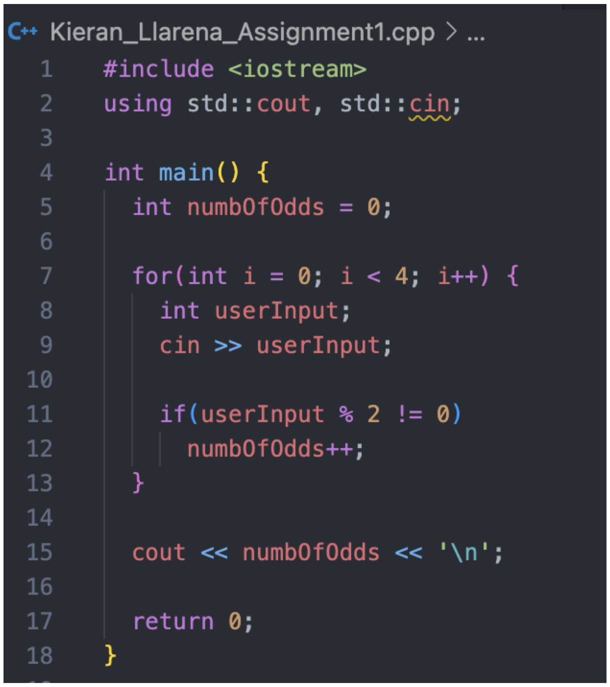
Assignment 1

Lab Explanation

The requirements for this lab were to create a program that would output the number of odd numbers based upon 4 inputted numbers from the user.

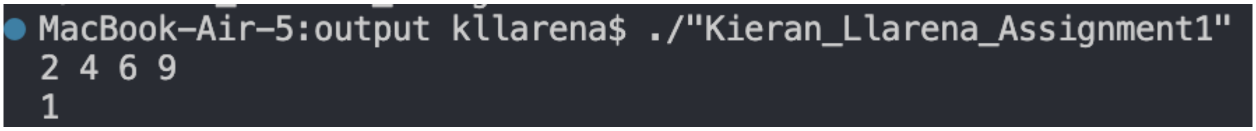
Code



```cpp
#include <iostream>
using std::cout, std::cin;

int main() {
  int numbOfOdds = 0;

  for(int i = 0; i < 4; i++) {
    int userInput;
    cin >> userInput;

    if(userInput % 2 != 0)
      numbOfOdds++;
  }

  cout << numbOfOdds << '\n';

  return 0;
}
```

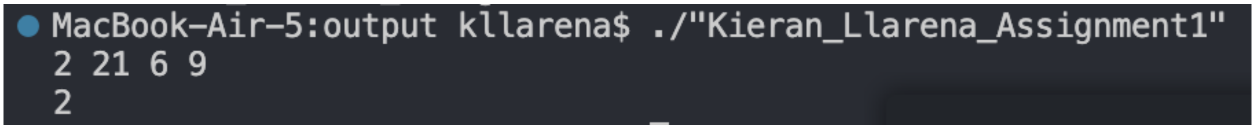
In my program, I decided to use a for loop to input 4 numbers from the user. Everytime a number was inputted, the input would be tested to see if it was odd. This was done using the modulus operator, as an odd number modulus 2 would result in a remainder that is greater than 0. Whenever an odd number was detected, the number of odd numbers (numbOfOdds) would increase by 1. At the end of the program, the total number of odd numbers would be outputted. Additionally, I decided to only use std::cout and std::cin instead of the entire std library to save memory and increase the speed of the application. I also decided to use ‘\n’ instead of std::endl to increase the speed of the application as well.

Test 1



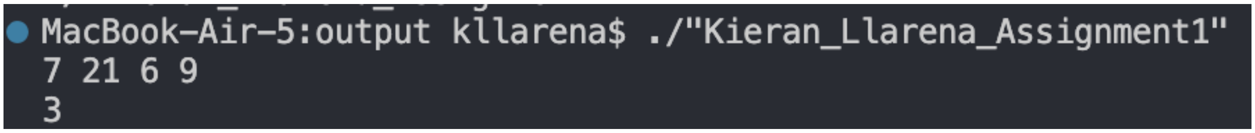
```
MacBook-Air-5:output kllarena$ ./"Kieran_Llarena_Assignment1"
2 4 6 9
1
```

Test 2



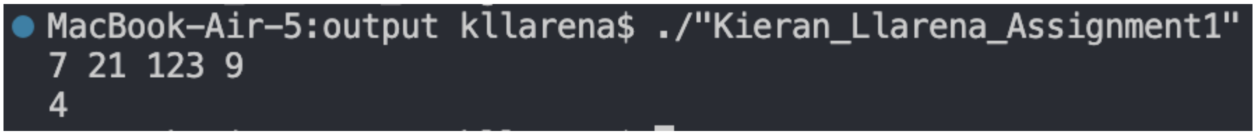
```
MacBook-Air-5:output kllarena$ ./"Kieran_Llarena_Assignment1"
2 21 6 9
2
```

Test 3



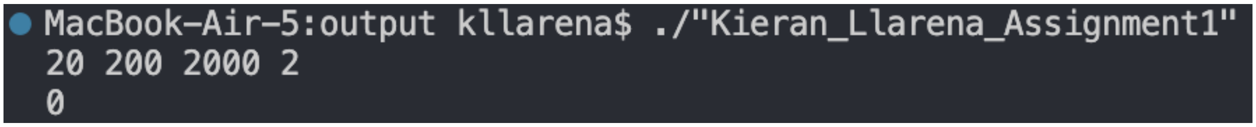
```
MacBook-Air-5:output kllarena$ ./"Kieran_Llarena_Assignment1"
7 21 6 9
3
```

Test 4



```
MacBook-Air-5:output kllarena$ ./"Kieran_Llarena_Assignment1"
7 21 123 9
4
```

Test 5



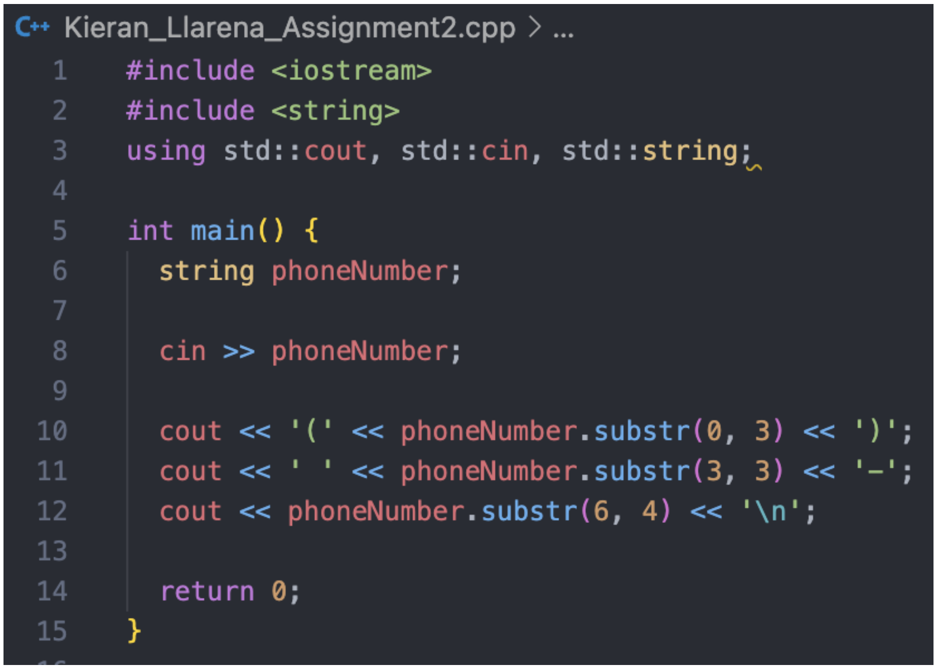
```
MacBook-Air-5:output kllarena$ ./"Kieran_Llarena_Assignment1"
20 200 2000 2
0
```

Assignment 2

Lab Explanation

The requirements for this lab were to create a program that would take an input of a string composed of 10 digits from the user and convert it into a phone number.

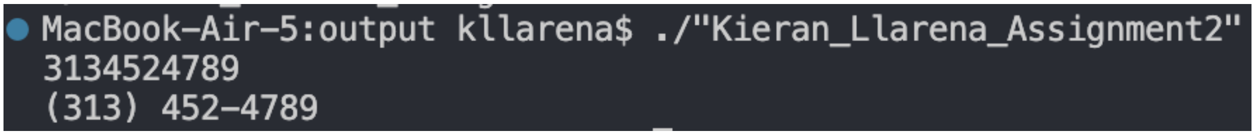
Code



```cpp
#include <iostream>
#include <string>
using std::cout, std::cin, std::string;

int main() {
  string phoneNumber;

  cin >> phoneNumber;

  cout << '(' << phoneNumber.substr(0, 3) << ')';
  cout << ' ' << phoneNumber.substr(3, 3) << '-';
  cout << phoneNumber.substr(6, 4) << '\n';

  return 0;
}
```

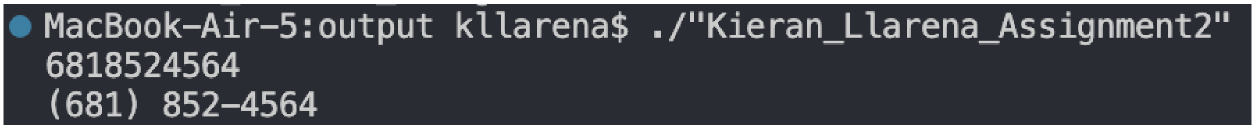
First off, I decided to only use std::cout, std::cin, and std::string to save memory and increase the speed of the application. I then declared the string phoneNumber to get the input of a 10 digit string from the user. After that, I would use the substr() method to then convert the 10 digit string to a phone number by taking the indice of the string (1st parameter of substr) and the number of characters (2nd parameter of substr), and then concatenating the values together to produce the final output. ‘\n’ was used instead of std::endl to increase the speed of the application.

Test 1



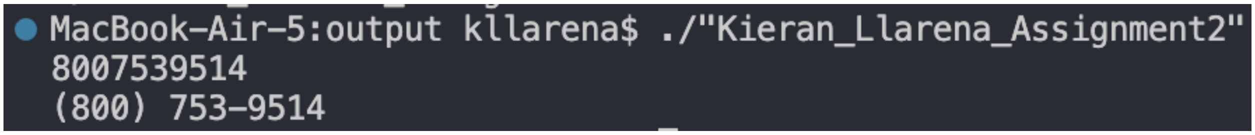
```
MacBook-Air-5:output kllarena$ ./"Kieran_Llarena_Assignment2"
3134524789
(313) 452-4789
```

Test 2



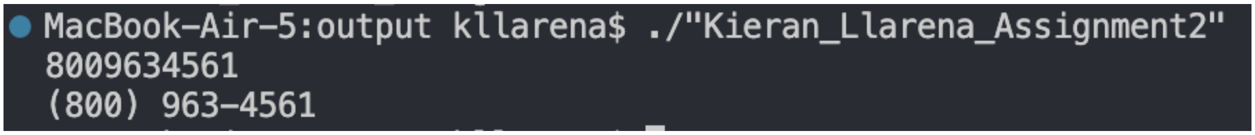
```
MacBook-Air-5:output kllarena$ ./"Kieran_Llarena_Assignment2"
6818524564
(681) 852-4564
```

Test 3



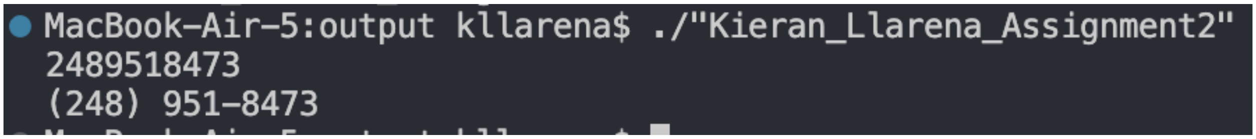
```
MacBook-Air-5:output kllarena$ ./"Kieran_Llarena_Assignment2"
8007539514
(800) 753-9514
```

Test 4



```
MacBook-Air-5:output kllarena$ ./"Kieran_Llarena_Assignment2"
8009634561
(800) 963-4561
```

Test 5



```
MacBook-Air-5:output kllarena$ ./"Kieran_Llarena_Assignment2"
2489518473
(248) 951-8473
```