

Idées de correction

1. Alternatives (5 points)**Exercice 1.**

On trouve dans un programme C l'instruction suivante dans laquelle les variables a, b et c sont entières et peuvent prendre des valeurs 1 ou 0. La variable x est aussi un entier :

```
if( a ) if( !b )
    if( !a ) x =0; else if( c ) x =1;
    else
        x =0;
else if( !b ) x =1;
else     if( !c ) x =0; else x =1;
        else if( c )
x =1; else x =0 ;
```

1) Réécrire cette instruction en respectant la présentation conseillée en cours.

```
if (a)
    if (!b)
        if (!a) x =0;
        else
            if( c ) x =1;
            else x =0;
    else
        if (!b) x =1;
        else
            if (!c) x =0;
            else x =1;
else
    if (c) x =1;
    else x =0 ;
```

2) Simplifier le plus possible cette instruction.

```
if (c)
    x =1;
else
    x =0 ;
```

Exercice 2 :

Ecrire un programme qui calcule les réductions du tarif d'une formule d'assurance auto comme suit : Le programme demandera le type de couverture souhaité par le client (couverture 1 à 300€, couverture 2 à 400€ et couverture 3 à 450€), le sexe du client, son âge et s'il est étudiant ou chômeur. Ensuite, selon ces données, le programme pourra calculer une réduction éventuelle et/ou une majoration. Les réductions possibles seront calculées de la façon suivante :

- Si le client est une femme, elle a une réduction de 25%
- Si le client a entre 18 et 25 ans, et s'il est étudiant, il a une réduction de 50%, mais s'il n'est pas étudiant la réduction n'est que de 30%
- Si le client a plus de 25 ans, mais est quand même étudiant, alors il a une réduction de 30%
- Si le client est chômeur, il a une réduction de 20%.
- Si le client a plus de 60 ans, le tarif est majoré de 10%
- Si le client a moins de 3 ans de permis, le tarif est majoré de 20%.

Attention, il n'est pas possible de cumuler les réductions. On appliquera la réduction la plus intéressante pour le client. En revanche, les majorations sont applicables et cumulables même si il y a eu une réduction (on appliquera les majorations sur le tarif réduit).

```
#include <stdio.h>
void main()
{
    float tarif;
    char etu, sexe, chom;
    int age, temps_permis, plan;

    do{
        printf("Quelle couverture souhaitez vous avoir (1, 2 ou 3) ?\n ");
        scanf("%d",&plan);
        if(plan == 1)
            tarif = 300;
        else
            if( plan == 2)
                tarif = 400;
            else
                if( plan == 3)
                    tarif = 550;
                else
                    printf("Plan inconnu");
    }while( plan != 1 && plan != 2 && plan != 3);

    printf("Vous etes une femme (F) ou un homme (H)");
    fflush(stdin);
    scanf("%c", &sexe);
    printf("Quel age avez-vous ?");
    scanf("%d", &age);
    printf("Depuis combien d'annees avez vous le permis");
    scanf("%d",&temps_permis);
    printf("Etes vous etudiant (O/N)?");
    fflush(stdin);
    scanf("%c",&etu);
    printf("Etes vous chomeur (O/N)?");
    fflush(stdin);
    scanf("%c",&chom);

    if( age <= 25 && age>18)
        if ( etu == 'O' || etu == 'o')
            tarif *= 0.5;          //jeune etudiant
        else
            tarif *= 0.7;          //jeune non etudiant
    else
        if( etu == 'O' || etu == 'o')
            tarif *= 0.70;          //etudiant plus de 25 ans
        else
            if (sexe == 'F')
                tarif *= 0.75;      //femme plus de 25 ans et non etudiante
            else
                if( chom == 'O' || chom == 'o')
                    tarif *= 0.8;    //chomeur plus de 25 ans masculin

    if( temps_permis < 3)
        tarif *= 1.20;
    if (age>60)
        tarif *= 1.1;

    printf("Votre tarif s'eleve à %6.2f euros",tarif);
}
```

2. Suite de nombres (5 points)

Ecrire un programme qui lit une suite de nombres réels terminée par la valeur 0 et affiche les résultats suivants :

- indique si la suite est croissante, décroissante, monotone ou non ordonnée. Dans le cas d'une suite non ordonnée, le programme s'arrête dès qu'il a pu le détecter.
- indique aussi le nombre de valeurs lues (sans compter le 0).

On supposera que la suite contient au moins 2 valeurs non nulles avant le 0.

```
#include <stdio.h>
#include <stdlib.h>

void main()
{
float nbc,nbp; /* valeurs lues: courante et précédente*/
int nbN; /* nombre de valeurs lues*/
int crois, decr ; /* compteurs de comparaisons croissantes ou décroissantes*/

printf("\ndonner une suite de nombres terminées par 0");
scanf("%f %f",&nbp,&nbc);
nbN=2;
crois=decr=0;
while ((nbc!=0) && (crois==0 || decr == 0)) { /* arret si valeur 0 lue ou suite
non ordonnée*/
    if (nbp<nbc)
        crois++;
    else
        if (nbp>nbc)
            decr++;
    nbp=nbc;
    nbN++;
    scanf("%f",&nbc);
}
if (crois!=0 && decr!=0)
    printf("suite non ordonnee, %d valeurs lues", nbN);
else
    if (crois==0 && decr==0 )
        printf("suite monotone de %d valeurs", nbN-1);
    else
        if (crois!=0 ) printf ("suite croissante de %d valeurs", nbN-1);
        else printf("suite décroissante de %d valeurs", nbN-1);
}
```

3. Pyramide inversée de lettres (5 points)

Ecrire un programme qui affiche une pyramide inversée comme le montre la figure ci-dessous. La hauteur de la pyramide correspondra à la lettre entrée par l'utilisateur, elle devra forcément être testée et comprise entre a et z.

Rappel : on peut utiliser les opérateurs arithmétiques sur les caractères.

```
void main()
{ int i,j;
  char caract_choisi,carac;
  int nb_ligne;

  do {
    printf("Entrez le caractère : ") ;
    scanf("%c", &caract_choisi);
  }
  while( (caract_choisi<'a') || (caract_choisi>'z') ) ;

  nb_ligne = (caract_choisi - 'a') + 1 ;
  carac = 'a';
  for ( i=1 ; i<=nb_ligne ; i ++ ) {
    for (j=1 ; j<i; j++)
      printf(" ");
    for (j=i ; j<=2*nb_ligne -1 ; j++)
      printf("%c",carac) ;
  }
```

```

    printf("\n") ;
    caract++ ;
}
}

```

4. Fonctions (5 points)

Ecrire un programme qui construit et affiche le triangle de Pascal en calculant les coefficients binomiaux:

```

1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
. . .

```

Les coefficients sont calculés d'après la formule ci-dessous :

$$C_p^q = \frac{p!}{q!(p-q)!}$$

Ainsi une ligne est composée des coefficients binomiaux $C_p^0 C_p^1 C_p^2 \dots C_p^p$.

- 1) Ecrire la fonction `int fact(int n)` qui renvoie la factorielle d'un entier `n`.
- 2) Ecrire la fonction `int C(int p, int q)` qui renvoie le calcul de la formule ci-dessus.
- 3) Ecrire la fonction `void ligne(int l)` qui affiche la ligne numéro `l` du triangle.
- 4) Ecrire la fonction `void triangle(int nbL)` qui affiche le triangle composé de `nbL` lignes.
- 5) Ecrire le programme principal qui demande à l'utilisateur le nombre de lignes voulues dans le triangle et affiche le triangle.

```

int fact (int n) {
    int f=1;
    while (n>0){
        f *= n;
        n-- ;
    }
    return f;
}

int C(int p, int q) {
    return fact(p)/(fact(q)*fact(p-q));
}

void ligne(int l) {
    int i ;
    for (i=0; i<=l-1; i++)
        printf("%d\t", C(l-1,i));
    printf("\n");
}

void triangle (int nbL) {
    int i;
    for (i=1; i<=nbL; i++)
        ligne(i);
}

void main() {
    int n;
    printf("Donnez le nombre de lignes SVP (>0) : ");
    scanf("%d", &n);
    triangle(n);
}

```