

Programmation I : TD4 (Pointeurs et Fonctions)

Exercice 1 : Ecrivez un programme qui range les éléments d'un tableau T de type float dans l'ordre inverse en utilisant uniquement le formalisme pointeur.

Exercice 2 : Ecrivez un programme qui lit un tableau A du type int au clavier et élimine toutes les occurrences de la valeur 0 dans A en tassant les éléments restants. Le programme utilisera deux pointeurs P1 et P2 pour parcourir le tableau.

Exercice 3 : Ecrire un programme qui effectue la transposition tA d'une matrice A de dimensions N et M. La matrice transposée sera mémorisée dans une deuxième matrice B qui sera affichée. Utiliser le formalisme pointeur et l'allocation dynamique de la mémoire à chaque fois que cela est possible.

Exercice 4 : Donnez la séquence d'affichages produits par le programme suivant :

```
#include <stdio.h>
int fct ( int );
void f ( void );
int n = 10, q = 2 ;
main()
{ int n = 0, p = 5;
  n = fct ( p ) ;
  printf(" A : dans main, n = %d, p = %d, q = %d\n ", n , p , q ) ;
  f() ;
}
int fct ( int p )
{ int q ;
  q = 2 * p + n ;
  printf(" B : dans fct, n = %d, p = %d, q = %d\n ", n , p , q ) ;
  return q ;
}
void f ( void )
{ int p = q * n ;
  printf(" C : dans f, n = %d, p = %d, q = %d\n ", n , p , q ) ;
}
```

Exercice 5 : Qu'affiche le programme suivant à l'écran ?

```
#include<stdio.h>
int mystere(int a,int *b)
{ a=*b*2;
  *b=*b+a;
  return (*b+1);
}
```

```
int main(){
  int x=1,y=1,z=1;
  z=mystere(x,&y);
  printf("x=%d,y=%d,z=%d\n",x,y,z);
  x=mystere(x,&y);
  printf("x=%d,y=%d,z=%d\n",x,y,z);
}
```

Exercice 6 : Le pgcd de deux entiers positifs est le plus grand entier positif qui divise les deux.

Quand on étudie les propriétés du pgcd on voit que :

$\text{pgcd}(x, y) = \text{pgcd}(x - y, y)$ si $x > y$

$\text{pgcd}(x, y) = \text{pgcd}(x, y - x)$ si $x < y$

$\text{pgcd}(x, 0) = \text{pgcd}(0, x) = x$

1. Ecrire une fonction récursive pour le calcul du pgcd.
2. Ecrire un programme C qui fait appel à la fonction pgcd.