```
// George Geste   BSCpE-2A      Quiz Stack #1
#include <sstream>
#include <iostream>
using namespace std;
    int size =0, top =-1;
    string str, stack[100];
class Stack {
     public:
     void input();
     void getpalindromeprime ();
     bool isFull();
     bool isEmpty();
     void push (~~int~~ string value);
     void pop();
     void topstack();
     void bottomstack();
     void display();
};
void Stack :: input () {
   cout << "Input: " ;
   getline ( cin , str);
}
void Stack :: getpalindromeprime(){
   String cut;
   stringstream ss(str);
    while (ss>>cut) {
     int c = cut. length();
     int count =0, m= 0, flag =0;
     for (int i = 0 ; i< c; i++) {
           if ( cut [i] != cut [c-i-1])
           count ++;
           break;
      }
      m = c/2;
      for( int j = 2 ; j<=m; j++) {
           if ( c % j ==0)
             flag = 1;
       }
       if (count != 1 && flag==0){
           top++;
           size++;
           stack[top] = cut;
        }
     }
```

```
cout << "Palindromic Prime Words:"<< endl;
 for ( int i = 0 ; i < size ; i++ ) {
     cout << stack [i]<< " " << endl;
    }
 cout<< "Size of stack: "<< size << endl;

 bool Stack ::isEmpty() {
   if (top == -1)
     return true;
     else
    return false;
   }

 bool Stack ::isFull () {
    if (top >= size-1)
     return true;
     else
     return false;
 }
 void Stack :: push (string value) {
   if ( isFull()) {
   cout << "Stack Overflow"<< endl;
  }
   else {
   top++;
   stack [top] = value;
  }}
 void Stack :: pop () {
   if( isEmpty()) {
   cout<< "Stack UnderFlow"<< endl;
  }
   else {
    top--;
  }}

 void Stack :: topstack () {
  if ( is Empty ()) {
  Cout << "Stack UnderFlow << endl;
   }
   else {
   Cout << "Top of stack is"<< stack [top] << endl;
   }}
```

```
void Stack :: bottomstack() {
  if (isEmpty ()) {
   cout << "Stack Underflow" << endl;
  else {
    int bot = 0 ;
   cout << " Bottom of the stack is " << stack [bot] << endl;
  }}
 void Stack :: display () {
   cout << "Stack is " << endl;
  for ( int i = top ; i >= 0 ; i --) {
   cout << "\t" << stack[i] << endl;
   }}

int main ()  {

 int op;
  string val;
 Stack s;
  s. input ();
  s. getpalindromeprime ();
 cout << endl << " Stack Operations: " << endl;
   do {
        cout << "[1] Push()" << endl;
        cout << "[2] Pop()" << endl;
        cout << "[3] Top()" << endl;
        cout << "[4] Bottom()" << endl;
        cout << "[5] Display()" << endl;
      cout << endl << "Enter Option" << endl;
      cin >> op;
      switch (op) {
              case 0 : {
                   break;
                }
              case 1: {  size++;
                 cout << "Enter value to be pushed!" << endl;
                 cin >> val ;
                 s.push ( val);
                 break;
                 }
              case 2: {
                  s. pop();
                  break;
               }
    case 3: {
        s.topstack();
        break;
    }
    case 4: {
        s.bottomstack();
        break;
     }
    case 5: {
        s.display();
        break;
     }
    default : {
     cout << "Invalid Option"<<endl;
     }
   }
  } while ( op!=0)
  return 0;
}
```