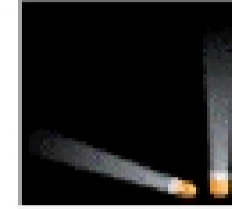
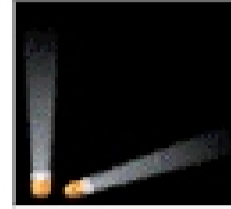
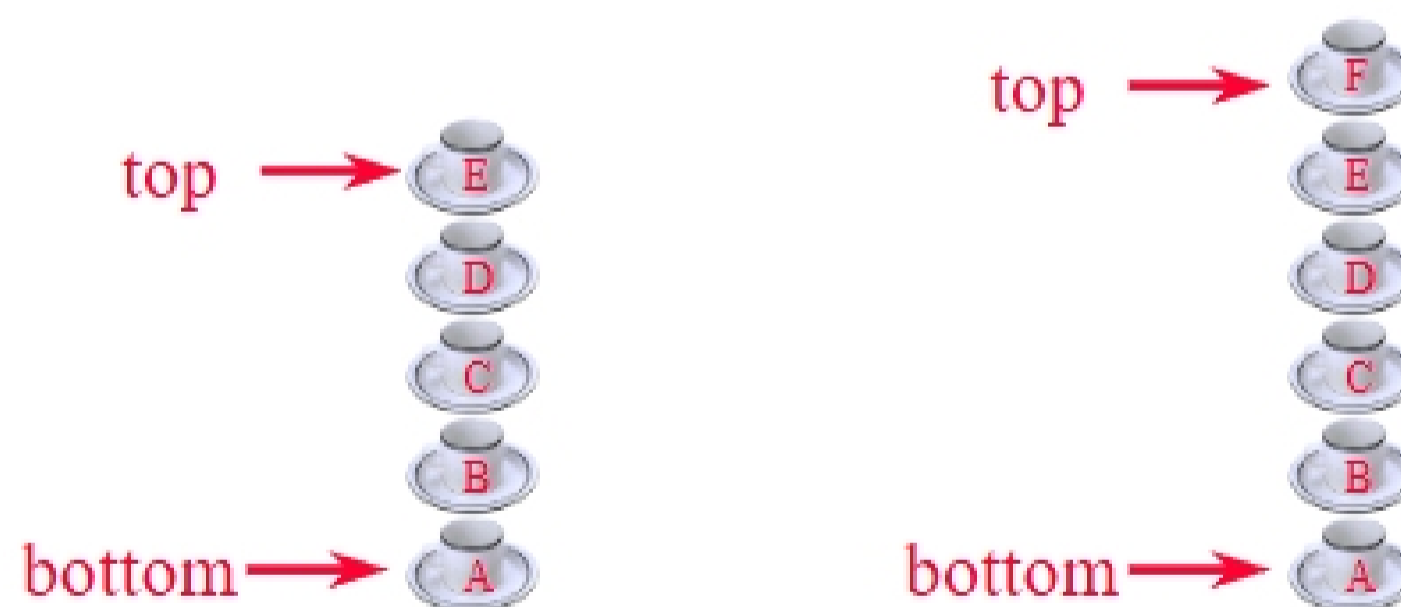


Stacks



- Linear list.
- One end is called **top**.
- Other end is called **bottom**.
- Additions to and removals from the **top** end only.

Stack Of Cups



- Add a cup to the stack.
- Remove a cup from new stack.
- A stack is a LIFO list.

The Interface Stack

```
public interface Stack
{
    public boolean empty();
    public Object peek();
    public void push(Object theObject);
    public Object pop();
}
```

Parentheses Matching

- $((a+b)*c+d-e)/(f+g)-(h+j)*(k-l))/(m-n)$
 - Output pairs (u,v) such that the left parenthesis at position u is matched with the right parenthesis at v .
 - $(2,6)$ $(1,13)$ $(15,19)$ $(21,25)$ $(27,31)$ $(0,32)$ $(34,38)$
- $(a+b)*((c+d)$
 - $(0,4)$
 - right parenthesis at 5 has no matching left parenthesis
 - $(8,12)$
 - left parenthesis at 7 has no matching right parenthesis

Parentheses Matching

- scan expression from left to right
- when a left parenthesis is encountered, add its position to the stack
- when a right parenthesis is encountered, remove matching position from stack

Example

- $((a+b)*c+d-e)/(f+g)-(h+j)*(k-l))/(m-n)$

