

## Module: POO C++

### Nom complete: YASSINE JARRADI

```
#include <iostream>
#include <vector>

class Stack {
private:
    std::vector<int> stack;
    int capacité;

    bool isFull() const {
        return stack.size() == capacité;
    }

    bool isEmpty() const {
        return stack.empty();
    }
public:
    // on a crée un constructeur nomée stack
    Stack(int capacité = 20) : capacité(capacité) {
        stack.reserve(capacité);
    }

    bool operator<<(int n) {
        if (isFull()) {
            return false;
            //stack est rempli
        }
        // cet cas stack n'est pas rempli donc en ajout l'entier n a la pile
        stack.push_back(n);
        return true;
    }

    bool operator>>(int& n) {
        if (isEmpty()) {
            return false;
            //stack n'est pas repli
        }
        // en utilise stack.back pour retirer le dernier element de la pile
        n = stack.back();
        stack.pop_back();
        return true;
    }

    bool operator++() const { // p++ vale 1 si la pile p est pleine et 0 dans le
cas contraire
        return isFull();
    }

    bool operator--() const { // p-- vale 1 si la pile p est vide et 0 dans le cas
contraire.
        return isEmpty();
    }

};
```

```

int main() {
    Stack p(8); // on a crée une pile avec la capacité de 8

    p << 1;
    p << 2;
    p << 3;
    p << 4;
    p << 5;
    p << 6;
    p << 7;
    p << 8;

    int n;
    p >> n;
    std::cout << "la valeur est : " << n << std::endl;

    p << 9; // la pile est plain dans le cas de 9

    if (++p) {
        std::cout << "Stack est rempli" << std::endl;
    }
    else {
        std::cout << "Stack n'est pas rempli" << std::endl;
    }

    if (--p) {
        std::cout << "Stack n'est pas rempli" << std::endl;
    }
    else {
        std::cout << "Stack n'est pas rempli" << std::endl;
    }

    return 0;
}

```