

exercice 1 : (TP)

```
1- #include <stdio.h>
main()
{ float tab[100], max, min, somme;
  int i;
  printf("Entrez la valeur de tab[0] \n");
  scanf("%f", &tab[0]);
  max = min = somme = tab[0];
  for(i=1; i<100; i++) {
    printf("Entrez la valeur de tab[%d] \n", i);
    scanf("%f", &tab[i]);
    if(tab[i] > max) max = tab[i];
    if(tab[i] < min) min = tab[i];
    somme += tab[i];
  }
  printf("le max est %f \n", max);
  printf("le min est %f \n", min);
  printf("la moyenne est %f \n", somme/100);
}
```

```
2- #include <stdio.h>
main()
{ float max, min, somme, nb; int i;
  printf("Entrez la valeur num 1 \n"); scanf("%f", &nb);
  max = min = somme = nb;
  for(i=2; i<=100; i++) {
    printf("Entrez la valeur num %d \n", i); scanf("%f", &nb);
    if(nb > max) max = nb; if(nb < min) min = nb; somme += nb;
  }
  printf("le max est %f, le min est %f, la moyenne est %f \n", max, min, somme/100);
}
```

3- non, on peut pas car, ils seront écrasés !

Programmation Série 3 TD

exercice 2

```
#include <stdio.h>
main()
{
    int tab[50], n, i, temp;
    do {
        printf("Entrez la taille du tableau n tq n <= 50 \n");
        scanf("%d", &n);
    } while (n > 50 || n <= 0);
    for (i = 0; i < n; i++) {
        printf("Entrez la valeur T[%d] \n", i);
        scanf("%d", &T[i]);
    }
    for (i = 0; i <  $\frac{n}{2}$ ; i++) {
        temp = T[i];
        T[i] = T[n - i - 1];
        T[n - i - 1] = temp;
    }
}
```

autre méthode:

```
i = 0; j = n - 1;
while (i < j) {
    temp = T[i];
    T[i] = T[j];
    T[j] = temp;
    i++;
    j--;
}
```

exercice 3:

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

```
main()
```

```
{ int A[10] = {5, 6, 7, 10, 14, 24, 50, 82, 150};
```

```
  int VAL, temp1, temp2, i, j;
```

```
  printf("Entrez la valeur de VAL \n");
```

```
  scanf("%d", &VAL);
```

```
  for(i=0; i < 10; i++) {
```

```
    if (VAL <= T[i]) {
```

```
      temp1 = T[i];
```

```
      A[i] = VAL;
```

```
      for (j=i+1; j < 10; j++) {
```

```
        temp2 = A[j];
```

```
        A[j] = temp1;
```

```
        temp1 = temp2;
```

```
      }
    }
    break; }
  }
  if (i == 10) A[9] = VAL;
}
```

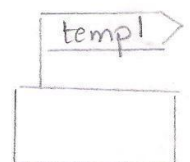
au lieu de temp2 je n'ai qu'à

utilisé VAL ! VAL = T[j];

T[j] = temp1;

temp1 = VAL;

5	6	7	10	14	24	50	82	150	
---	---	---	----	----	----	----	----	-----	--



deuxième méthode:

```
for (i=0; i < 9; i++)
```

```
  if (VAL <= A[i]) break;
```

```
  if (i != 9) { for (j=9; j > i; j--) A[j] = A[j-1];
```

```
    A[i] = VAL; }
```

```
else A[i] = VAL;
```

exercice 4:

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

```
main ( )
```

```
{ int T[50], N, i, j;
```

```
do {
```

```
printf("Entrez la taille du tableau N inférieur ou égal à 50:\n");
```

```
scanf("%d", &N); } while (N > 50 || N <= 0);
```

```
for (i=0; i < N; i++) {
```

```
printf("Entrez la valeur T[%d]\n", i);
```

```
scanf("%d", &T[i]);
```

```
}
```

```
for (i=0; i < N; i++)
```

```
printf("T[%d] = %d\n", i, T[i]);
```

```
for (i=0; i < N; i++) {
```

```
if (T[i] == 0) {
```

```
j = i;
```

```
do { if (T[j+1] != 0) break;
```

```
j++; } while (j < N);
```

```
T[i] = T[j+1];
```

```
T[j+1] = 0;
```

```
}
```

```
}
```

```
for (i=0; i < N; i++)
```

```
printf("T[%d] = %d\n", i, T[i]);
```

```
}
```

autre méthode :

```
k = 0;
```

```
for (i=0; i < N; i++)
```

```
if (T[i] != 0) { T[k] = T[i]; k++; }
```

```
for (i=0; i < k; i++) affichage du tableau
```

```
for (i=0; i < N; i++)
```

```
if (T[i] == 0)
```

```
{ for (j=i; j < N-1; j++)
```

```
T[j] = T[j+1];
```

```
i--;
```

```
N--;
```

```
}
```

exercice 5 :

tri d'un tableau par insertion

ecire un programme c qui trie par ordre croissant un tab d'entiers en utilisant la méthode du tri par insertion. le programme affichera le tableau ainsi trié.

```
#include <stdio.h>
main()
{
    int T[50], n, i, min, tmp, j;
    do { printf("Entrez la taille n du tableau (max n = 50): ");
        scanf("%d", &n); } while (n > 50);
    for (i = 0; i < n; i++) {
        printf("Entrez T[%d]: ", i); scanf("%d", &T[i]);
    }
    for (i = 0; i < n - 1; i++)
    {
        min = i;
        for (j = i + 1; j < n; j++)
            if (T[j] < T[min]) min = j;
        if (min != i) { tmp = T[i];
                       T[i] = T[min];
                       T[min] = tmp; }
    }
    for (i = 0; i < n; i++)
        printf("T[%d] = %d \n", i, T[i]);
}
```

autre méthode :

```
for (i = 1; i < m; i++)
```

```
for (j = i; j < 0; j--) {
```

```
if (T[j] < T[j-1]) { x = T[j];  
                    T[j] = T[j-1];  
                    T[j-1] = x; }
```

```
else break; }
```

exercice 6:

```
#include <stdio.h>
main ()
{ float A[20][30], B[20][30], temp;
  int N, M, i, j;
  //saisi des dimensions de la matrice A
  do {
    printf("Entrez le nb de lignes N et de colonnes M de la mat A (max N=20 et max M=30):");
    scanf("%d%d", &N, &M); } while (N > 20 || M > 30);
  //saisi de la matrice A
  for (i=0; i < N; i++)
    for (j=0; j < M; j++)
    { printf("Entrez la valeur de A[%d][%d] : ", i, j);
      scanf("%f", &A[i][j]);
    }
  //affichage de la matrice
  printf("La matrice donnée est : \n");
  for (i=0; i < N; i++)
    { for (j=0; j < M; j++)
      printf("%f\t", A[i][j]);
      printf("\n");
    }
  //calcul de la transposé
  if (N == M) { // A est une matrice carré, la transposé sera mémorisé dans A
    for (i=0; i < N; i++)
      for (j=0; j < i; j++)
      { temp = A[i][j];
        A[i][j] = A[j][i];
        A[j][i] = temp; }
  } else { // affectation de la matrice transposé à B
    for (i=0; i < N; i++)
      for (j=0; j < M; j++)
        B[j][i] = A[i][j]; }
```

// affichage du résultat

printf("La transposé de la matrice A est : \n");

for (i=0; i < M; i++)

{ for (j=0; j < N; j++)

{ if (N == M) printf("%f\t", A[i][j]);

else printf("%f\t", B[i][j]); }

printf("\n");

}

}