#include <iostream>

#define GARBAGE -99999

using namespace std;

class Stack

{

    int size;

    int top = -1;

    int \*arr;

public:

    Stack(int s)

    {

        size = s;

        arr = new int[s];

    }

    bool isFull()

    {

        return top == size - 1;

    }

    bool isEmpty()

    {

        return top == -1;

    }

    void push(int element)

    {

        if (isFull())

        {

            cout << "OVERFLOW" << endl;

            return;

        }

        top++;

        arr[top] = element;

    }

    int pop()

    {

        if (isEmpty())

            return GARBAGE;

        int element = arr[top];

        top--;

        return element;

    }

    void display()

    {

        if (isEmpty())

        {

            cout << "Empty Stack" << endl;

            return;

        }

        cout << arr[top] << " <--- TOP" << endl;

        for (int i = top - 1; i >= 0; i--)

            cout << arr[i] << endl;

    }

};

int main()

{

    int n, element, choice;

    cout << "Enter max size of stack: ";

    cin >> n;

    Stack s(n);

    while (true)

    {

        cout << "------Stack-MENU------" << endl;

        cout << "1.Push" << endl;

        cout << "2.Pop" << endl;

        cout << "3.Display" << endl;

        cout << "4.Exit" << endl;

        cout << "Enter your choice" << endl;

        cin >> choice;

        if (choice == 4)

        {

            cout << "Exiting program" << endl;

            break;

        }

        switch (choice)

        {

        case 1:

            cout << "Enter element: ";

            cin >> element;

            s.push(element);

            break;

        case 2:

            element = s.pop();

            if (element == GARBAGE)

                cout << "UNDERFLOW" << endl;

            else

                cout << "POPED: " << element << endl;

            break;

        case 3:

            s.display();

            break;

        default:

            cout << "Make a valid choice" << endl;

            break;

        }

    }

    return 0;

}

