
Fiche TP N° 02 « Programmation en langage C sous Linux »

Remarques:

- Les commandes Linux doivent être écrites en lettres minuscules
- **exit** pour quitter le shell actuel
- **id** pour afficher le numéro de l'utilisateur courant ainsi que les groupes auxquels ce dernier appartient.
- Pendant cette séance de TP, on va utiliser l'éditeur de texte **gedit** pour écrire nos programmes (vous pouvez utiliser d'autres éditeurs tels que : vi, emacs et nano,...). Pour nano les raccourcis les plus importants sont affichés en bas de sa fenêtre.
 - Compiler le programme avec le compilateur **gcc** du projet GNU (pour les programmes C et C++) en tapant la commande : **gcc -o nom_du_programme nom_du_programme.c**
 - Exécuter le programme par : **./nom_du_programme**

Exercice 1:

- 1) Compiler et exécuter le programme suivant:

```
# include <stdio.h>
main()
{
int NOMBRE, SOMME, COMPTEUR; SOMME=0;COMPTEUR=0;
While (COMPTEUR<4)
    {Printf ("Entrer un nombre entier:");
    Scanf (%i, &NOMBRE); SOMME+=NOMBRE;COMPTEUR++;}
    printf ("la somme est: %i \n", SOMME);
    Return 0;
}
```

- 2) Qu'est ce qu'il fait ce programme:

Exercice 2:

- 1) Corriger le programme ci-dessous et faire son exécution.

```
# include <stdio.h>
int main(void)
{
    int i, som, nbm
    double moy
    int t[20]
    for (i=0; i<20, i++);
    {
    Printf ("donnez la note numéro %d:"; i+1)
    Scan ("%d", &t[i]);
    }
    for(i=0, som=0; i<20; i++) som += t[i];
    moy = som / 20
    print ("\n\n moyenne de la classe : %f\n", moy);
    for (i=0, nbm=0; i<20, i++ )
    if (t[i] > moy ) nbm++
    printf (%d élèves ont plus de cette moyenne, nbm);
    return 0
}
```

- 2) Qu'elle la fonction principale de ce programme.

Exercice 3:

Soit les deux entiers a et b tel que a=-1972 et b=2017

- 1) Ecrire un programme en C qui permet de calculer et afficher : $a-b$, $a+b$, $a*b$, a/b et $a\%b$ en format décimal.
- 2) Que va-t-il se passer lors de l'exécution du programme précédent si a et b sont des variables réels (afficher le résultat).

Exercice 4:

- 1) Compiler et exécuter le programme suivant:

```
# include <stdio.h>
main()
{
  int T[20];
  int N, I;
  long SOM;
  printf ("donner la dimension du tableau T:");
  do
  scanf ("%d", &N);
  while ((N<0) || (N>20));
  for (I=0; I<N; I++)
  {
  printf ("Elément %d",I);
  scanf ("%d", &T[I]);
  }
  printf ("Tableau donné:\n");
  for (I=0;I<n;I++)
  printf ("%d", &T[I]);
  printf ("\n");
  for (SOM=0;I=0;I<N;i++)
  SOM+=T[I];
  Printf ("la somme d'éléments est: %ld\n,SOM);
}
```

Exercice 5:

- 1) Compiler et exécuter le programme suivant:

```
# include <stdio.h>
main()
int A[10][20], B[20][30];
int N,M;/*dimensions des matrices*/
int I,J;/*indices courants*/
/*saisie des données*/
printf ("Nombre de lignes de A:");
scanf ("%d",&N);
printf ("Nombre de colonnes de A:");
scanf ("%d",&M);
for (I=0,i<N;I++)
for (J=0;J<M;J++)
{printf ("Element [%d][%d]:", I,J);
Scanf("%d",&A[i][j]);}
/*affichage des matrices*/
printf("Matrice donnée A:\n");
for (I=0,i<N;I++)
for (J=0;J<M;J++)
printf ("%7d", A[i][j]);
printf ("\n");
/*affectation de la matrice transposée à B*/
for (I=0,i<N;I++)
for (J=0;J<M;J++)
B[J][I]=A[I][J];
/*Edition des resultants*/
Printf("Matrice transposée de A:\n");
for (I=0,i<M;I++)
{for (J=0;J<N;J++)
printf ("%7d", B[I][J]);
printf("\n");}
return 0;
}
```

- 2) Exploiter les résultats de ce programme en modifiant les paramètres internes et déduire d'autres applications si possible.