour une voiture du type 1,
- 30 € pour une voiture du type 2,
- 40 € pour une voiture du type 3.

Ce prix est majoré de 15% pour les conducteurs âgés de moins de 25 ans (toutefois, dans le cas où le conducteur est une femme, la majoration n'est que de 10%), et âgés de plus de 55 ans.

Les conducteurs ayant plus de 10 ans de permis ont une réduction de 5% et ceux ayant moins de 5 ans ont une majoration de 5% (pour les femmes, la majoration n'est que de 3%).

Le prix final change aussi en fonction du nombre de journées de location : après la 7ème journée de location, le prix décroît de 2 € par semaine de location.

Ecrire un programme Pascal qui pose des questions au client et affiche ensuite le prix total de location, en l'invitant soit à aller au guichet pour signer le contrat soit à abandonner l'opération.

**program location;**

**var**

**voiture, age, permis, nb\_jours, nb\_semaines : integer;**  
**prix\_base, prix\_total, fact : real;**  
**sexe, rep : char;**

**begin**

**writeln('=====');**  
**writeln('Bienvenue à !!!!!!! Rent a Car !!!!!!!');**  
**writeln('Veuillez faire vos choix de location et en suite valider le prix proposé afin**  
**d"aller signer votre contrat au guichet');**  
**writeln('=====');**  
**writeln('Quel type de voiture voulez-vous louer ? (Type 1 : 20,00 par journée; Type**  
**2 : 30,00; Type 3 : 40,00)');**  
**write('Type : ');**  
**readln(voiture);**

```

writeln('Quel âge avez-vous ?');
readln(age);
writeln('Etes-vous un homme (H) ou une femme (F) ?');
readln(sexe);
writeln('Depuis combien d'années avez-vous le permis ?');
readln(permis);
writeln('Vous voulez louer la voiture pour combien de jours ?');
readln(nb_jours);
case voiture of
    1 : prix_base := 20;
    2 : prix_base := 30;
    3 : prix_base := 40;
end;
fact := 1;
if (age < 25) OR (age > 55) then
    if (age < 25) AND (sexe = 'F') then
        fact := 1.10
    else
        fact := 1.15;
if permis > 10 then
    fact := fact*0.95
else
    if permis < 5 then
        if sexe = 'F' then
            fact := fact*1.03
        else
            fact := fact*1.05;
{calcul du prix total}
prix_total := prix_base*fact*nb_jours;

{calcul du nb de semaines}
nb_semaines := nb_jours div 7;

{calcul du prix total après réduction par semaine de location (après la 7ème
journée)}
if nb_semaines >= 1 then
    prix_total := prix_total - (nb_semaines - 1)*2;

writeln('Le prix total est : ', prix_total:5:2);
writeln('Vous êtes d'accord (o/n) ?');
readln(rep);
if rep = 'o' then
    writeln('Dirigez-vous au guichet pour la signature de votre contrat')
else
    writeln('Nous vous remercions de votre choix. A bientôt sur Rent a Car !');

    readln(rep);
end.

```

## 2<sup>ème</sup> partie : dessin de lettres (5 points)

On vous demande d'écrire un programme en Pascal qui affiche à l'écran les lettres L, U ou I à l'aide du caractère étoile : \* .

L'utilisateur doit pouvoir choisir :

- la taille de la lettre (fenêtre d'affichage de dimension maximale 100 x 100 caractères)
- la lettre à afficher (L, U ou I)

Votre programme doit donner à l'utilisateur la possibilité d'afficher plusieurs lettres s'il le désire.

N.B. : La lettre U sera affichée avec des coins inférieurs non arrondis.

```
program lettre;
var
  choix : char;
  l, c, dim : integer;

begin

  repeat
    writeln('L, U, I');
    read(choix);
  until (choix = 'L') or (choix = 'U') or (choix = 'I');

  writeln('dimension de l''affichage < 100');
  read (dim);

  case of choix :
    'I' : for i := 1 to dim do writeln( '*');

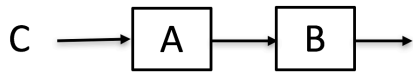
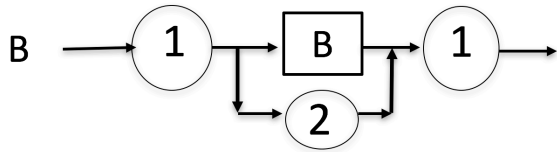
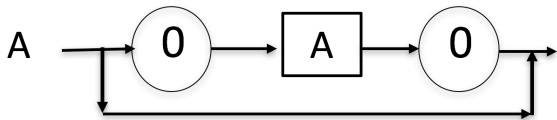
    'U' : begin
      for i := 1 to (dim-1) do
        begin
          write ('*');
          for c := 1 to dim -2 do write(' ');
          writeln('*');
        end;
      for c:=1 to dim do write ('*')
    end;

    'L': begin
      for i := 1 to (dim - 1) do writeln ('*');
      for c := 1 to dim do write ('*');
      writeln;
    end;
  end.
end.
```

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

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

On considère les diagrammes de Conway ci-dessous :



Définir à l'aide d'une formule le langage généré par C.

Réponse :  $0^{2n}1^m21^m$ ,  $m > 0$ ,  $n \geq 0$

Donner deux exemples de mots acceptés et deux exemples de mots refusés

Acceptés : 121, 0011211, 0000121

Refusés : 0121, 001211

## 2) Expressions (2 points)

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

- $2 * \cos(x+3) + \sin(x)$

$E \rightarrow ES \rightarrow T + T \rightarrow F * F + T \rightarrow \text{EntierNS} * F + T \rightarrow \text{EntierNS} * \text{idFn Params} + T \rightarrow \text{EntierNS} * \text{idFn}(E) + T \rightarrow \text{EntierNS} * \text{idFn}(ES) + T \rightarrow \text{EntierNS} * \text{idFn}(T+T) + T \rightarrow \text{EntierNS} * \text{idFn}(F + F) + T \dots \rightarrow \text{EntierNS} * \text{idFn}(\text{var} + \text{EntierNS}) + \text{idFn}(\text{var})$

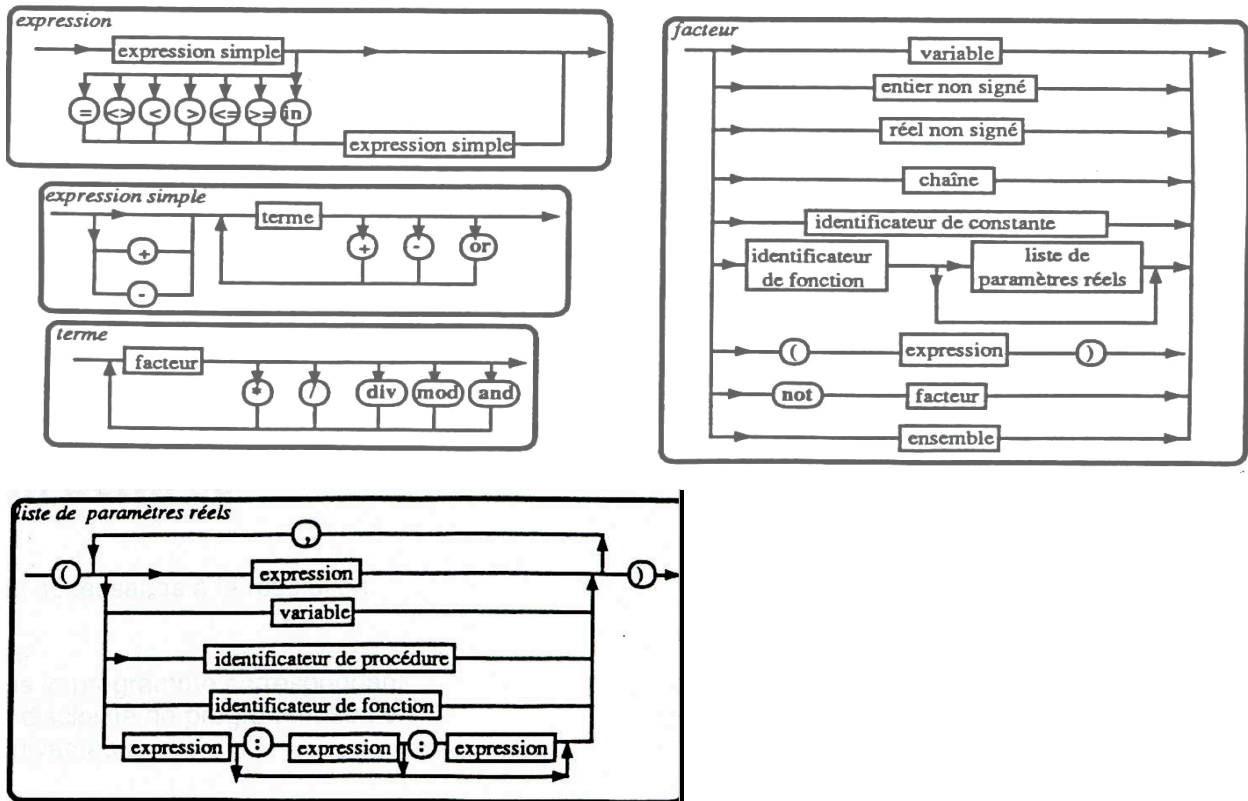
Priorité : +, \*, +

- $a = b \text{ and } (a = c + 1)$

$E \rightarrow ES = ES \rightarrow T = ES \rightarrow F = ES \rightarrow \text{var} = ES \rightarrow \text{var} = T \rightarrow \text{var} = F \text{ and } F \rightarrow \text{var} = \text{var} \text{ and } (E) \rightarrow \text{var} = \text{var} \text{ and } (ES = ES) \dots \rightarrow \text{var} = \text{var} \text{ and } (\text{var} = \text{var} + \text{EntierNS})$

Priorité : addition, =, and, =

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



#### 4<sup>ème</sup> partie : « Le Savetier et le Financier » (5 points)

Le Savetier<sup>1</sup> rencontra Le Financier, et lui dit : « Ce mois-ci, qui compte 30 jours, je te propose le marché suivant : - je te verse 100 écus le 1<sup>er</sup> jour, 150 écus le 2<sup>ème</sup> jour, 200 écus le 3<sup>ème</sup> jour, etc... en ajoutant 50 écus chaque jour par rapport à la veille. - En échange, tu me donnes 1 écu le 1<sup>er</sup> jour, 2 écus le 2<sup>ème</sup> jour, 4 écus le 3<sup>ème</sup> jour, etc... en doublant à chaque fois ce que tu m'as donné la veille. L'engagement dure un mois ! Es-tu prêt à signer le contrat ? » « Quel imbécile ! » se dit Le Financier, qui accepta sans sourciller le contrat.

(D'après une fable de La Fontaine)

On définit en Pascal, le début du programme suivant :

```

program FabledeLaFontaine;
const
  N = 30;
var
  a, b, s, f : ???; // Vous pouvez utiliser un entier ou un
                    // tableau d'entiers comme vous le souhaitez
  jourd, jourg : integer;

```

Vous allez écrire la suite du programme en considérant que chaque question permet d'écrire les instructions à la suite :

1. Calcul et affichage de la suite  $a$  qui correspond à la somme d'argent donné par le Savetier chaque jour (voir **Erreur ! Nous n'avons pas trouvé la source du renvoi.** : 2<sup>o</sup> colonne). (1 point)

<sup>1</sup> Savetier ancien nom donné aux cordonniers...

On ajoute :

var

i : integer;

```
a := 100;
writeln('Jour 1 : ', a);
for i := 2 to N do
  begin
    a := a + 50;
    writeln('Jour ', i, ' : ', a);
  end;
```

2. Calcul et affichage de la suite  $b$  qui correspond à la somme d'argent donné par le Financier chaque jour (voir **Erreur ! Nous n'avons pas trouvé la source du renvoi.** : 3<sup>e</sup> colonne). (1 point)

```
b := 1;
writeln('Jour 1 : ', b);

for i := 2 to N do
  begin
    b := b*2;
    writeln('Jour ', i, ' : ', b);
  end;
```

3. Calcul et affichage des suites  $s$  et  $f$  qui correspondent aux sommes cumulées d'argent gagnées ou perdues respectivement par le Savetier et le Financier (**Erreur ! Nous n'avons pas trouvé la source du renvoi.** : colonnes 6 et 7). (1 point)

```
writeln('Suite s : Somme cumulée par le Savetier');
a := 100;
b := 1;
s := b - a;
```

```
writeln('Jour 1 : ', s);
for i := 2 to N do
  begin
    a := a + 50;
    b := b*2;
    s := s + b - a;
    writeln('Jour ', i, ' : ', s);
  end;
```

```
writeln('Suite f : Somme cumulee par le Financier');
a := 100;
b := 1;
f := a - b;
writeln('Jour 1 : ', f);
```

```

for i := 2 to N do
begin
  a := a + 50;
  b := b*2;
  f := f + a - b;
  writeln('Jour ', i, ' : ', f);
end;

```

4. Calcul de `jourd`, le jour à partir duquel, le Financier donne plus d'argent que le Savetier (voir **Erreur ! Nous n'avons pas trouvé la source du renvoi.** : le résultat est 11). (1 point)

```

i := 1;
a := 100;
b := 1;
repeat
  i := i + 1;
  a := a + 50;
  b := b*2;
until b > a;
jourd := i;

```

5. Calcul de `jourG`, le jour à partir duquel, le Savetier gagne plus d'argent cumulé que le Financier (voir **Erreur ! Nous n'avons pas trouvé la source du renvoi.** : le résultat est 13). (1 point)

```

i := 1;
a := 100;
b := 1;
s := b - a;
repeat
  i := i + 1;
  a := a + 50;
  b := b*2;
  s := s + b - a;
until s > 0;
jourG := i;

```