Quiz 1

1 Problem 1

Write a simple program to confirm if the string is **Palindromes**. A regular palindrome is a string of numbers or letters that is the same forward as backward. For example, the string "ABCDEDCBA" is a palindrome because it is the same when the string is read from left to right as when the string is read from right to left. You must use **std::stack** to implement the program.

p1.cpp

```
#include<iostream>
#include<stack>
#include<string>
int main()
{
    std::string s;
    std::stack<char> p;
    while(std::cin>>s)
    {
         //fill you code

          //if the string is a palindrome
          std::cout<<"The string is not a Palindrome"<<std::endl;

          //if the string is not a palindrome
          std::cout<<"The string is a Palindrome"<<std::endl;
    }
    return 0;
}
```

1.1 Input

Input consists of strings (one per line) each of which will consist of one to twenty valid characters. There will be no invalid characters in any of the strings.

1.2 Output

Acording to the input is palindrome or not , print “The string is a Palindrome” or “The string is not a Palindrome”

1.3 How to compile the code?

You can use this command to compile the code:

```
$ g++ -std=c++14    p1.cpp –o p1 -Wall -Wextra -pedantic -g3
```

2 Problem 2

Using **linked list** to implement the **stack** ,Complete the operations (comprised of main.cpp, p2.hpp, and p2.cpp), that takes a line as input and outputs the sentence.

Note: You are permitted to modify **p2.cpp** only.

main.cpp

```
#include<iostream>
#include "p2.hpp"

int main()
{
    char s;
    struct Node *top=NULL;
    int num;
    while(std::cin>>s)
    {
        if(s=='i')
        {
            std::cin>>num;
            //push the node
            top=oop::push(top,num);
        }
        else if(s=='d')
        {
            //pop the node
            top=oop::pop(top);
        }
    }
    oop::print(top);
    return 0;
}
```

p2.hpp

```
//#ifndef _oop_linking list_H
//#define _oop_linking list_H
 struct Node
{
    int num;
    Node *next;
};
namespace oop
{
    Node* push(Node *top,int n);
    Node* pop(Node *top);
    void print(Node *top );
};
//#endif
```

p2.cpp

```
#include "p2.hpp"

namespace oop{

    Node* push(Node *top,int n){
    }
    Node* pop(Node *top){
    }
    void print(Node *top ){
    //Output specification: top->space->top->space->……
    //std::cout<<top<<" "; //show example
    }
}
```

2.1 Sample Input

```
i 1 i 2 i 3 d i 4 i 5 d
```

2.2 Sample Output

```
4 2 1
```

2.3 How to compile the code?

You can use this command to compile the code:

```
$ g++ -std=c++14    main.cpp    p2.cpp -o main -Wall -Wextra -pedantic -g3
```

3 How to submit the assignment?

Just upload p1.cpp p2.cpp to E3. Do not rename the file or put in into any directory.