Corrigé Contrôle1 LgeC

Exercice 1 : (6pts)

Programme initial :

```
0 #include <stdio.h>
1 #define TAILLE_MAX 10

2 void Carre(int i, int c)
3 {
4  c = i*i;
5 }

6 int main()
7 {
    /* Déclarations e variables */

8    int Tab[TAILLE_MAX] = {0, 8, 0, 12, 2, 6, 0, 18, 0,
9};
9    int i, Nb, C;

/* Corps du pgme */

10   for (i = 1; i <= TAILLE_MAX; i++)
11      if (tab[i] = 0) printf("0 à la position %d \n", &i);
12   do
13    {
14      printf("Saisir un entier positif : \n");
15      scanf( "%c", Nb) ;
16    }
17   while nb <0;
18   Carre(Nb,C);
19   printf("Carré de % = %d\n", Nb,C);
20 }
```

Programme corrigé :

```
0 #include <stdio.h>
1 #define TAILLE_MAX 10

2 void Carre(int i, int * c)
3 {
4  *c = i*i;
5 }

6 int main()
7 {
    /* Déclarations de variables */

8    int Tab[TAILLE_MAX] = {0, 8, 0, 12, 2, 6, 0, 18, 0,
9};
9    int i, Nb, C;

/* Corps du pgme */

10   for (i = 0; i < TAILLE_MAX; i++)
11      if (Tab[i] == 0) printf("0 à la position %d \n", i);
12   do
13    {
14      printf("Saisir un entier positif : \n");
15      scanf( "%d", &Nb) ;
16    }
17   while (Nb <0);
18   Carre(Nb,&C);
19   printf("Carré de %d = %d\n", Nb,C);
20 }
```

Exercice2 : (7pts)

Voici 3 versions possibles de la fonction qui met les valeurs nulles d'une table à la fin de la table.

```
#define ECHANGE(a,b){int s = a; a = b; b = s;} //non obligatoire

void Permut(int *a, int *b)   // non obligatoire
{
  int s =*a; a = *b; *b = s;
}

void aufondZero1(int n, int *t)
{
  int i = 0, j= n-1;
  while(i <= j)
    if (t[i]==0)
      {
       if (t[j] != 0)
         {
           /* échange avec 3 instructions */
            int s = t[i] ;  t[i] = t[j] ; t[j] = s ;
            i++;
         }
       j--;
      }
    else i++;
}

void aufondZero2(int n, int t[])
{
  int i, j = n-1 ;
  for(i = 0; i< n; i++)
    if (t[i]==0)
     {
     while (t[j] == 0 && j>i) j--;
     /* échange avec une macro-définition */
     ECHANGE(t[i],t[j]);
     }
}

void aufondZero3(int n, int t[])
{
  int i,j;
  for(i = 0; i< n; i++)
     for (j=0;j<n-i-1;j++)
         if (t[j]==0 )
            /* échange avec une fonction void */
             Permut(&t[j],&t[j+1]);
}
```

Remarque : L'échange de 2 valeurs de la table peut se faire de 3 façons :

- utilisation d'une macro-définition (cf #define …..)
- utilisation d'une fonction sans retour (cf void …..)
- utilisation de 3 instructions : int s = t[i] ; t[i] = t[j] ; t[j] = s ;

Exercice3 : (7pts)

```
#include <stdio.h>

/* Fonction qui renvoie le kième chiffre, à partir de la droite, d'un nombre n donné */
/* Version 1 */
int kieme1 (int n, int k);
/* Version 2 */
int kieme2 (int n, int k);

int main()      //
{
  int nb, pos;
  char rep;
  do
  {
    printf("Saisir un nombre :");
    scanf("%d", &nb);
    printf("Saisir une position :");
    scanf("%d", &pos);
    /* Test avec la version 1 */
    printf("Le chiffre n° %d de %d est : %d\n", pos, nb, kieme1(nb, pos));
    /* Test avec la version 2 */
    printf("Le chiffre n° %d de %d est : %d\n", pos, nb, kieme2(nb, pos));

    printf("Encore?o/n: ");getchar();
    scanf("%c",&rep);
  }
  while(rep=='o' );
}


int kieme1(int n, int k)
{
  while (k!=1)
  {
    k-=1;  n/=10;
  }
  return n%10;
}

int kieme2(int n, int k)
{
  for ( int i = 1; i<k; i++, n/=10 );
// ou : for ( int i = 1; i <= k-1; i++, n/=10 );
//ou : for ( int i = 0; i< k-1; i++, n/=10 );
  return n%10;
}
```