

Travaux dirigés Numéro 2

Dans ce TD, nous abordons le problème du passage de paramètres dans les fonctions, et la notion de pointeurs. Un certain nombre d'exercices permettant de mettre oeuvre ces notions vous sont proposés ici.

1 A programmer

1.1 Echange du contenu de deux variables

1. Ecrire une fonction `Permute` qui échange le contenu de deux variables entières.
2. Ecrire la fonction principale qui initialise deux variables entières et qui permute le contenu de ces deux variables en utilisant la fonction `Permute`. Afficher ensuite le contenu de ces deux variables.

1.2 Allocation dynamique de mémoire

Soit le programme suivant :

```
#define TAILLE 10

int main(void)
{
    char * chaine;
    int * tab;
    int i;
```

```
for (i=0;i<TAILLE-1;i++) {
    tab[i]=i+1;
    chaine[i]=(char)(i+65);
}
chaine[TAILLE-1]=0;
return 0;
}
```

Que va donner l'exécution de ce programme? Modifier le programme (uniquement en ajoutant des lignes) de telle façon que son exécution soit correcte.

1.3 Paramètres de la fonction main

Il est possible en C de passer des paramètres à la fonction `main` (point d'entrée de tout programme C). Dans ce cas, l'en-tête de la fonction `main` devient : `int main(int argc, char * argv[])` ou `argc` est le nombre de paramètres et `argv` un tableau de chaînes comprenant les paramètres (le nom du programme exécutable est considéré comme le premier paramètre).

Ecrire un programme C qui accepte 3 paramètres (et seulement 3, la fonction doit manifester son désaccord si ce nombre n'est pas 3) et qui se contente d'afficher ces 3 paramètres.

2 A cogiter

2.1 Fonction et passage de paramètres

Quelles sont les valeurs des entiers `*a`, `b` et `c` à la fin de l'exécution du programme principal? Commentez.

```
#include <stdio.h>
int Pipo( int i, int * j, int k)
{
    *j = 4;
    i = 5;
    k +=10;
    return k;
}
```

```
int main(void)
{
    int * a;
    int b=3;
    int c;

    a = &b;
    b = 2;

    c = Pipo(b, a, *a);
    return 0;
}
```

2.2 Classes de mémorisation

Soit le programme suivant :

```
#include <stdio.h>

static int var1=0;
int var2=3;
int var3=6;

void f(int var2)
{
    static int i=0;
    int j=9;
    int var1;

    i++;
    j--;
    var1=5;
    var2=6;
    var3=8;
}

int main(void)
```

```
{  
    int var2;  
  
    var2=5;  
    f(var2);  
    f(var1);  
    f(var3);  
    return 0;  
}
```

Donnez les valeurs de var2, var1 et var3 en fin de programme ainsi que la valeur de i et j à la fin de chaque appel de la fonction f.