

## Idées de correction

**1. Nombres aléatoires (copie 1)**

On se propose d'étudier le générateur de nombres aléatoires Randu, qui permet de générer une suite de nombres aléatoires à partir de la formule suivante :

$$X_{n+1} = (65539 * X_n) \text{ modulo } 2^{31} \quad \text{avec } X_0 \text{ impair et } > 0$$

On veut étudier la périodicité de Randu à savoir, la possibilité de générer plusieurs fois le même nombre pour un  $X_0$  donné (appelé germe). Pour cela, nous allons générer 1 nombre de rang 1 ( $X_1$ ) et afficher le nombre de tirages nécessaires pour obtenir une seconde fois le même nombre. On répétera ce processus 3 fois en s'assurant que les  $X_0$  sont bien impairs et positifs.

Ecrire le programme C correspondant.

Le programme devra afficher :

- Entrez le germe : 45167
- Nombre aléatoire généré : 812716365
- Ce nombre a été généré une seconde fois au bout de : 87565458 tirages
  
- Entrez le germe : 65789
- Nombre aléatoire généré : 1111
- Ce nombre a été généré une seconde fois au bout de : 2164261623 tirages

Remarque : on supposera que le type int admet des entiers longs

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

void main()
{
    int x0; /* nombre saisi*/
    int x, nbAlea, i, nb; /* variables du pb*/

    for (i=0; i<3; i++) {

        do {
            printf("entrez un nombre impair:") ;
            scanf("%d", &x0) ;
        }
        while ( (x0%2) == 0);

        nbAlea = 65539 * x0 % pow(2. ,31) ;
        printf( "Nombre aleatoire genere : %d \n", nbAlea) ;
        x = nbAlea;
        nb = 0;

        do {
            x = 65539 * x % pow(2. ,31) ;
            nb++;
        }
        while (x!=nbAlea);

        printf("Ce nombre a été genere une seconde fois en %d tirages \n" ; nb) ;
    } /* fin de la boucle for */
} /* fin du main */
```

## 2. Boucles (copie 2)

1. Qu'affiche l'extrait de programme C suivant :

```
int i, j;
for( i = 1; i <= 5; i++){
for( j = 1; j <= (i - 1); j++)
printf("%d ", i - j);
printf("- \n");
}
```

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

2. Réécrire correctement cet extrait en respectant les règles d'indentation.

```
int i, j;
for( i = 1; i <= 5; i++){
    for( j = 1; j <= (i - 1); j++)
        printf("%d ", i - j);
    printf("- \n");
}
```

3. Compléter l'extrait pour que s'affiche à la fin de chaque ligne la somme (si celle-ci est non nulle) des nombres qui y figurent.

```
int i, j;
int somme;          /* calcul de la somme */
for( i = 1; i <= 5; i++){
    somme=0;
    for( j = 1; j <= (i - 1); j++) {
        printf("%d ", i - j);
        somme+= i - j;
    }
    if (somme!=0)
        printf("- %d \n", somme);
    else
        printf("- \n");
}
```

4. Modifier l'extrait pour que l'utilisateur affiche une par une chaque ligne en appuyant sur les touches « S » puis « entrée », sans pouvoir cependant afficher plus de 9 lignes à l'écran. L'appui sur une touche différente de « S » arrête l'affichage.

```
int i, j;
int somme;          /* calcul de la somme */
int cont = 0;      /* nb de lignes deja affichees*/
char rep = 'S';    /* pour continuer */

i = 1;
while(cont <= 9 && rep == 'S'){
    somme=0;
    for( j = 1; j <= (i - 1); j++) {
        printf("%d ", i - j);
        somme+= i - j;
    }
    if (somme!=0)
        printf("- %d \n", somme);
    else
        printf("- \n");

    fflush(stdin);
    scanf("%c", &rep);
    cont++;
    i++;
}
```

### 3. Questions diverses (copie 3)

1. Calculer les expressions suivantes :

54 % 4 + 10 / 3 vaut 5  
3 + 10 > 12 vaut 1  
3 + 10 > 20 || 3 \* 45 == 100 / 2 + 5 vaut 0

9 \* 5 == 2 + 3 \* 9 vaut 0  
28 - (5 == 2 + 3) vaut 27  
4 == 4 || 4 <= 4 && 4 < 4 vaut 1

2. Que vaut i à chaque étape ?

i = 10 ; i vaut 10  
i += 4\*5 ; i vaut 30  
i \*= i +=10 ; i vaut 1600

3. Qu'affiche le printf ?

```
int x, y ;  
int *ptx ;  
  
x = 10 ;  
y = 10 ;  
ptx = &x ;  
*ptx = 11 + x ;  
ptx = &y ;  
*ptx = 11 + x ;  
printf("x= %d, y= %d, ptx=%d", x, y, *ptx) ;
```

x = 21, y = 32, ptx = 32

4. Ecrire un programme qui calcule et affiche le salaire d'un employé payé à l'heure, à partir de son salaire horaire et du nombre d'heures de travail. Le taux horaire est augmenté de 20 % pour les heures au delà de 160 heures, de 35% pour les heures au delà de 180 heures et 50% pour les heures au delà de 200 heures.

```
#include <stdio.h>  
  
void main()  
{  
    float salH, nbH; /* données de type réel*/  
    float salaire; /* résultat : salaire mensuel*/  
  
    printf("Entrez le salaire horaire SVP") ;  
    scanf("%f", &salH) ;  
    printf("Entrez le nombre d'heures travaillees ce mois") ;  
    scanf("%f ", &nbH) ;  
  
    if (nbh <= 160)  
        salaire = nbH* salH;  
    else  
        if (nbh <= 180)  
            salaire = (160 * salH )+ (nbH-160) *salH *1.2;  
        else  
            if (nbh <= 200)  
                salaire = (160 * salH )+ 20 *salH *1.2 +(nbH-180)* salH * 1.35;  
            else  
                salaire = (160 * salH )+ 20*salH*1.2 + 20*salH*1.35 + (nbH-200)* salH * 1.5;  
    printf( "Salaire pour ce mois= %f \n", salaire) ;  
}
```

#### 4. Emprunt bancaire (copie 4)

Une personne désireuse de faire un emprunt pour l'achat d'une maison voudrait connaître les mensualités de remboursement. Il dispose pour cela des taux que proposent 2 banques différentes. Par ailleurs il a visité 25 maisons et il hésite entre un remboursement sur 15 ans ou sur 20 ans. Dans tous les cas, les mensualités ne pourront pas dépasser un certain plafond fixé par cette personne.

1. Ecrire un programme qui calcule à partir des taux bancaires et du prix de chaque maison, le montant des mensualités correspondant à 15 ou 20 ans de remboursement. On affichera le montant de la mensualité uniquement si celui-ci ne dépasse pas le plafond donné par l'utilisateur. Le programme affichera ensuite combien de maisons répondent à son critère.

La formule de calcul des mensualités d'un prêt est la suivante :

$$\text{mensualite} = \frac{\text{montantMaison} * \frac{\text{Taux}}{12} * \left(1 + \frac{\text{Taux}}{12}\right)^{\text{Nb\_mensualite}}}{\left(1 + \frac{\text{Taux}}{12}\right)^{\text{Nb\_mensualite}} - 1}$$

Remarque : on n'utilise pas de tableaux dans cet exercice.

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

void main()
{
    float taux1, taux2, prix, plafond ; /* données de type réel*/
    int nbmens1, nbmens2;              /* données de type entier */
    float mens;                        /* résultat du calcul de mensualité */
    int i;                              /* boucle */
    int M;                              /* indicateur pour une maison */
    int nbMais ;                       /* résultat : nombre de maisons convenant aux critères énoncés */

    printf("Entrez le taux de chaque banque SVP") ;
    scanf("%f %f", &taux1, &taux2) ;
    printf("Entrez le plafond") ;
    scanf("%f ", &plafond) ;
    nbmens1= 15*12;
    nbmens2 = 20*12;

    nbMais=0;
    for (i=1 ; i<= 25; i++) {
        printf("entrez le prix de la maison :)") ;
        scanf("%f", &prix) ;
        M=0;
        mens=(prix*taux1/12.*pow(1+taux1 /12. , nbmens1))/(pow(1 + taux1 /12. ,nbmens1) -1);
        if (mens < plafond) {
            printf( "Mensualité avec le taux %f sur %d mois = %f \n", taux1,nbmens1, mens) ;
            M++ ;
        }
        mens=(prix*taux2/12.*pow(1+taux2 /12. , nbmens1))/(pow(1 + taux2 /12. ,nbmens1) -1);
        if (mens < plafond) {
            printf( "Mensualité avec le taux %f sur %d mois = %f \n", taux2,nbmens1, mens) ;
            M++ ;
        }
        mens=(prix*taux1/12.*pow(1+taux1 /12. , nbmens2))/(pow(1 + taux1 /12. ,nbmens2) -1);
        if (mens < plafond) {
            printf( "Mensualité avec le taux %f sur %d mois = %f \n", taux1,nbmens2, mens) ;
            M++ ;
        }
        mens=(prix*taux2/12.*pow(1+taux2 /12. , nbmens2))/(pow(1 + taux2 /12. ,nbmens2) -1);
        if (mens < plafond) {
            printf( "Mensualité avec le taux %f sur %d mois = %f \n", taux2,nbmens2, mens) ;
            M++ ;
        }

        if ( M > 0)
            nbMais++;
    } /* fin du for */
    printf( " Il y  %d maisons répondant aux critères \n", nbMais) ;
}
```

2. On aurait pu définir et utiliser une fonction dans cet exercice. Décrivez en quelques mots : son rôle, les données dont elle a besoin et le résultat qu'elle produit.

On aurait pu écrire une fonction qui calcule la valeur des mensualités afin d'éviter d'écrire plusieurs fois la formule. Les données sont le taux de la banque, le prix de la maison et le nombre de mensualités. Le résultat est la valeur de la mensualité.