پشته
Stacks

بهزاد منتظری
دانشکده فنی ومهندسی دانشگاه رازی
نیمسال دوم ۸۳-۸۲

Stack

An alternative storage structure for collections of entities is a stack.

A stack is a simplified form of a linked list in which all insertions and deletions occur at one end of the list, known as the top or head of the stack.

When an element is inserted at the top of a stack it is said to have been pushed onto the stack.

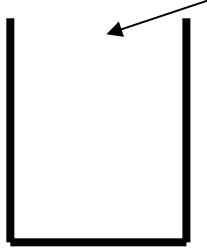
Stack

When an item has been deleted from the top of the stack it is said to have been popped off the stack.

When a pop occurs the item at the top of the stack is returned as the output from the pop operation.

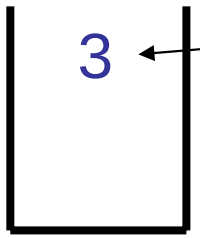
Stack

Initially the stack is empty



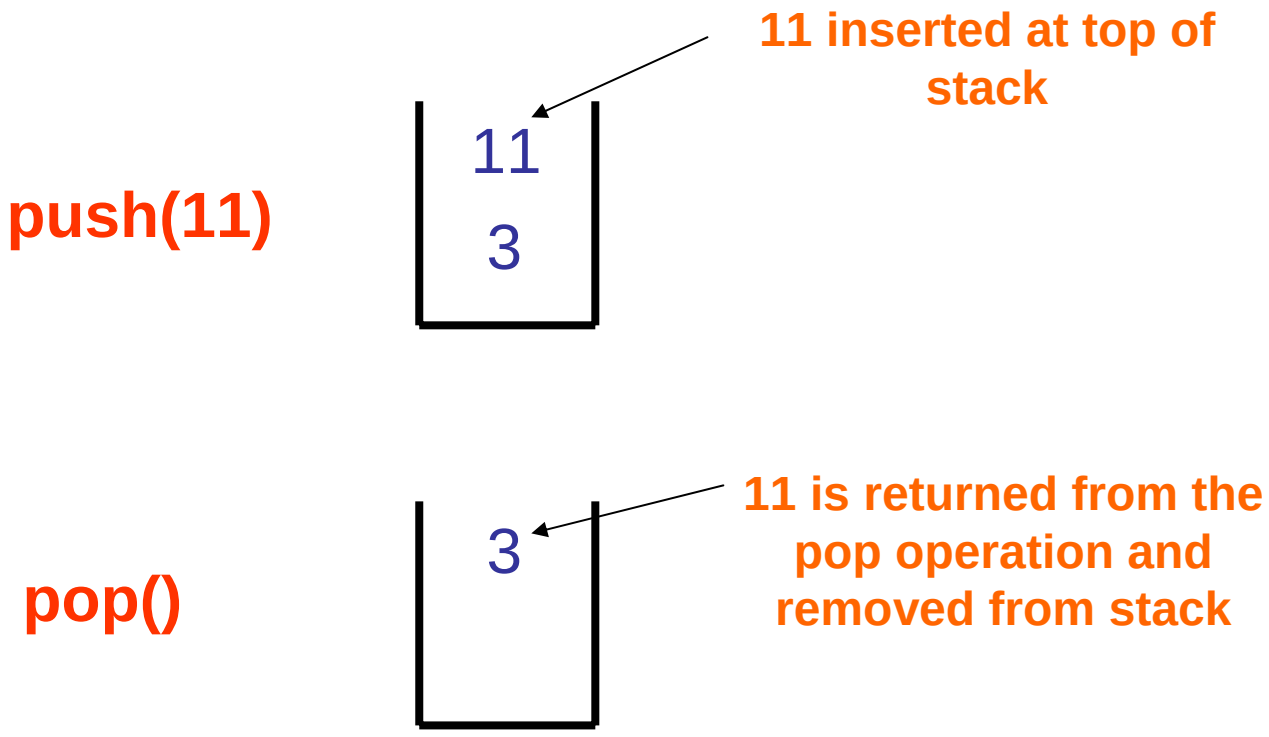
No items can be popped

push(3)



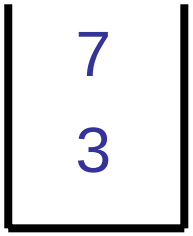
3 placed on top of stack

Stack

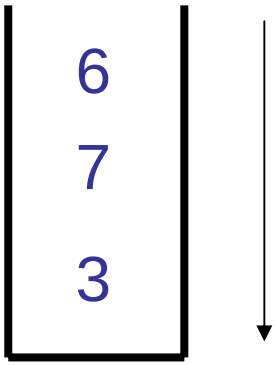

11 inserted at top of stack
push(11)
11
3
pop()
3
11 is returned from the pop operation and removed from stack

Stack

push(7)



push(6)



stack grows downwards

Stack

Since the last element pushed onto the stack is the first one to be popped , Stacks are known as Last-In-First-Out (LIFO) lists.

Stacks and Function Invocation

Stacks are very useful data structures in Computer Science. Their LIFO operation makes them suitable for certain types of problems, in particular function invocation.

تمرین: کلاس IntStack را پیاده سازی کنید.

```
#include <iostream.h>.
const int maxsize = 20;
class IntStack {
  int   element[ maxsize ];      // عناصر پشته
  int   topindex;                // اندیس بالاترین عنصر
public:
       IntStack() { topindex = -1; } // سازنده
  int   top();                   //  مقداربازگشتی بالاترین عنصر
  int   top( int * t );
                                 // بالاترین عنصر را در  t قرار میدهد
                                 // 0 = ok, -1 =error   مقدار بازگشتی
  int   pop();                   //  مقداربازگشتی بالاترین عنصروحذف آن از پشته
  int   pop( int * t );
  int   push( int e );           // اضافه کردن عنصر به پشته
                                 // 0 = ok, -1 =error   مقدار بازگشتی
  inline int   empty(){return topindex == - 1;}                //
 اگر پشته خالی باشد مقدار بازگشتی ۱- در غیر اینصورت ۰
  ostream& print( ostream& o );
};
ostream& operator<<( ostream& o, IntStack& s );
```