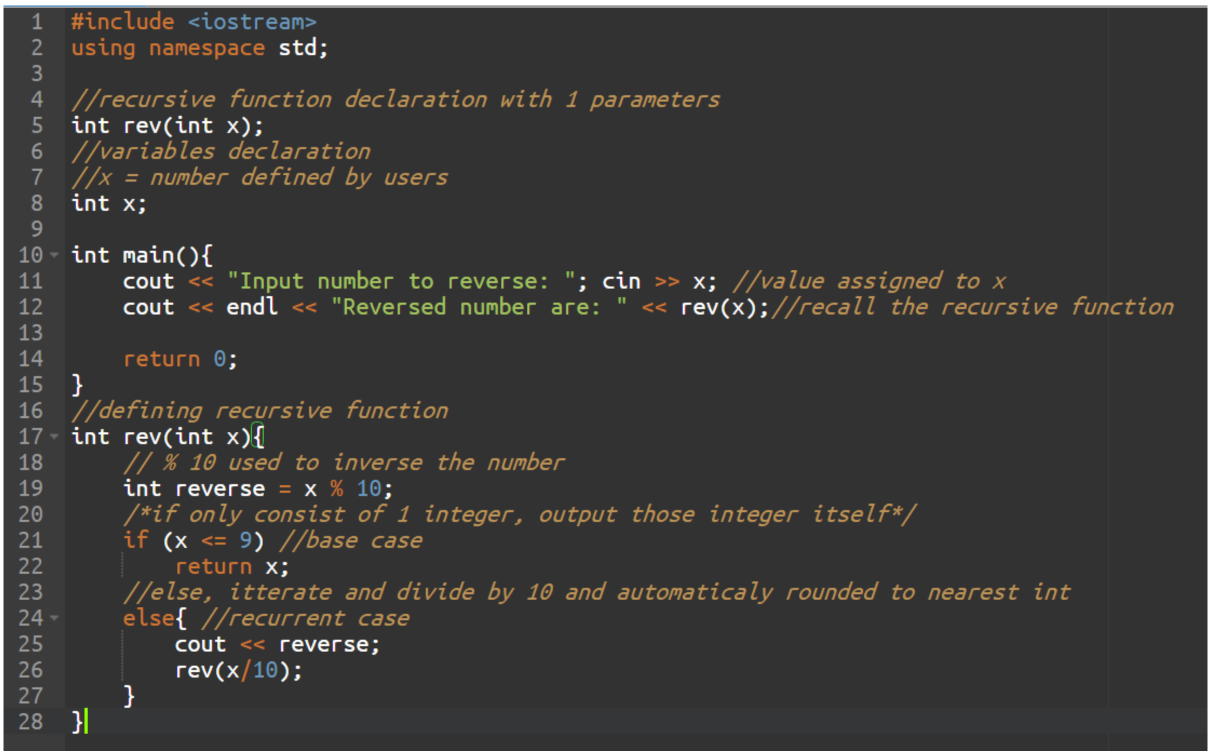
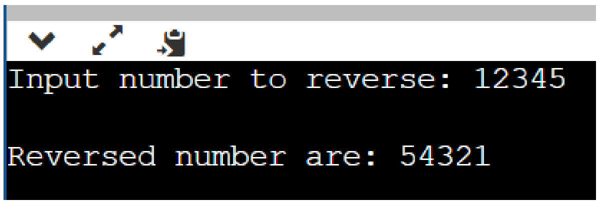
Ramzy Izza Wardhana – 472698 – CSB – Lab Programming Assignment 8

Program 1

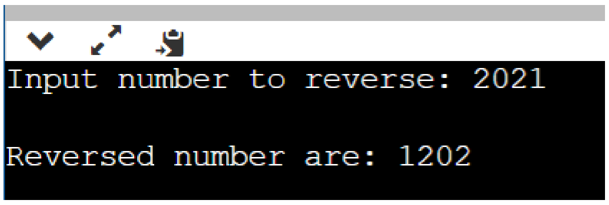
```
#include <iostream>
using namespace std;

//recursive function declaration with 1 parameters
int rev(int x);
//variables declaration
//x = number defined by users
int x;

int main(){
   cout << "Input number to reverse: "; cin >> x; //value assigned to x
   cout << endl << "Reversed number are: " << rev(x);//recall the recursive function

   return 0;
}
//defining recursive function
int rev(int x){
   // % 10 used to inverse the number
   int reverse = x % 10;
   /*if only consist of 1 integer, output those integer itself*/
   if (x <= 9) //base case
      return x;
   //else, itterate and divide by 10 and automaticaly rounded to nearest int
   else{ //recurrent case
      cout << reverse;
      rev(x/10);
   }
}
```



```
#include <iostream>
using namespace std;

//recursive function declaration with 1 parameters
int rev(int x);
//variables declaration
//x = number defined by users
int x;

int main(){
    cout << "Input number to reverse: "; cin >> x; //value assigned to x
    cout << endl << "Reversed number are: " << rev(x);//recall the recursive function

    return 0;
}
//defining recursive function
int rev(int x){
    // % 10 used to inverse the number
    int reverse = x % 10;
    /*if only consist of 1 integer, output those integer itself*/
    if (x <= 9) //base case
        return x;
    //else, itterate and divide by 10 and automaticaly rounded to nearest int
    else{ //recurrent case
        cout << reverse;
        rev(x/10);
    }
}
```



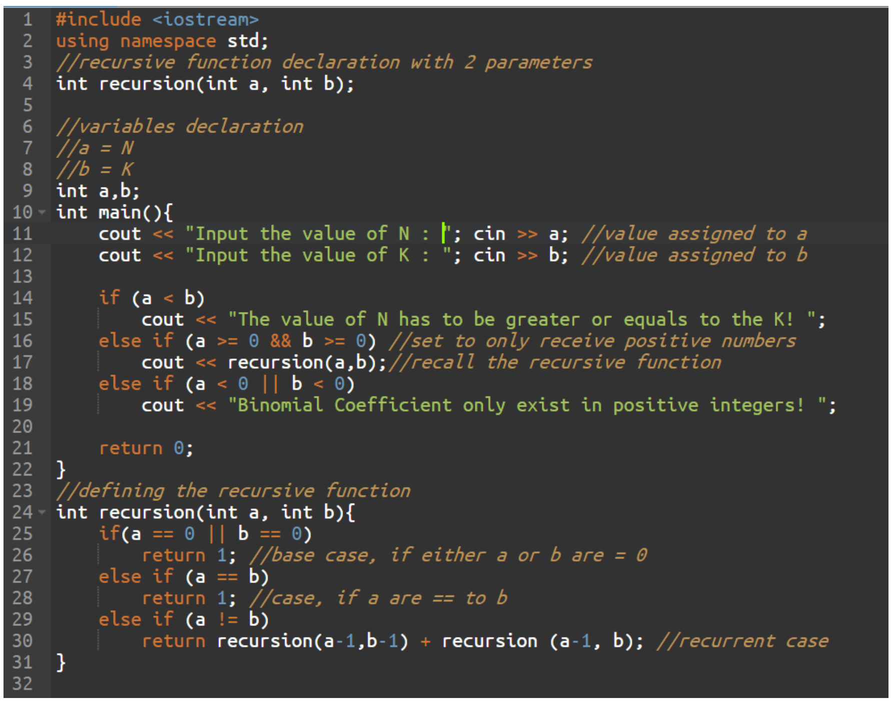
```
Input number to reverse: 12345

Reversed number are: 54321
```



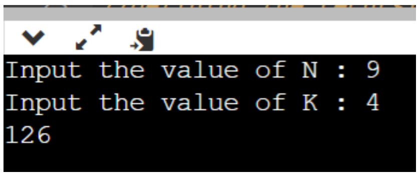
```
Input number to reverse: 2021

Reversed number are: 1202
```

Program 2

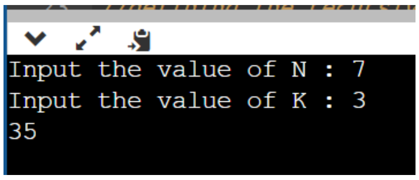
```cpp
#include <iostream>
using namespace std;
//recursive function declaration with 2 parameters
int recursion(int a, int b);

//variables declaration
//a = N
//b = K
int a,b;
int main(){
  cout << "Input the value of N : "; cin >> a; //value assigned to a
  cout << "Input the value of K : "; cin >> b; //value assigned to b

  if (a < b)
    cout << "The value of N has to be greater or equals to the K! ";
  else if (a >= 0 && b >= 0) //set to only receive positive numbers
    cout << recursion(a,b);//recall the recursive function
  else if (a < 0 || b < 0)
    cout << "Binomial Coefficient only exist in positive integers! ";

  return 0;
}
//defining the recursive function
int recursion(int a, int b){
  if(a == 0 || b == 0)
    return 1; //base case, if either a or b are = 0
  else if (a == b)
    return 1; //case, if a are == to b
  else if (a != b)
    return recursion(a-1,b-1) + recursion (a-1, b); //recurrent case
}
```



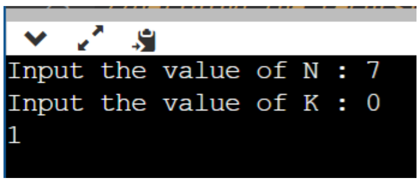
```cpp
#include <iostream>
using namespace std;
//recursive function declaration with 2 parameters
int recursion(int a, int b);

//variables declaration
//a = N
//b = K
int a,b;
int main(){
    cout << "Input the value of N : "; cin >> a; //value assigned to a
    cout << "Input the value of K : "; cin >> b; //value assigned to b

    if (a < b)
        cout << "The value of N has to be greater or equals to the K! ";
    else if (a >= 0 && b >= 0) //set to only receive positive numbers
        cout << recursion(a,b);//recall the recursive function
    else if (a < 0 || b < 0)
        cout << "Binomial Coefficient only exist in positive integers! ";

    return 0;
}
//defining the recursive function
int recursion(int a, int b){
    if(a == 0 || b == 0)
        return 1; //base case, if either a or b are = 0
    else if (a == b)
        return 1; //case, if a are == to b
    else if (a != b)
        return recursion(a-1,b-1) + recursion (a-1, b); //recurrent case
}
```



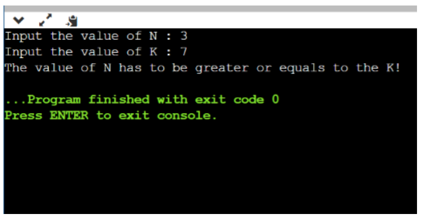
```
Input the value of N : 9
Input the value of K : 4
126
```



```
Input the value of N : 7
Input the value of K : 3
35
```



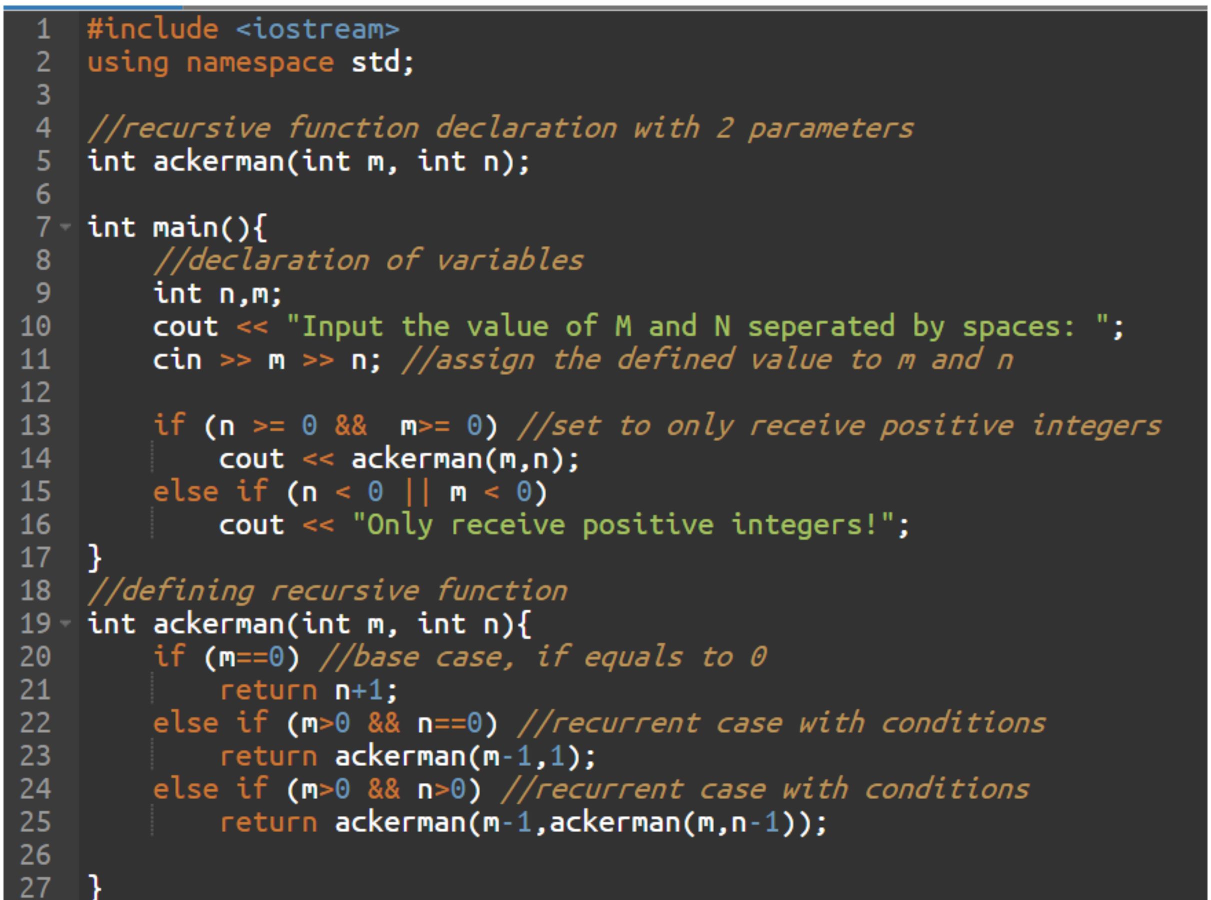
```
Input the value of N : 7
Input the value of K : 0
1
```



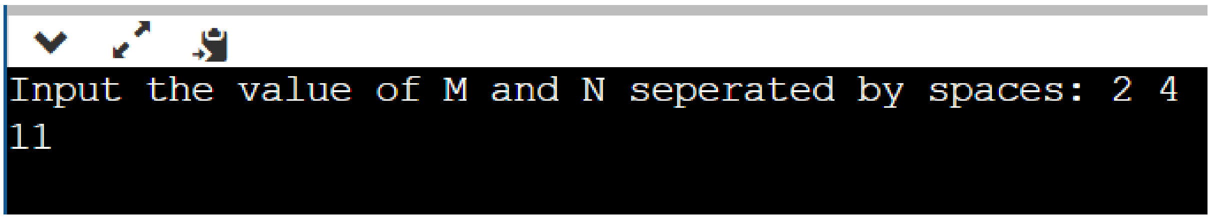
```
Input the value of N : 3
Input the value of K : 7
The value of N has to be greater or equals to the K!

...Program finished with exit code 0
Press ENTER to exit console.
```

Program 3

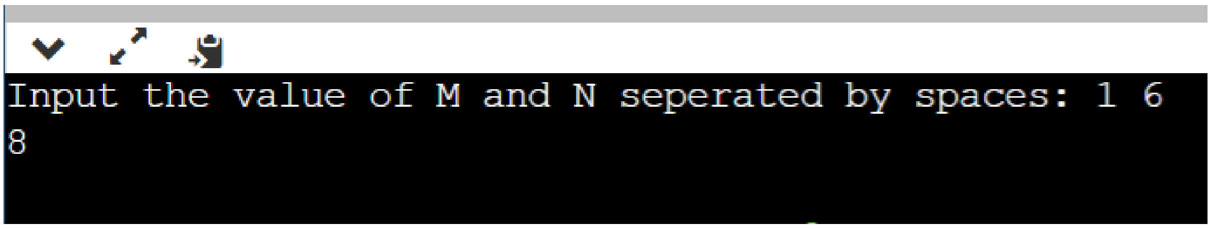
```
#include <iostream>
using namespace std;

//recursive function declaration with 2 parameters
int ackerman(int m, int n);

int main(){
    //declaration of variables
    int n,m;
    cout << "Input the value of M and N seperated by spaces: ";
    cin >> m >> n; //assign the defined value to m and n

    if (n >= 0 &&  m>= 0) //set to only receive positive integers
        cout << ackerman(m,n);
    else if (n < 0 || m < 0)
        cout << "Only receive positive integers!";
}
//defining recursive function
int ackerman(int m, int n){
    if (m==0) //base case, if equals to 0
        return n+1;
    else if (m>0 && n==0) //recurrent case with conditions
        return ackerman(m-1,1);
    else if (m>0 && n>0) //recurrent case with conditions
        return ackerman(m-1,ackerman(m,n-1));

}
```



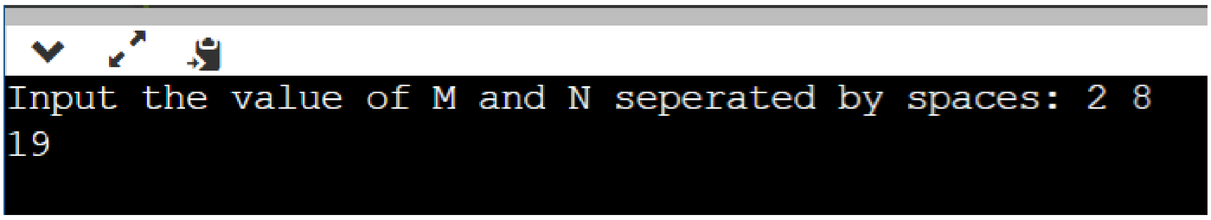
```
#include <iostream>
using namespace std;

//recursive function declaration with 2 parameters
int ackerman(int m, int n);

int main(){
    //declaration of variables
    int n,m;
    cout << "Input the value of M and N seperated by spaces: ";
    cin >> m >> n; //assign the defined value to m and n

    if (n >= 0 &&  m>= 0) //set to only receive positive integers
        cout << ackerman(m,n);
    else if (n < 0 || m < 0)
        cout << "Only receive positive integers!";
}
//defining recursive function
int ackerman(int m, int n){
    if (m==0) //base case, if equals to 0
        return n+1;
    else if (m>0 && n==0) //recurrent case with conditions
        return ackerman(m-1,1);
    else if (m>0 && n>0) //recurrent case with conditions
        return ackerman(m-1,ackerman(m,n-1));

}
```



```
Input the value of M and N seperated by spaces: 2 4
11
```



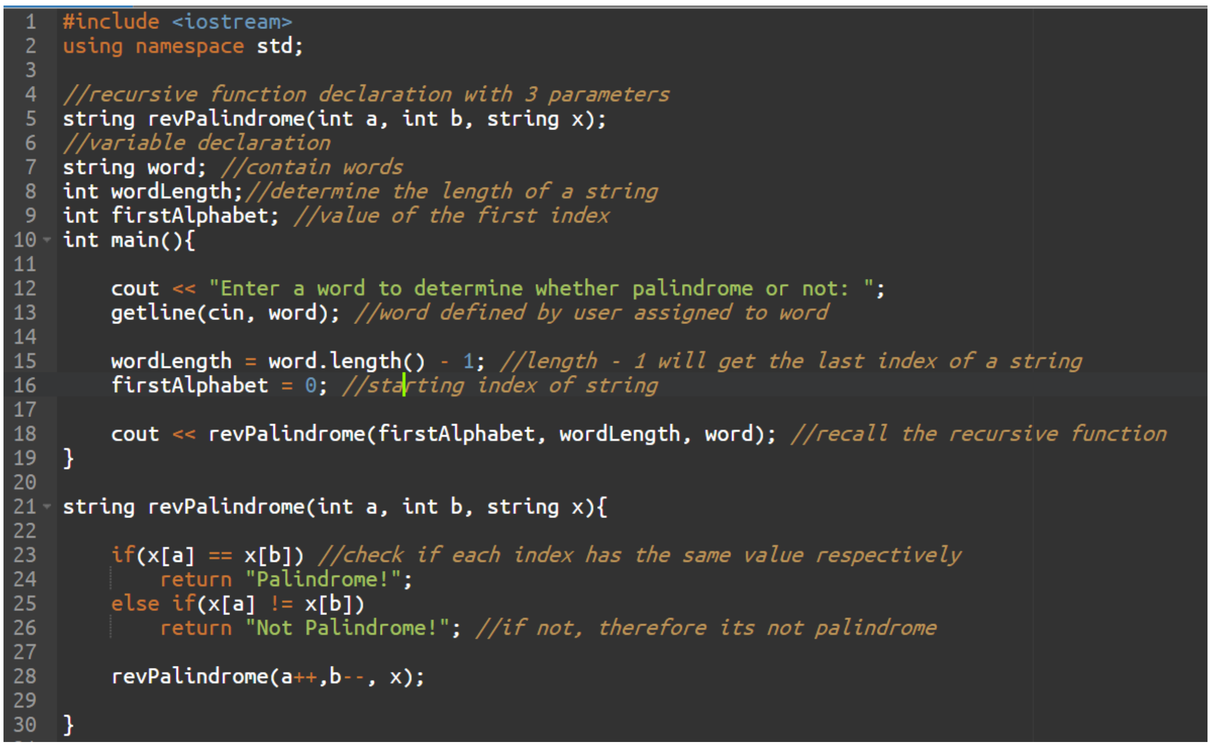
```
Input the value of M and N seperated by spaces: 1 6
8
```



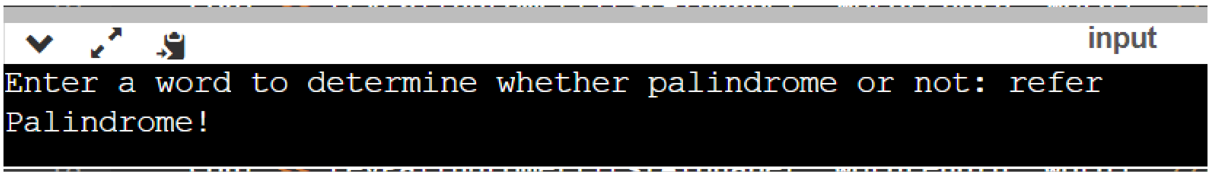
```
Input the value of M and N seperated by spaces: 2 8
19
```

Program 4

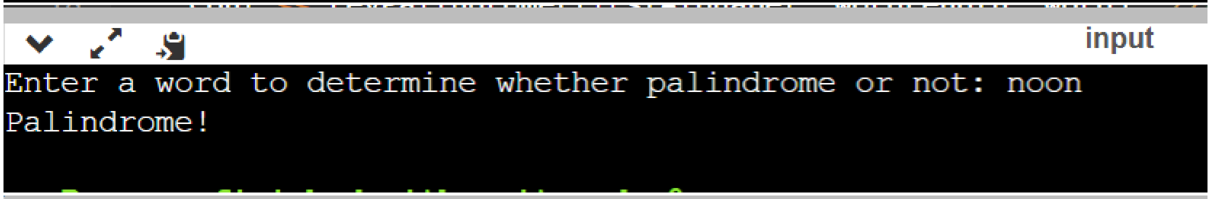
```
#include <iostream>
using namespace std;

//recursive function declaration with 3 parameters
string revPalindrome(int a, int b, string x);
//variable declaration
string word; //contain words
int wordLength;//determine the length of a string
int firstAlphabet; //value of the first index
int main(){

    cout << "Enter a word to determine whether palindrome or not: ";
    getline(cin, word); //word defined by user assigned to word

    wordLength = word.length() - 1; //length - 1 will get the last index of a string
    firstAlphabet = 0; //starting index of string

    cout << revPalindrome(firstAlphabet, wordLength, word); //recall the recursive function
}

string revPalindrome(int a, int b, string x){

    if(x[a] == x[b]) //check if each index has the same value respectively
        return "Palindrome!";
    else if(x[a] != x[b])
        return "Not Palindrome!"; //if not, therefore its not palindrome

    revPalindrome(a++,b--, x);

}
```



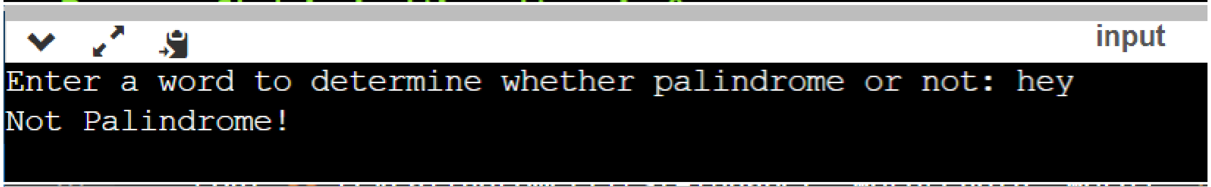
```
#include <iostream>
using namespace std;

//recursive function declaration with 3 parameters
string revPalindrome(int a, int b, string x);
//variable declaration
string word; //contain words
int wordLength;//determine the length of a string
int firstAlphabet; //value of the first index
int main(){

    cout << "Enter a word to determine whether palindrome or not: ";
    getline(cin, word); //word defined by user assigned to word

    wordLength = word.length() - 1; //length - 1 will get the last index of a string
    firstAlphabet = 0; //starting index of string

    cout << revPalindrome(firstAlphabet, wordLength, word); //recall the recursive function
}

string revPalindrome(int a, int b, string x){

    if(x[a] == x[b]) //check if each index has the same value respectively
        return "Palindrome!";
    else if(x[a] != x[b])
        return "Not Palindrome!"; //if not, therefore its not palindrome

    revPalindrome(a++,b--, x);

}
```



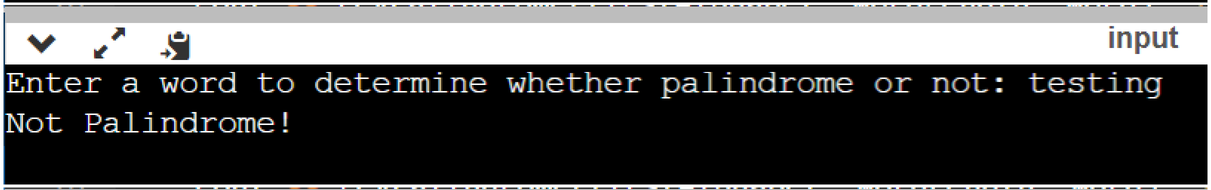
```
Enter a word to determine whether palindrome or not: refer
Palindrome!
```



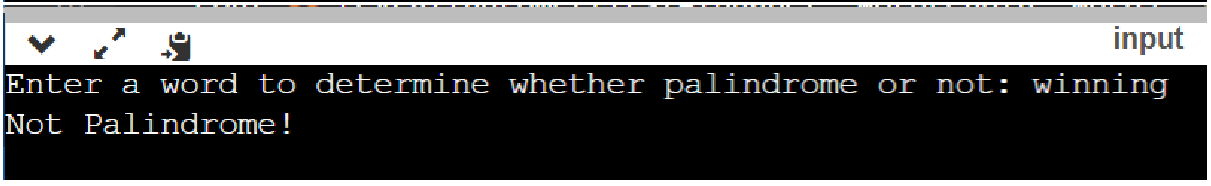
```
Enter a word to determine whether palindrome or not: noon
Palindrome!
```



```
Enter a word to determine whether palindrome or not: hey
Not Palindrome!
```



```
Enter a word to determine whether palindrome or not: testing
Not Palindrome!
```



```
Enter a word to determine whether palindrome or not: winning
Not Palindrome!
```