

Feuille 3

- I. Démonstration de passage par valeur et passage par référence des fonctions. On écrit deux fonctions « swap » et « echange » dont le but est d'échanger les valeurs de deux variables xval et yval. Écrivez et compilez le programme suivant :

```
#include <cstdlib>
#include <iostream>

using namespace std; // Nécessaire pour une bonne compilation C++

void swap(float ,float ); // déclaration de la fonction « swap » - passage par valeur
void echange(float& ,float& ); // déclaration de « echange » – passage par référence

main()
{ float xval=2.5,yval=10; // déclaration des variables réels « xval » et « yval »

printf("\nDémonstration de passage par valeur\n");
cout <<"Avant l'appel de la fonction swap";//cout vient de la librairie C++ <iostream>
cout <<" xval=" << xval << " yval=" << yval << endl;
swap(xval,yval);
cout << "Après l'appel de la fonction swap";
cout << " xval=" << xval << " yval=" << yval << endl;

printf("\nDémonstration de passage par référence\n");
echange(xval,yval);
cout << "Après l'appel de la fonction echange";
cout << " xval=" << xval << " yval=" << yval << endl;

system("PAUSE"); return EXIT_SUCCESS;
} // fin de la fonction main

// passage par valeur, l'échange des valeurs x et y n'a pas d'effet dans le "main"
void swap(float x,float y{ float tmp=x;
    x=y; y=tmp;
    cout << "A l'intérieur de la fonction swap";
    cout << " x=" << x << " y=" << y <<endl;
}

// passage par référence, l'échange des valeurs x et y s'effectue aussi dans "main"
void echange(float& x,float& y)
{ float tmp=x;
    x=y; y=tmp;
}
```

Expliquer le comportement de ce programme.

- II. Avec les « **structures** », vous pouvez créer vos propres types de variables. Ici on va créer une structure « **Personne** » qui contient le nom d'une personne, son adresse, et son age. On écrit une fonction « **init** » qui va demander l'utilisateur d'entrer les informations sur la personne, et une fonction « **affiche** » qui va afficher ses informations :

```
#include <cstdlib>
#include <iostream>

using namespace std;

struct Personne { // Nouvelle structure Personne
    char nom[30];
    char adresse[40];
    int age;
};

void init(Personne& p){ // Initialiser les infos sur la personne
    printf("\nEntrez le nom : "); gets(p.nom);
    printf("\nEntrez l'adresse : "); gets(p.adresse);
    printf("\nEntrez l'age : "); scanf("%d",&p.age); getchar();
}

void affiche(const Personne& p){ // Afficher les infos sur la personne
    cout << "\n" << p.nom << " age de " << p.age ;
    cout << " ans habite " << p.adresse << endl;
}

main()
{ Personne employ1,employ2;

    init(employ1); init(employ2);
    affiche(employ2); affiche(employ1);

    cout << "\n" << employ1.nom << " a " << employ1.age;
    cout << " ans et " << employ2.nom;
    cout << " a " << employ2.age << " ans \n" << endl;

    system("PAUSE");
    return EXIT_SUCCESS;
}
```

Pourquoi nous n'avons pas besoin de déclarer les fonctions « **init** » et « **affiche** » dans cet exercice ?