Pr10-2.cpp

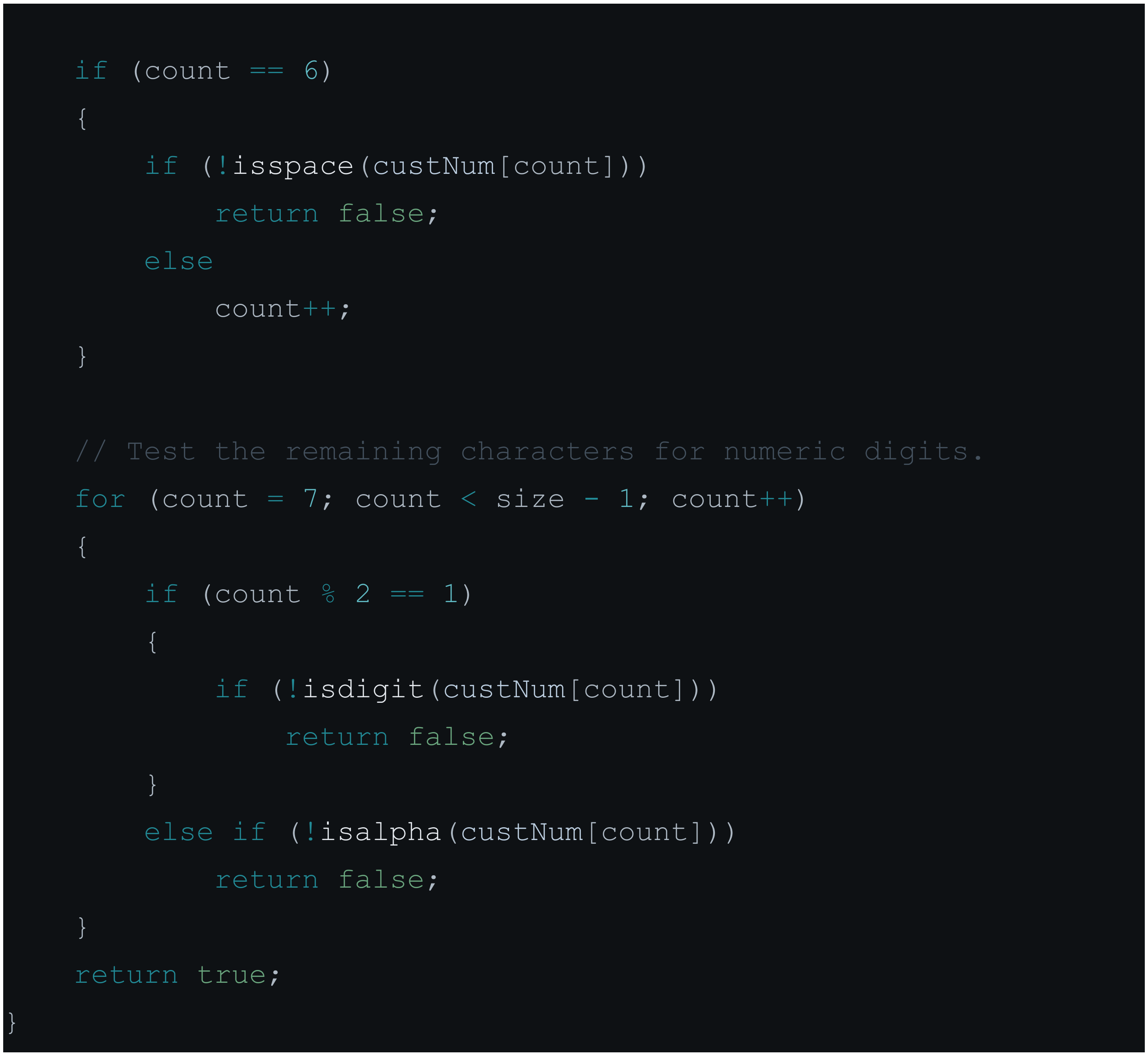
```cpp
// This program tests a customer number to determine whether
// it is in the proper format.
#include "customer.h"
#include <iostream>
#include <cctype>
using namespace std;

// Function prototype

int main()
{
   const int SIZE = 13; // Array size
   char customer[SIZE]; // To hold a customer number
   Customer c;

   // Get the customer number.
   cout << "Enter a customer number in the form ";
   cout << "LNLNLN<space>NLNLNL\n";
   cout << "(L=Letters N=Numbers): ";
   cin.getline(customer, SIZE);

   // Determine whether it is valid.
   if (c.testNum(customer, SIZE))
      cout << "That's a valid customer number.\n";
   else
   {
      cout << "That is not the proper format of the ";
      cout << "customer number.\nHere is an example:\n";
      cout << "   a1b2c3 1a2b3c\n";
   }
   return 0;
}
```

Customer.h

```
class Customer
{
private:
public:
    bool testNum(char[], int);
};
```

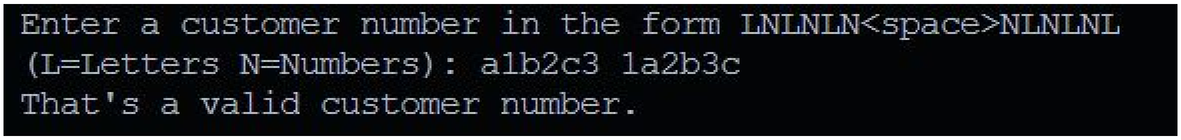
Customer.cpp

```
#include "customer.h"
#include <string>

using namespace std;

//**********************************************************
// Definition of function testNum.                         *
// This function determines whether the custNum parameter  *
// holds a valid customer number. The size parameter is    *
// the size of the custNum array.                          *
//**********************************************************

bool Customer::testNum(char custNum[], int size)
{
    int count; // Loop counter

    // Test the first three characters for alphabetic letters.
    for (count = 0; count < 6; count++)
        if (count % 2 == 0)
        {
            if (!isalpha(custNum[count]))
                return false;
        }
        else if (!isdigit(custNum[count]))
            return false;
```



```
    if (count == 6)
    {
        if (!isspace(custNum[count]))
            return false;
        else
            count++;
    }

    // Test the remaining characters for numeric digits.
    for (count = 7; count < size - 1; count++)
    {
        if (count % 2 == 1)
        {
            if (!isdigit(custNum[count]))
                return false;
        }
        else if (!isalpha(custNum[count]))
            return false;
    }
    return true;
}
```

Output:



```
Enter a customer number in the form LNLNLN<space>NLNLNL
(L=Letters N=Numbers): a1b2c3 1a2b3c
That's a valid customer number.
```