

Exercices de C supplémentaires

Passage par adresse

1) Faire une fonction pour échanger la valeur de deux nombres avec le prototype suivant :

```
void swap(int *n1, int *n2);
```

2) Faire une fonction pour trier trois nombres avec le prototype suivant :

```
void triatris(int* a, int* b, int* c);
```

Les valeurs de *a*, *b* et *c* doivent vérifier après l'appel : $a \leq b \leq c$

3) Transactions Bancaires

Faire une fonction avec le prototype suivant :

```
int virement(int* compte_orig, int* compte_dest, int somme)
```

Cette fonction doit renvoyer 1 si et seulement si le versement de la *somme* de *compte_orig* à *compte_dest* a été effectué (c'est-à-dire si le versement ne rend pas le solde de *compte_orig* négatif) et renvoie 0 sinon. On peut déclarer les comptes comme variables globales de la façon suivante puis utiliser la procédure *etat_comptes* pour montrer leur solde :

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

```
void etat_comptes(int compte1, int compte2, int compte3, int compte4) {  
    printf("Etat des comptes : \n");  
    printf("Compte n 1 : %d\n",compte1);  
    printf("Compte n 2 : %d\n",compte2);  
    printf("Compte n 3 : %d\n",compte3);  
    printf("Compte n 4 : %d\n",compte4);  
}
```

```
int virement(int* compte_orig, int* compte_dest, int somme){  
    // Completer  
}
```

```
int main(){  
    int compte1 = 1000;  
    int compte2 = 2000;  
    int compte3 = 1500;  
    int compte4 = 3000;  
  
    etat_comptes(compte1,compte2,compte3,compte4);  
  
    // Completer  
  
    etat_comptes(compte1,compte2,compte3,compte4);  
  
    return 0;  
}
```

4. Expliquer (et surtout comprendre) le résultat de :

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

```
void fonction (int x, int **y);
```

```
void fonction2 ();
```

```
int a;
```

```
int main ()
```

```
{
```

```
    int a, b;
```

```
    int *p;
```

```
    b = 8;
```

```
    p = &b;
```

```
    a = 32 + b;
```

```
    p = &a;
```

```
    *p = 32 - b;
```

```
    fonction (a, &p);
```

```
    *p = 2;
```

```
    printf ("a=%d b=%d \n", a, b);
```

```
    fonction2();
```

```
    return 0;
```

```
}
```

```
void fonction (int x, int **y)
```

```
{
```

```
    a = 15;
```

```
    **y = x - a;
```

```
    *y = &a;
```

```
}
```

```
void fonction2 () {
```

```
    printf ("a=%d \n", a);
```

```
}
```