

TP2 :Pointeurs

1 Exercice 1

Testez le code suivant :

```
1 #include <iostream>
2 using namespace std;
3
4 int main()
5 {
6     int mon_entier(23);
7     //Affichage de l'adresse de la variable en utilisant &
8     //Elle sera donnée en base 16 (d'où la présence de lettres)
9     cout << "L'adresse est : " << &mon_entier << endl;
10    return 0;
11 }
```

Rajouter les lignes suivantes à votre code :

```
1 //Ce code déclare un pointeur qui peut contenir
2 //l'adresse d'une variable de type int.
3 int *pointeurInt;
4 //On peut faire de même pour n'importe quel type :
5 double const *pointeurDoubleConst;
6 vector<int> *pointeurVector; //etc ...
```

Pour le moment, les pointeurs ne contiennent aucune adresse connue, on ne sait donc pas quelle case de la mémoire est manipulée quand on utilise le pointeur. Pour éviter les bug, il est préférable de donner au pointeur la valeur 0. Cela signifie qu'il ne contient l'adresse d'aucune case. Modifiez le code de la manière suivante :

```
1 //Ce code déclare un pointeur qui peut contenir
2 //l'adresse d'une variable de type int.
3 int *pointeurInt(0);
4 //On peut faire de même pour n'importe quel type :
5 double const *pointeurDoubleConst(0);
6 vector<int> *pointeurVector(0); //etc ...
```

Testez le code suivant et décrivez les instructions qui y sont effectuées :

```
1 int mon_entier(23); //Une variable de type int
2 int *ptr(0); //Un pointeur pouvant contenir l'adresse d'un int
3 ptr = &mon_entier; //L'adresse de 'mon_entier' est mise dans le pointeur 'ptr'
4 //On dit alors que le pointeur ptr pointe sur mon_entier
5 cout << "L'adresse de 'mon_entier' est : " << &mon_entier << endl;
6 cout << "La valeur de 'ptr' est : " << ptr << endl;
7 cout << "La valeur est : " << *ptr << endl;
```

2 Exercice 2 : allocation dynamique

Rajoutez à votre code les lignes suivantes et testez les :

```
1 int *pointeur(0); //Définit un pointeur pouvant contenir l'adresse d'un int
2 pointeur = new int; //Demande 1case et renvoie un pointeur pointant vers celle-ci
3 *pointeur = 2; //Accède à la case mémoire pour en modifier la valeur
4 delete pointeur; //Libère la case mémoire
5 //Attention le pointeur pointe toujours mais vers une case vide !!!
6 pointeur = 0; //Indique que le pointeur ne pointe plus vers rien
```

3 Exercice 3

Ecrivez un programme qui demande l'âge de l'utilisateur, et l'affiche à l'aide d'un pointeur. Le pointeur doit être l'unique variable utilisée.

4 Exercice 4 : QCM

Rajoutez à votre code les lignes suivantes et testez les :

```
1 #include <iostream>
2 #include <string>
3 using namespace std;
4
5 int main()
6 {
7     string repA("Euler"), repB("Runge-Kutta");
8     cout << "Quel schéma en temps souhaitez-vous utiliser ? " << endl;
9     cout << "A) " << repA << endl;
10    cout << "B) " << repB << endl;
11
12    char votre_reponse;
13    cout << "Votre reponse (A ou B) : ";
14    cin >> votre_reponse; //Récupère la réponse de l'utilisateur
15
16    string *reponseUtilisateur(0); //Un pointeur qui pointera sur la réponse
17
18    switch(votre_reponse)
19    {
20    case 'A':
21        reponseUtilisateur = &repA; //Déplace le pointeur sur la réponse choisie
22        break;
23
24    case 'B':
25        reponseUtilisateur = &repB;
26        break;
27    default:
28        cout << "Ce choix n'est pas valable !" << endl;
29        exit(0); //Le programme s'arrête
30    }
31    //Utilise le pointeur pour afficher la réponse choisie
32    cout << "Vous avez choisi la reponse : " << *reponseUtilisateur << endl;
33    //Le pointeur n'ayant pas été défini par un "new"
34    //il n'est pas nécessaire de libérer la case mémoire
35    return 0;
36 }
```

Que se passe-t-il si on retire la ligne 28 et que l'on ne répond ni par A ni par B ?

Créez votre propre QCM.