Stack II

Mr. Khaleifah Al.jada'
Data Structures
Al-Majma'ah Community College

Class Stack

```
Typedef int stack_entry;
Class stack
{
  public:
  stack();
  bool empty() const;
  bool pop();
```

Cont..

```
bool push(const Node_entry &item);
 bool top(Node_entry &item);
 ~stack();
 private:
 Node *top_node;
};
```

Constructor

```
Stack::stack()
{
  top_node  =  NULL;
}

// just to initial top_node pointer to NULL
```

Method Push

```
Bool stack::push(const Node_entry& item)
{
  Node * new_top = new Node(item,top_node);
  if(new_top == NULL)
      return false;
  top_node = new_top;
  return true;
}
```

Method Pop

```
bool stack:: pop()
{
  Node * old_top = top_node;
  if(top_node == NULL)
      return false;
  top_node = top_node→next;
  delete old_top;
Return true;
}
```

Method Top

```
bool stack::top(Node_entry &item)const
{
  if(empty())
      return false;
  item = top_node→data;
  return true;
}
```

Method empty

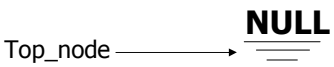
```
bool stack::empty() const
{
  return top_node== NULL;
}
```

Destructor

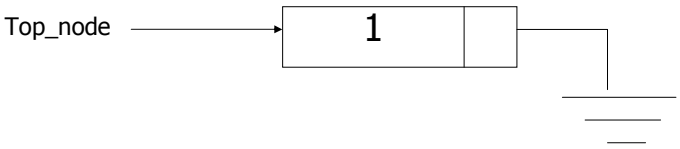
```
stack::~stack()
{
    while( !empty())
    pop();
}
/*if the stack still has items when the
  program finish, this method delete all
  items.*/
```

Example from A to Z

After declaring a new stack:

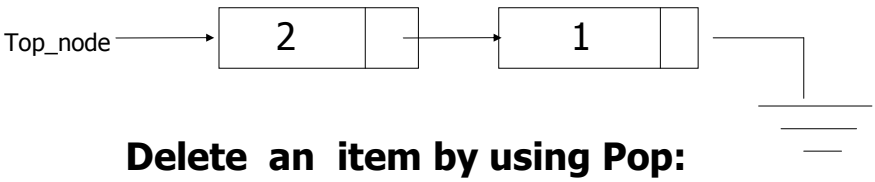


After Adding first item by using Push:

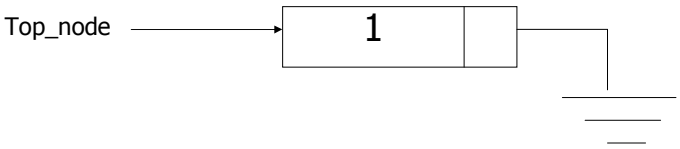


Cont..

Adding another item:

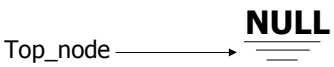


Delete an item by using Pop:



Cont…

After Deleting the last item in the stack:



Try to draw in details all the steps of push and pop in the previous example.

Extra functions

Write a function to find size of the stack.

```
int size( stack s)
{
    int count = 0;
    while(!s.empty())
    {
        s.pop();
        count++;
    }
    return count;
}
```

Cont..

Write a method that reads the bottom of the stack?

```
bool read_bottom(Node_entry &item)const
{
    if(empty())
        return false;
    else
    {
        Node *walk = top_node;
        while(walk→next != NULL)
            walk = walk→next;
        item = walk→entry;
    }
    return true;
}
```

Cont…

Write a method to insert a new item into middle of the stack?

```
bool insert_mid(Node_entry &item)
{
  int mid = size() / 2;
  if(empty())
      push(item);
```

Cont..

```
  int count = 1;
  Node *walk = top_node;
  while(count < mid)
  {
      count++;
      walk = walk → next;
  }
```

Cont..

```
  Node *new_node = new Node(item, walk→next);
  if(new_node == NULL)
      return false;
  walk→next = new_node;
  return true;
}
```