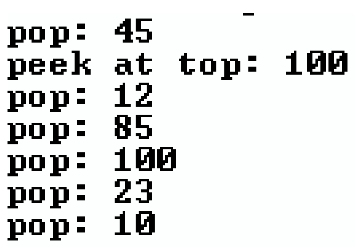
```cpp
#include <iostream>
using namespace std;

// implementing the Stack ADT using array


class Stack{

    private:
        int *array;
        int maxSize; // useful to decide if resizing (doubling the array size) is needed
        int topOfStack; // same as endOfArray

    public:
        Stack(int size){
            array = new int[size];
            maxSize = size;
            topOfStack = -1;
        }

        bool isEmpty(){

            if (topOfStack == -1)
                return true;

            return false;
        }

        void resize(int s){

            int *tempArray = array;

            array = new int[s];

            for (int index = 0; index < min(s, topOfStack+1); index++){
                array[index] = tempArray[index];
            }

            maxSize = s;

        }


        void push(int data){    // same as insert 'at the end'

            if (topOfStack == maxSize-1)
                resize(2*maxSize);

            array[++topOfStack] = data;

        }

        /*
        // Insert at Index is not allowed in Stack
        // We have to always insert at the end (top) of the Stack


        */

        int peek(){

            if (topOfStack >= 0)
                return array[topOfStack];
            else
```

```cpp
                return -1000000;// an invalid value indicating
                                // stack is empty

        }


        int pop(){

            if (topOfStack >= 0){
                return array[topOfStack--];
                // the topOfStack is decreased by one
            }
            else
                return -1000000; // an invalid value indicating
                                // stack is empty
        }





};

int main(){

    Stack stack(1);

    stack.push(10);
    stack.push(23);
    stack.push(100);
    stack.push(45);
    cout << "pop: " << stack.pop() << endl;
    cout << "peek at top: " << stack.peek() << endl;

    stack.push(85);
    stack.push(12);

    while (!stack.isEmpty())
        cout << "pop: " << stack.pop() << endl;


return 0;
}
```



```
pop: 45
peek at top: 100
pop: 12
pop: 85
pop: 100
pop: 23
pop: 10
```