



Université Abdelmalek Essaadi
Faculté des Sciences et Techniques de Tanger
Département Génie Informatique
IDAI-54



DEVOIR A RENDU :

Réaliser par :

Boulidat Yassine

Encadrée par :

Prof: IKRAM BEN ABDEL OUAHAB

```

#include <iostream>
using namespace std;
class Stack
{
private:
    int* Array;
    int top;
    int size;

public:
    // Constructeur avec une valeur par défaut de 20
    Stack(int s = 20) : size(s), top(-1)
    {
        Array = new int[size];
    }

    // Destructeur
    ~Stack()
    {
        delete[] Array;
    }

    // Surcharge de l'opérateur
    Stack& operator<<(int n)
    {
        if (top < size - 1)
        {
            Array[++top] = n;
        }
        return *this;
    }

    // Surcharge de l'opérateur
    Stack& operator>>(int& n)
    {
        if (top >= 0)
        {
            n = Array[top--];
        }
        return *this;
    }

    // Surcharge de l'opérateur ++
    operator int()
    {

```

```

        return (top == size - 1) ? 1 : 0;
    }

    // Surcharge de l'opérateur --
    operator int() const
    {
        return (top == -1) ? 1 : 0;
    }

    // Interdiction de l'affectation entre piles
    Stack& operator=(const Stack&) = delete;
};

int main()
{
    Stack p(5); // Exemple d'utilisation avec une pile de taille 5

    p << 10 << 20 << 30; // Ajout de valeurs à la pile

    int n1, n2, n3;
    p >> n1 >> n2 >> n3; // Extraction des valeurs de la pile

    cout << "Valeurs : " << n1 << ", " << n2 << ", " << n3 << endl;

    // Exemples d'utilisation des opérateurs ++ et --
    cout << "La pile est pleine ? " << (p++ ? "Oui" : "Non") << endl;
    cout << "La pile est vide ? " << (p-- ? "Oui" : "Non") << endl;

    // Tentative d'affectation entre piles (interdite)
    Stack p2;
    // p2 = p; // Cela entraînera une erreur à la compilation

    return 0;
}

```