STACKS

The Stack Concepts

Linear List or Linear Array: any place deletion, insertion, reading possible

The Stack Abstract Data Type

Designed to store data on a LAST IN FIRST OUT basis

LIFO Data Structure

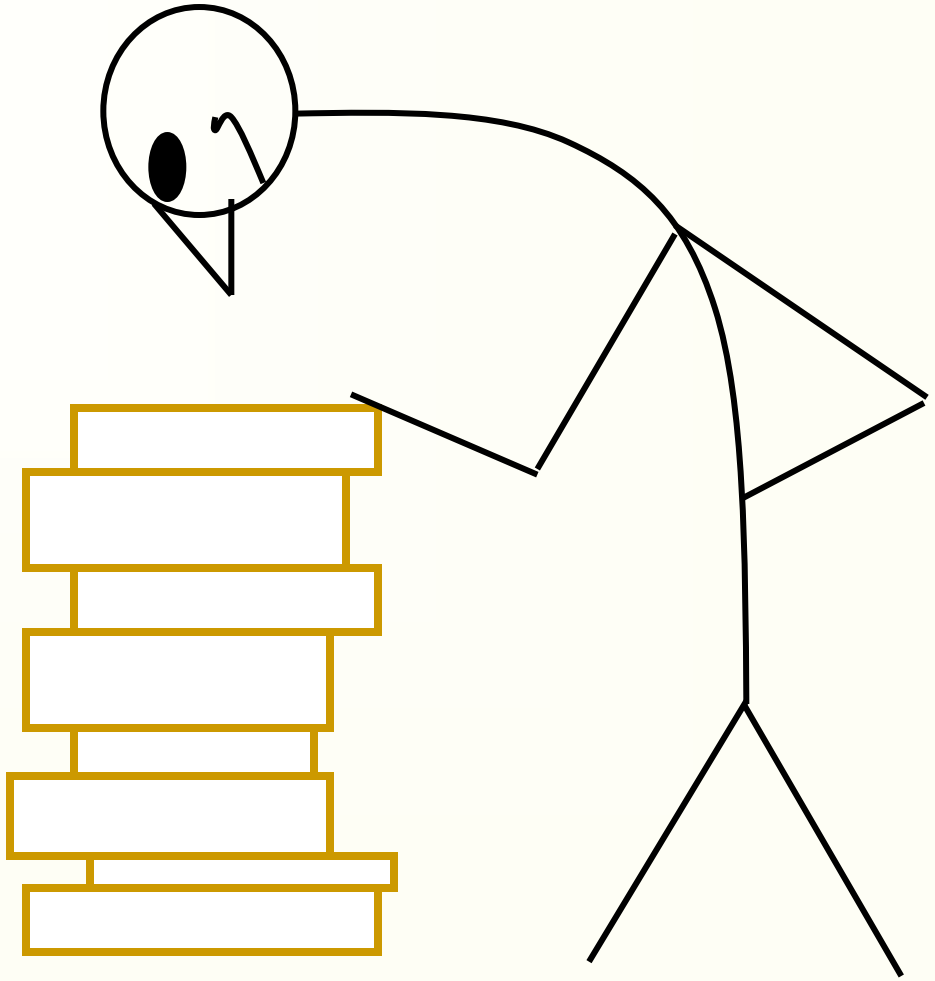
Functions

- push
- pop
- peek = pop + push
- initialize

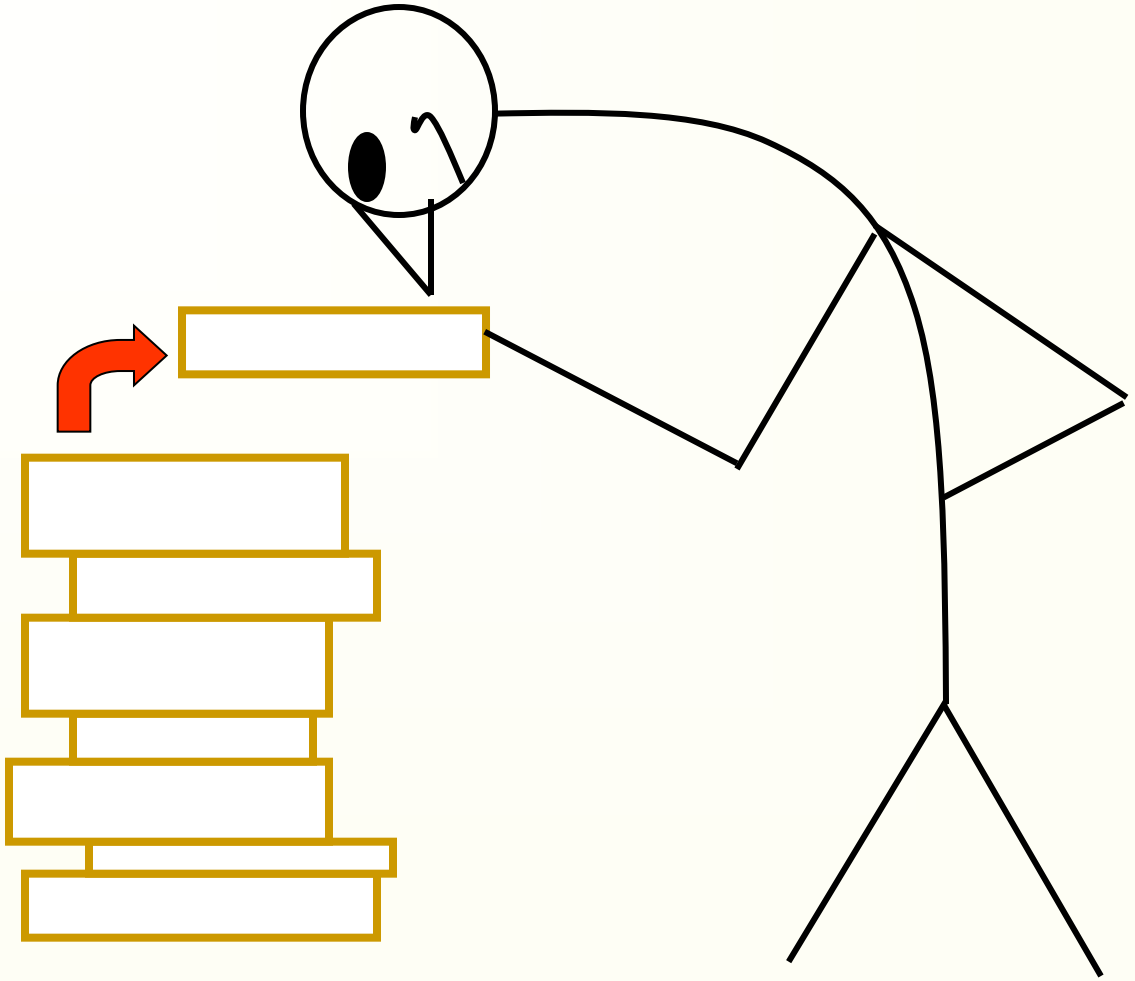
Stacks are also called PILES/ PUSH Down Lists

An Example – Stack of Books

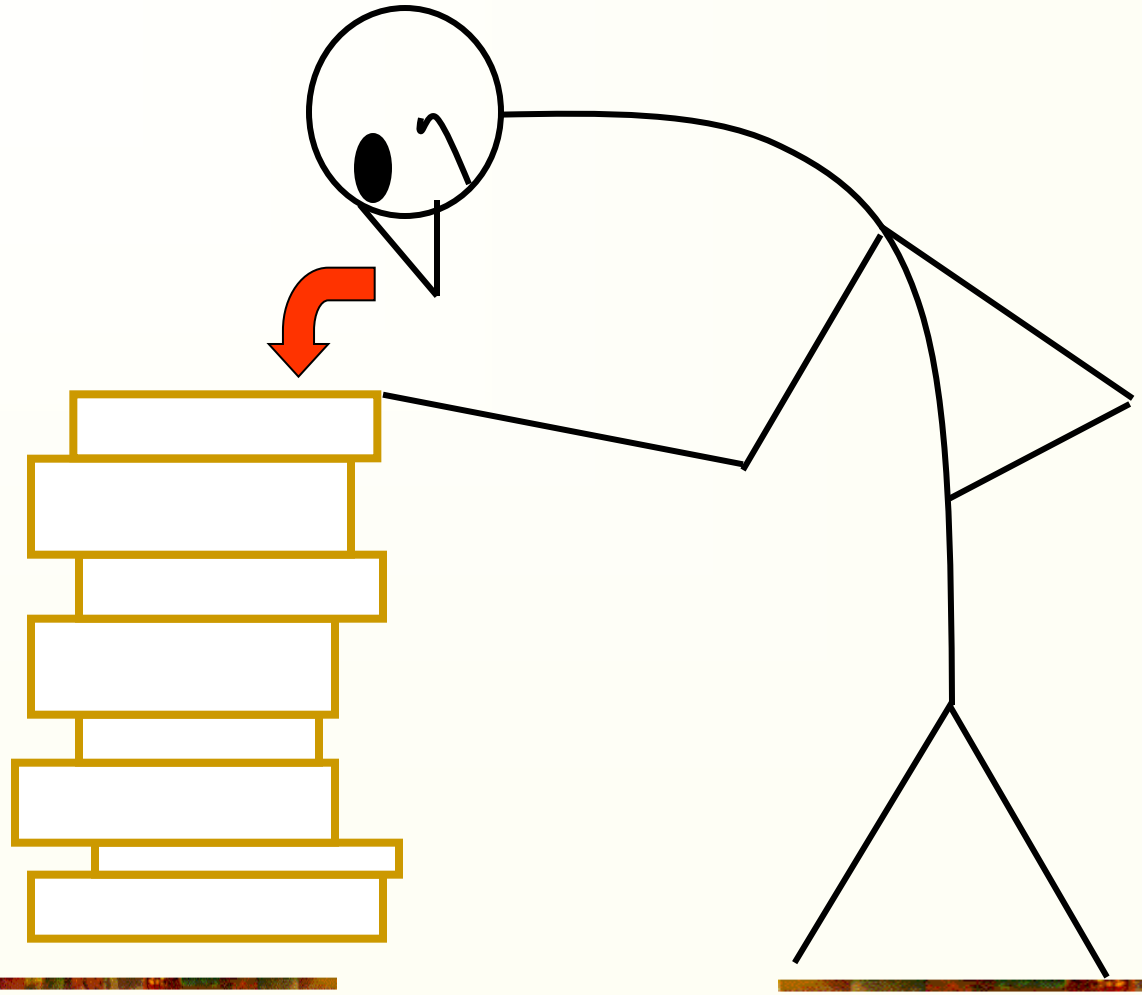
Question
How many books are there?



The pop operation – remove an item



The push operation – add an item



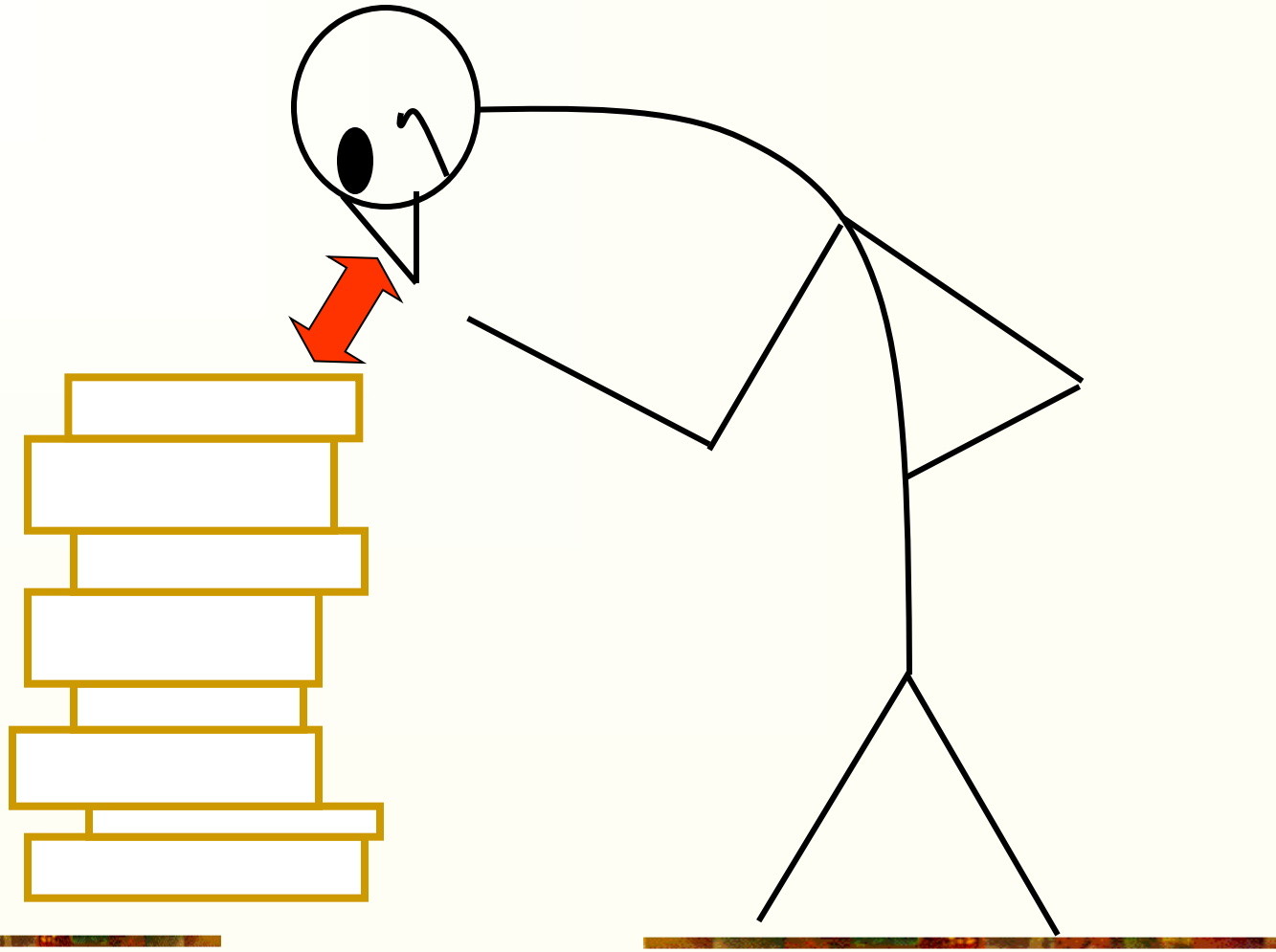
The peek operation – examine an item

same as

pop

followed by

push



The Complete ADT

push	inserts an ITEM
pop	removes an ITEM
peek	gives ITEM without removing
isEmpty	checks if stack is EMPTY
isFull	checks if stack is FULL
init	initializes stack

Review of stack1.c

```
/* stack declarations */
#define MAXSTACKSIZE 100

typedef double ITEMTYPE;

typedef struct
  {
   ITEMTYPE store[MAXSTACKSIZE];
   int top;
  } STACK;
```

General purpose design for any type of data to be stored in stack.

Review of stack1.c

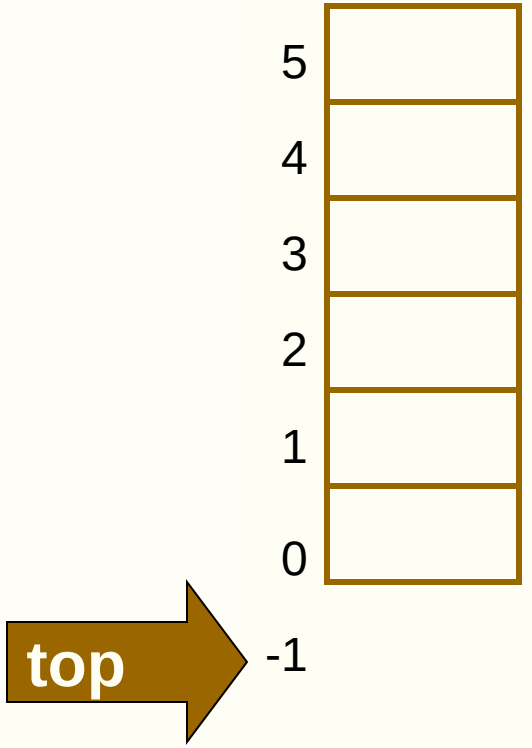

```
void initStack(STACK * sp);
void push(STACK * sp, ITEMTYPE x);
ITEMTYPE pop(STACK * sp);
ITEMTYPE peek(STACK * sp);
int isEmptyStack(STACK * sp);
int isFullStack(STACK * sp);
```

send only address to function

space economy – no duplicate of stack is created by the call-by-value mechanism of C language

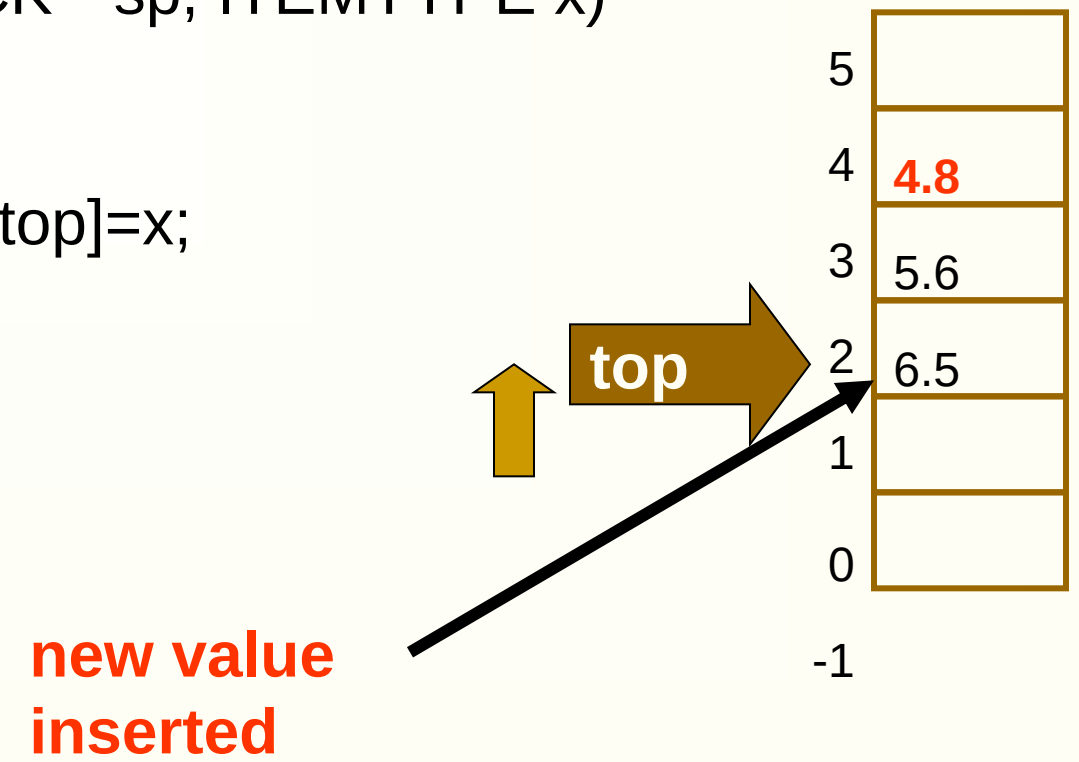
Review of stack1.c


```
void initStack(STACK * sp)
{
 sp->top=-1;  //(*sp).top
 return;
}
```



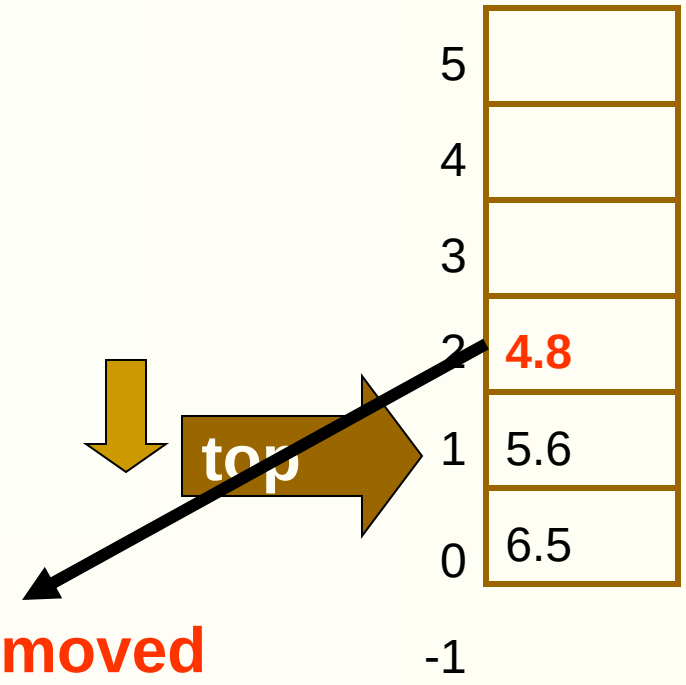
Review of stack1.c


```
void push(STACK * sp, ITEMTYPE x)
{
  sp->top++;
  sp->store[sp->top]=x;
  return;
}
```



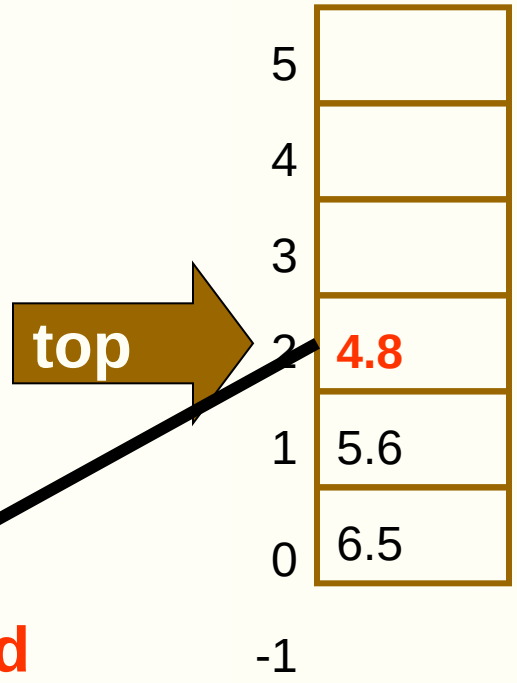
Review of stack1.c


```
ITEMTYPE pop(STACK * sp)
{
 return sp->store[sp->top--];
}
```



Review of stack1.c


```
ITEMTYPE peek(STACK * sp)
{
 return sp->store[sp->top];
}
```



top value returned
but not removed

Review of stack1.c


```
int isEmptyStack(STACK * sp)
{
 if(sp->top==-1)
   return 1;
 else
   return 0;
}

int isFullStack(STACK * sp)
{
 if(sp->top==MAXSTACKSIZE-1)
   return 1;
 else
   return 0;
}
```

Uses of Stack

- To reverse a string
- To check whether a string is palindrome or not
- To convert an infix expression to its postfix expression
- To evaluate one postfix expression

THANK YOU!