George Gesite BSCpE - 2A

```
#include <iostream>
#include <string.h>
#include <sstream>
using namespace std;
template<typename T>
class Stack {
    public:
     struct node {
       T data;
       node *next, *prev;
     };
  node *bottom, *top;
  int size;
  stack() {
      bottom = NULL;
      top = NULL;
        size = 0;
    }
   bool isEmpty() {
       return (bottom == NULL);
    }
  void push(T n) {
      node *newNode = new node;
      newNode -> data = n;
      if(isEmpty()) {
           bottom = newNode;
           top = newNode;
           newNode -> prev = NULL;
           top -> next = NULL;
       }
      else {
          top -> next = newNode;
          newNode -> prev = top;
          top = newNode;
          top -> next = NULL;
       }
      size++;
   }
```

```
T pop () {
  T temp ;
 temp = top -> data;
 top = top -> prev ;
  size--;
  return temp ;
}
void display() {
  node *temp = bottom ;
for ( ; temp != top; temp = temp -> next) {
      cout << temp -> data << " ";
  } cout << top -> data << endl;
}
bool isPrime (int x) {
  if (x == 2 || x == 1)
    return true;
  for (int i = 2; i < x ; i++) {
    if (x % i == 0)
      return false;
   }
   return true;
 }
string reverse (string x) {
  Stack < char> hold;
   char ch;
  for (int i = 0 ; i < x . length () ; i++) {
        ch = (char) x [i];
          hold . push (ch) ;
      }
   string result = "";
     for (int i = 0 ; i < x.length() ; i++ ) {
        result += hold . pop() ;
      }
    return result ;
  }
```

```
void evaluate (Stack <string> *x , string y ) {
    stringstream ss ( y );
    string temp ;
    while ( ss >> temp ) {
      if ( isPrime ( temp.length() )) {
        temp = reverse ( temp );
          x -> push(temp);
      }
    }
  }
  T getTop() {
    return top -> data;
  }
  T getBottom() {
    return bottom -> data;
  }
};
int main () {
 Stack <string> s;
 string input ;
 cout << "Input : " ;
 getline ( cin , input );
 cout << "Words : " ;
 s . display ();
 cout << "size of stack is " << s. size;
 return 0;
}
```