

## Objectif

Le but de ce TP est d'assimiler les notions suivantes :

- Les spécificités de C++
- Notions du pointeur et référence
- Passage par valeur, par référence et par adresse
- L'allocation dynamique
- Les bibliothèques <string> et vector<

## Exercice 1

Question 1 : Donnez les valeurs de A,B,C, P1 et P2 :

```
int A =1, B=2, C=3 ;
int *P1, *P2 ;
P1=&A;
P2=&C;
*P1=(*P2)++;
P1=P2;
P2=&B;
*P1-=*P2;
++*P2;
*P1*=*P2;
A=++*P2**P1;
P1=&A;
*P2=*P1/=*P2;
```

Question 2 : Un pointeur et une référence

- ☐ Déclarez un tableau d'entiers A de dimension 10
- ☐ Développez deux méthodes, remplitA et afficherA, permettant de remplir et d'afficher le tableau.
- ☐ Déclarez une référence vers l'élément 5 du tableau et incrémenter sa valeur
- ☐ Déclarez un pointeur vers l'élément 7, incrémenter le pointeur et la valeur pointée

Question 3 : Passage de paramètre par valeur vs Passage de paramètre par référence vs Passage de paramètre par pointeur

Voici trois implémentations de la méthode square :

```
int square(int n) {
    cout << "In square(): " << &n << endl; // 0x22ff00
    n *= n;
    return n;
}

void square(int * pNumber) {
    cout << "In square(): " << pNumber << endl; // 0x22ff1c
    *pNumber *= *pNumber;
}

void square(int & rNumber) {
    cout << "In square(): " << &rNumber << endl; // 0x22ff1c
    rNumber *= rNumber
}
```

Implémentez la méthode main, où vous appelez la méthode square. Appelez la méthode cout pour afficher la valeur de l'entier passé en paramètre. Quand est-ce que la valeur du nombre passé en paramètre change ?

## Exercice 2

Trouvez les erreurs dans les suites d'instructions suivantes et proposez les corrections :

a) int * p; int x=34; *p=x;	b) int x=17; int * p=x; *p=17 ;	c) double * q; int x=17; int * p=&x; q=p ;	d) int x; int * p; &x=p;	e) char mot[10]; char car='A'; char * pc=&car; mot=pc;
--------------------------------------	--	--	-----------------------------------	--

## Exercice 3

Ecrire plus simplement en C++, les instructions suivantes, en utilisant les opérateurs *new* et *delete* :

```
int* adi ;
double * add ;
.....
adi = (int*) malloc (sizeof (int)) ;
add = (double*) malloc (sizeof (double) * 100);
```

## Exercice 4

Expliquez pourquoi les deux sous-programmes sont incorrectes.

Que faut-il modifier pour obtenir une version correcte ?

int* mauvais1 (int n) { int tab[n]; for (int i=0 ; i<n ; i++) cin >> tab[i] return tab; }	Const int MAX = 50 ; int* mauvais2 () { int tab[MAX]; for (int i=0 ; i<MAX ; i++) cin >> tab[i] return tab; }
---	--

## Exercice 5

Ecrire un programme qui déclare un tableau dynamique de chaîne de caractères et qui, après avoir demandé le nombre de chaînes à stocker, en effectue la saisie.

Vous allez implémenter ces deux cas :

- ☐ Le tableau est créé avec un *new*. Les chaînes de caractères sont de type *char\**.
- ☐ Utiliser la classe *vector* et la classe *string*