Langage C : TD3

Exercice 1 : Écrire une fonction qui saisit n valeurs entières à stocker dans un tableau t pouvant contenir au maximum 100 valeurs, puis affiche ces valeurs dans l'ordre inverse de la saisie.

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

void exercice1(int n)
{
  int t[100];
  int a;
  srand(time(NULL));

  for(int i=0 ; i<n && i<100 ; i++)
  {
    a = rand();
    t[i] = a;
  }

  for(int i=n-1 ; i>=0 ; i--)
  {
    printf("%d\t", t[i]);
  }
}

int main()
{
  exercice1(5);
  return 0;
}
```

Exercice 2 : Écrire une fonction qui insère une valeur entière donnée v à sa place dans un tableau t contenant n éléments ordonnés dans l'ordre croissant, et de capacité maximale 100 .

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

void tri_insertion(int t[], int n)
{
  int x, j;
  for(int i = 0 ; i < n ; i++)
  {
    x = t[i];
    j = i;
    while(j > 0 & t[j-1] > x)
    {
      t[j] = t[j-1];
      j = j-1;
    }
    t[j] = x;
  }
}
```

```c
void exercice2(int n, int v)
{
  int t[100], a;
  srand(time(NULL));
  for(int i = 0 ; i < n-1 && i < 100 ; i++)
  {
    a = rand();
    t[i] = a;
  }
  t[n-1] = v;
  tri_insertion(t,n);
  for(int i = 0 ; i < n && i < 100 ; i++)
  {
    printf("%d\t", t[i]);
  }
}

int main()
{
  exercice2(10, 8789);
  return 0;
}
```

Exercice 3 : Écrire une fonction qui calcule et retourne le nombre de valeurs dupliquées dans un tableau t contenant n valeurs entières. La fonction retourne par exemple 2 si $t = [4, 2, 7, 7, 4]$ et 5 si $t = [4, 4, 2, 4, 4, 2, 2]$.

```c
#include <stdio.h>

int exercice3(int t[], int n)
{
  int compteur = 0;

  for(int i = 0 ; i < n ; i++)
  {
    for(int j = i-1 ; j >= 0 ; j--)
    {
      if(t[j]==t[i])
      {
        compteur++;
        break;
      }
    }
  }

  return compteur;
}

int main()
{
  int t1[5]={4, 2, 7, 7, 4};
  printf("%d", exercice3(t1, 5));
  int t2[7]={4, 4, 2, 4, 4, 2, 2};
  printf("\n%d", exercice3(t2, 7));

  return 0;
}
```

Exercice 4 : Écrire une fonction qui applique une rotation des n valeurs d'un tableau t pouvant contenir au plus 100 éléments, de m crans vers la gauche. Par exemple si $t = [1, 2, 3, 4, 5]$ $(n = 5)$ et $m = 4$ le tableau devient $[5, 1, 2, 3, 4]$.

```c
#include <stdio.h>

void exercice4(int t[], int n, int m)
{
  int reserve;

  for(int i = 0 ; i < m ; i++)
  {
    reserve = t[0];
    for(int j = 0 ; j < n-1 ; j++)
    {
      t[j] = t[j+1];
    }
    t[n-1] = reserve;
  }
}

int main()
{
  int t[5]={1, 2, 3, 4, 5}, n = 5;

  exercice4(t, n, 4);

  for(int i = 0 ; i < n ; i++)
  {
    printf("\t%d", t[i]);
  }

  return 0;
}
```

Exercice 5 : On considère un tableau à deux dimensions d'entiers représentant une matrice carrée et de taille 5×5.

1. Écrire une fonction qui initialise un tel tableau avec les valeurs $[1, 4, 9, 16, 25]$ sur la diagonale principale.

```c
#include <stdio.h>

void initialisation1(int t[5][5])
{
  for(int i = 0 ; i < 5 ; i++)
  {
    t[i][i] = (i+1)*(i+1);
  }
}

int main()
{
  int t[5][5]={{0, 0, 0, 0, 0},{0, 0, 0, 0, 0},{0, 0, 0, 0, 0},{0, 0, 0, 0, 0},{0, 0, 0, 0, 0}};

  initialisation1(t);

  for(int i = 0 ; i < 5 ; i++)
  {
    for(int j = 0 ; j < 5 ; j++)
    {
```

```
      printf("%d\t", t[i][j]);
    }
    printf("\n");
  }

  return 0;
}
```

2. Écrire une autre fonction qui initialise un tel tableau avec les valeurs $[5, 10, 20, 40, 80]$ sur la diagonale allant d'en haut à droite jusqu'en bas à gauche.

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

void initialisation2(int t[5][5])
{
  for(int i = 0 ; i < 5 ; i++)
  {
    t[i][4-i] = 5*pow(2,i);
  }
}

int main()
{
  int t[5][5]={{0, 0, 0, 0, 0},{0, 0, 0, 0, 0},{0, 0, 0, 0, 0},{0, 0, 0, 0, 0},{0, 0, 0, 0, 0}};

  initialisation2(t);

  for(int i = 0 ; i < 5 ; i++)
  {
    for(int j = 0 ; j < 5 ; j++)
    {
      printf("%d\t", t[i][j]);
    }
    printf("\n");
  }

  return 0;
}
```

3. Écrire une fonction qui calcule et affiche la somme des éléments au dessus et celle des éléments en dessous de la diagonale principale.

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

void initialisation2(int t[5][5])
{
  for(int i = 0 ; i < 5 ; i++)
  {
    t[i][4-i] = 5*pow(2,i);
  }
}

void exercice5(int t[5][5])
{
  int sommeH = 0, sommeB = 0;

  for(int i = 0 ; i < 5 ; i++)
  {
    for(int j = 0 ; j < 5 ; j++)
    {
      if(j>i) sommeH += t[i][j];
      if(j<i) sommeB += t[i][j];
```

```c
    }
  }
  printf("\nAu dessus de la diagonale : %d", sommeH);
  printf("\nEn dessous de la diagonale : %d", sommeB);
}

int main()
{
  int t[5][5]={{0, 0, 0, 0, 0},{0, 0, 0, 0, 0},{0, 0, 0, 0, 0},{0, 0, 0, 0, 0},{0, 0, 0, 0, 0}};

  initialisation2(t);
  exercice5(t);

  return 0;
}
```