



Les Pointeurs

Il s'agit dans ce TP d'apprendre à manipuler les pointeurs

Exercice 1 :

Q1) Compléter les instructions (pointillés noirs)

Q2) Exécuter le programme en mode pas à pas détaillé en observant les **contenus des variables** (à l'aide de la fenêtre "Watches") **et les affichages** sur la console. Compléter les pointillés verts.

```
#include <iostream>
using namespace std;

int main()
{
    char tabc[8]={'b','o','n','j','o','u','r','\0'};
    int tabn[5]={515,78,435,7,89};
    char *ptrc;
    int *ptrn;

    cout<<"Affichage de l'adresse du 1er élément de tabn : "<<.....<<endl;
    //Affectation de l'adresse du tableau tabn dans ptrn
    .....
    cout<<"Affichage de l'adresse contenue dans ptrn:"<<.....<<endl;
    cout<<"Affichage de l'élément pointé par ptrn:"<<.....<<endl;
    // Valeur de ptrn (ou adresse contenue dans ptrn) : .....
    // Valeur pointée par ptrn : .....

    ptrn++;
    cout<<"Affichage de l'adresse contenue dans ptrn:"<<.....<<endl;
    cout<<"Affichage de l'élément pointé par ptrn:"<<.....<<endl;
    // Valeur de ptrn (ou adresse contenue dans ptrn) : .....
    // Valeur pointée par ptrn : .....
    ptrn+=3;
    cout<<"Affichage de l'adresse contenue dans ptrn:"<<.....<<endl;
    cout<<"Affichage de l'élément pointé par ptrn:"<<.....<<endl;
    // Valeur de ptrn (ou adresse contenue dans ptrn) : .....
    // Valeur pointée par ptrn : .....

    cout<<"Affichage de l'adresse du 1er élément de tabc : "<<(void*)tabc<<endl;
    // Il est nécessaire de convertir l'adresse du 1er élément du tableau en
    // adresse générique pour pouvoir l'afficher sur la console
    //Affectation de l'adresse du tableau tabc dans ptrc
    .....
    cout<<"Affichage de l'élément pointé par ptrc:"<<.....<<endl;
    // Valeur de ptrc (ou adresse contenue dans ptrc) : .....
    // Valeur pointée par ptrc : .....

    ptrc++;
    cout<<"Affichage de l'élément pointé par ptrc:"<<.....<<endl;
    // Valeur de ptrc (ou adresse contenue dans ptrc) : .....
    // Valeur pointée par ptrc : .....

    ptrc+=2;
    cout<<"Affichage de l'élément pointé par ptrc:"<<.....<<endl;
    // Valeur de ptrc (ou adresse contenue dans ptrc) : .....
    // Valeur pointée par ptrc : .....
```

```
    return 0;
}
```

De combien est augmenté le contenu de *ptrc* lorsqu'on l'incrémente ? Justifier.

Même question pour *ptrn* ? Justifier.

Justifier la valeur prise par *ptrc* après l'exécution de l'instruction `ptrc+=2;`

Justifier la valeur prise par *ptrn* après l'exécution de l'instruction `ptrn+=3;`

Q3) Déclarer, définir et tester une fonction qui permet l'affichage d'un tableau d'entiers passé en paramètre sans utiliser l'opérateur [].

Exercice 2 :

Q1) Compléter le programme suivant afin de transférer le contenu de la variable *m* dans la variable *p* et d'afficher la variable *p*, en n'utilisant que les pointeurs *ptr1* et *ptr2*.

```
#include <iostream>
using namespace std;

int main()
{
    int m=60,p=80;
    int *ptr1, *ptr2;
    ...
    return 0;
}
```

Exercice 3 :

Suite de l'exercice 5 du TP sur les fonctions.

Ajouter une fonction nommée *getMaxEtMin()* permettant de "**récupérer**" dans la fonction **appelante** (c'est à dire la fonction *main()*) **les valeurs** maximale et minimale du tableau. Cette fonction ne devra pas réaliser d'affichage (pas de saisie non plus).

Les valeurs maximale et minimale obtenues devront donc être affichées dans la fonction *main()*. Il faudra bien entendu proposer un choix supplémentaire dans le menu.