

Exercice 1:

I- On appellera *dimension* d'une pile le nombre maximal d'éléments qu'elle peut contenir et *taille* d'une pile le nombre d'éléments qu'elle contient réellement. Le tableau représentant la pile est donc indexé de 0 (bas de la pile) à *taille-1* (haut de la pile). L'élément que l'on peut dépiler est donc dans la case d'indice *taille-1*.

Ecrire le fichier Pile.cpp correspondant au fichier Pile.h suivant :

```
#ifndef Pile H
#define Pile H
class Pile
{ public :
    Pile(int t = 10); // Constructeur qui construit une pile de dimension t (10 par défaut)
    Pile(const Pile &); // Constructeur par copie
    ~Pile(); // Destructeur
    void empile(int); // empile n en haut de la pile
    void depile(); // depile le sommet de la pile
    int donnetaille() const; // renvoie la taille de la pile
    Pile & operator=(const Pile &); // surcharge de l'opérateur d'affectation
    bool operator==(const Pile &); // surcharge de l'opérateur ==

private :
    int dim;
    int taille;
    int *adr;
};
#endif
```

II- Remplacer la fonction membre « empile » par l'opérateur < et la fonction membre « depile » par l'opérateur --.

p < n ajoute la valeur n sur la pile p

--p supprime la valeur du haut de la pile p.

III- Ajouter à la classe Pile la surdéfinition de l'opérateur [], de sorte que la notation p[i] ait un sens et retourne l'élément d'emplacement i de la pile p.

Utiliser ce nouvel opérateur pour écrire les fonctions affiche et saisie

On créera donc une fonction membre de prototype int &Pile::operator[](int i);

IV- En utilisant les fonctions amies, donner une signification aux opérations :

p+x et x+p où p désigne une pile et x un entier.(la somme d'une pile et un entier est obtenue en ajoutant à chaque élément de la pile l'entier x).

V- Ajouter à la classe pile la surcharge des opérateurs << et >>.

Exercice 2:

Définir une classe chaîne permettant de créer et de manipuler une chaîne de caractères:

données:

- adresse d'une zone allouée dynamiquement (inutile d'y ranger la constante \0)

méthodes:

- constructeur chaîne() initialise une chaîne vide
- constructeur chaîne(char *texte) initialise avec la chaîne passée en argument
- constructeur par recopie chaîne(const chaîne &ch)
- opérateurs affectation (=), comparaison (==), concaténation (+), accès à un caractère de rang donné ([]), << et >>.