

TP n° 2

I3 : Appels de fonctions, récursivité**Exercice 1** [Passage de paramètres]

1. Considérons le programme C suivant :

```
1      #include <stdio.h>
2      #include <stdlib.h>
3
4      void inverser( int a, int b ) {
5          int tmp;
6          tmp = a;
7          a = b;
8          b = tmp;
9      }
10
11     int main( ) {
12
13         int val1, val2;
14
15         printf( "Entrez le premier entier :\n" );
16         scanf( "%d", &val1 );
17         printf( "Entrez le deuxieme entier :\n" );
18         scanf( "%d", &val2 );
19
20         inverser( val1, val2 );
21
22         printf( "Valeurs apres inversion : val1 = %d, val2 = %d\n",
23             val1, val2 );
24
25         return EXIT_SUCCESS;
26     }
```

Quelles seront les valeurs affichées ? Le programme a-t-il le comportement désiré ?

2. Que faut-il modifier pour obtenir le bon résultat ?

Exercice 2 [Portée des variables]

Qu'affiche le programme C suivant ? Expliquez en détail votre réponse.

```
1      #include <stdio.h>
2      #include <stdlib.h>
3
4      void affiche1( int x ) {
5          x = x + 1;
6          printf( "affiche1 : x = %d\n", x );
7      }
8
9      int affiche2( int x ) {
```

```
10     x = x + 1;
11     printf( "affiche1 : x = %d\n", x );
12     return x;
13 }
14
15 int main(){
16     int x;
17     x = 10;
18     printf( "main : x = %d\n", x );
19     affiche1( x );
20     printf( "main : x = %d\n", x );
21     x = affiche2( x );
22     printf( "main : x = %d\n", x );
23     return EXIT_SUCCESS;
24 }
```

Exercice 3 [Calcul récursif de factorielle]

On veut calculer la valeur de $n!$.

1. Quelle est la formule permettant de calculer $n!$?
2. Quelle est la relation entre $n!$ et $!(n - 1)$?
3. Écrire une fonction C qui calcule récursivement la valeur de $n!$.

Exercice 4 [Calculatrice en ligne de commande]

Écrivez un programme C dont les arguments constituent une expression arithmétique simple (addition, soustraction, multiplication, division) de la forme : `./calcul operande1 operateur operande2`. Les opérandes sont des nombres entiers et l'opérateur est un caractère.

Par exemple, `./calcul 12 + 24` va afficher `12 + 24 = 36`.

Exercice 5 [Algorithme d'Euclide]

Vous avez vu en TD l'algorithme d'Euclide. Implémentez en C l'algorithme d'Euclide afin de donner le plus grand commun diviseur de deux entiers passés en arguments du programme. Vous écrirez deux versions de ce programme : la première utilisant une boucle, la deuxième utilisant une fonction récursive.

Vous pourrez utiliser la fonction `floor()` (à regarder dans l'aide en ligne). Vous devrez utiliser la bibliothèque `libm` fournissant des fonctions mathématiques (dont la fonction `floor()`) : ajoutez l'option `-lm` à la fin de votre ligne de compilation.