

```
//=====
// Name      : Assignment #6
// Author    : Kendrick Kwok (912351666)
// Version   : Eclipse C++ 3.8.1 – 5.1
// Date      : 4/15/16
// Description : This assignment was to show our thinking for recursion
//              and solve basic exercises
//=====
```

Assignment: This assignment was to show our thinking for recursion and solve basic exercises.

Problem 1: Compile the program in recursion.cpp. A string is entered and only recurses if the first letter and last letter matches. If the first letter and last letter matches, it recurses and gets deleted. Once it sees that there is only 1 or 0 letters in the string, it went through the whole string and can determine whether or not it is a Palindrome or not.

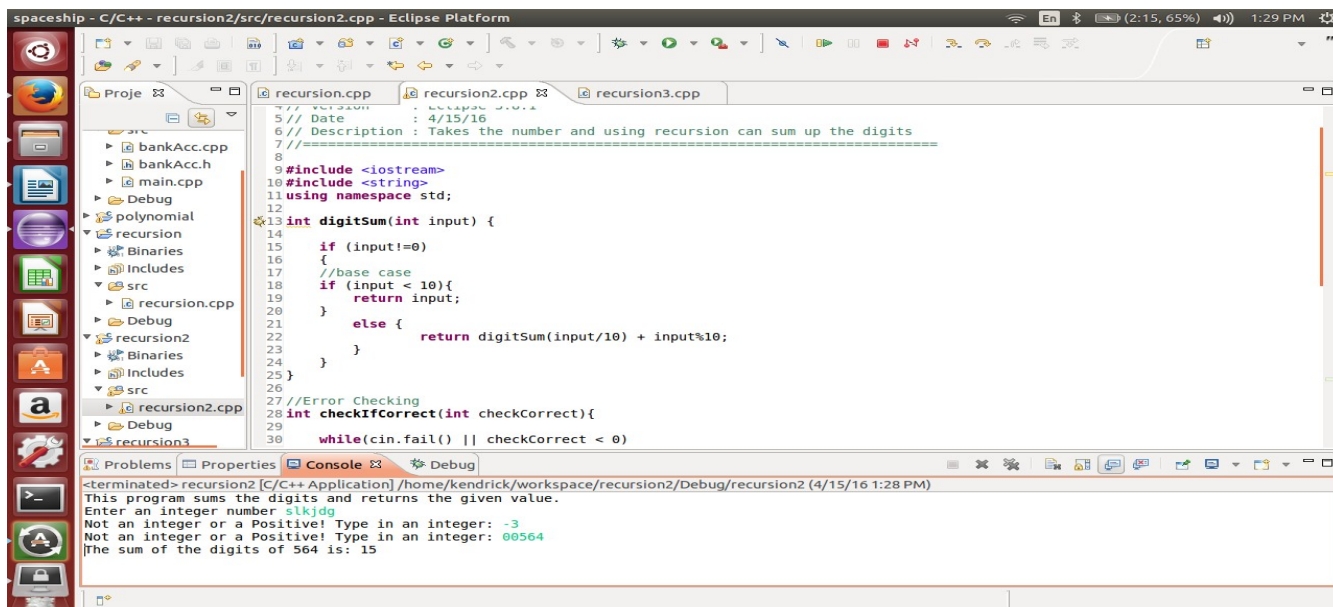
The screenshot shows the Eclipse IDE interface. The main editor window displays the code for `recursion.cpp`. The code defines a recursive function `isPalindrome` that checks if a string is a palindrome by comparing the first and last characters and recursing on the substring. The `main` function prompts the user to enter a word and prints the result.

```
15 bool isPalindrome(const string& input) {
16
17     //--basecase
18     //if after deletion there is 0 or 1 letter left in the string, it is determined
19     //....to be a palinedrome.
20     // Return true and out yes
21     if (input.length()==0 || input.length()==1){
22         cout << "Yes" << endl;
23         return true;
24     }
25
26     //if first letter does not match the last letter, then not a paliundrome
27     //return false
28     else if (input[0] != input[input.length()-1]){
29         cout << "No" << endl;
30         return false;
31     }
32
33     //if first letter does match with last letter, delete first and last letter and use recursion to match rest
34     else{
35         return isPalindrome (input.substr(1, input.length()-2) );
36     }
37
38 }
39
40 int main() {
41     string input;
42     cout << "Enter a word with no space and case sensitive to test: ";
43     getline(cin, input);
44     bool result = isPalindrome(input);
45     cout << "Is " << input << " a Palindrome?: ";
46     if (result) cout << "Yes";
47     else cout << "No";
48     cout << endl;
49 }
```

The console output shows the program execution:

```
<terminated> recursion [C/C++ Application] /home/kendrick/workspace/recursion/Debug/recursion (4/15/16 1:12 PM)
This Program determines whether or not the word is a Palindrome.
Enter a word with no space and case sensitive to test: GeorgewashingtonnotgnihsawegroeG
Is GeorgewashingtonnotgnihsawegroeG a Palindrome?: Yes
```

Problem 2: Compile the program in recursion2.cpp. The program takes an integer and adds the sum of it's digits. The compilation is seen below.

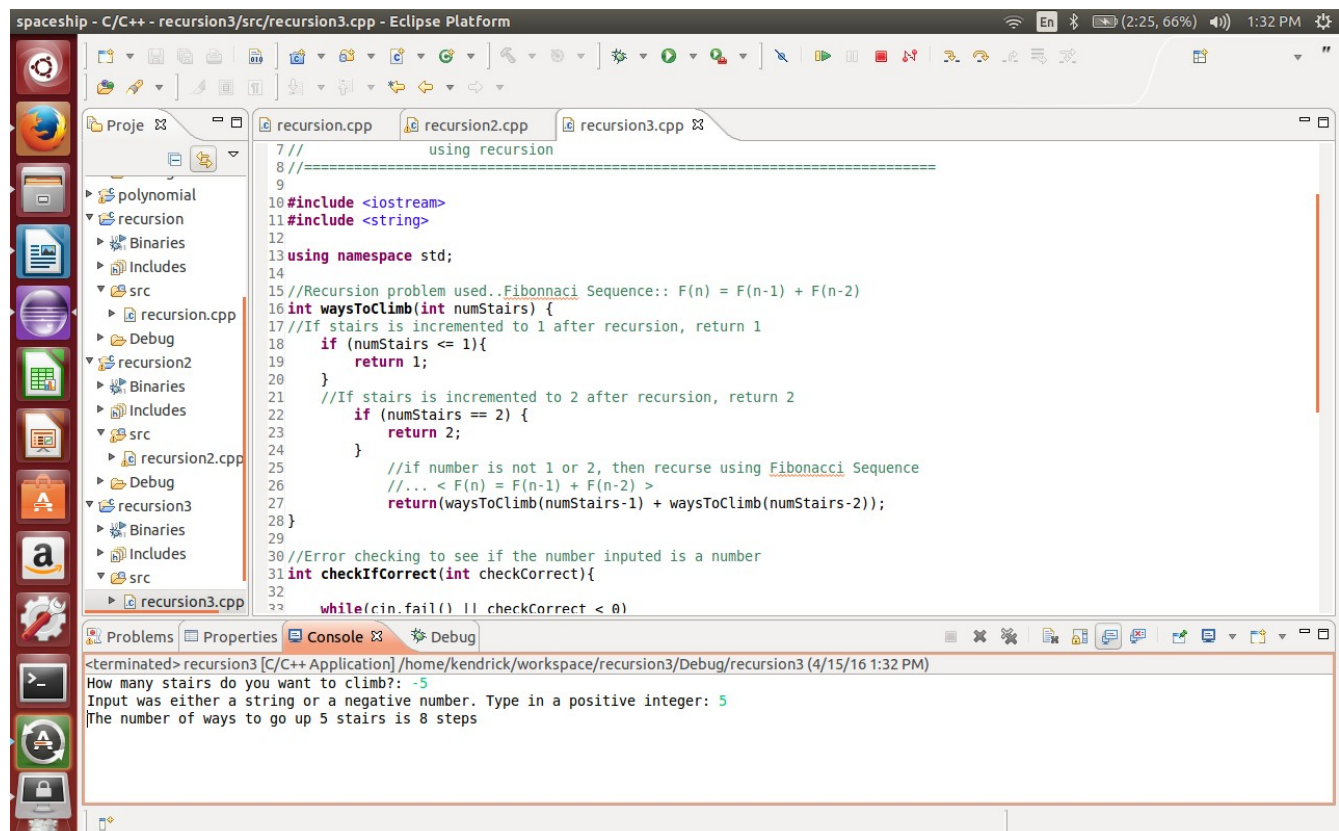


```
spaceship - C/C++ - recursion2/src/recursion2.cpp - Eclipse Platform

// Date : 4/15/16
// Description : Takes the number and using recursion can sum up the digits
//=====
9#include <iostream>
10#include <string>
11using namespace std;
12
13int digitSum(int input) {
14
15    if (input!=0)
16    {
17        //base case
18        if (input < 10){
19            return input;
20        }
21        else {
22            return digitSum(input/10) + input%10;
23        }
24    }
25}
26
27//Error Checking
28int checkIfCorrect(int checkCorrect){
29
30    while(cin.fail() || checkCorrect < 0)
```

```
<terminated> recursion2 [C/C++ Application] /home/kendrick/workspace/recursion2/Debug/recursion2 (4/15/16 1:28 PM)
This program sums the digits and returns the given value.
Enter an integer number s1kjdg
Not an integer or a Positive! Type in an integer: -3
Not an integer or a Positive! Type in an integer: 00564
The sum of the digits of 564 is: 15
```

Problem 3: Compile the program in recursion3.cpp. The program takes an integer number and using the Fibonnacci Sequence, determines the amount of ways a user can take steps up the stairs. Below is the sample compilation.



```
spaceship - C/C++ - recursion3/src/recursion3.cpp - Eclipse Platform

7//
8//=====
9
10#include <iostream>
11#include <string>
12
13using namespace std;
14
15//Recursion problem used..Fibonnaci Sequence:: F(n) = F(n-1) + F(n-2)
16int waysToClimb(int numStairs) {
17    //If stairs is incremented to 1 after recursion, return 1
18    if (numStairs <= 1){
19        return 1;
20    }
21    //If stairs is incremented to 2 after recursion, return 2
22    if (numStairs == 2) {
23        return 2;
24    }
25    //if number is not 1 or 2, then recurse using Fibonacci Sequence
26    //... < F(n) = F(n-1) + F(n-2) >
27    return(waysToClimb(numStairs-1) + waysToClimb(numStairs-2));
28}
29
30//Error checking to see if the number inputed is a number
31int checkIfCorrect(int checkCorrect){
32
33    while(cin.fail() || checkCorrect < 0)
```

```
<terminated> recursion3 [C/C++ Application] /home/kendrick/workspace/recursion3/Debug/recursion3 (4/15/16 1:32 PM)
How many stairs do you want to climb?: -5
Input was either a string or a negative number. Type in a positive integer: 5
The number of ways to go up 5 stairs is 8 steps
```