San Francisco Bay University

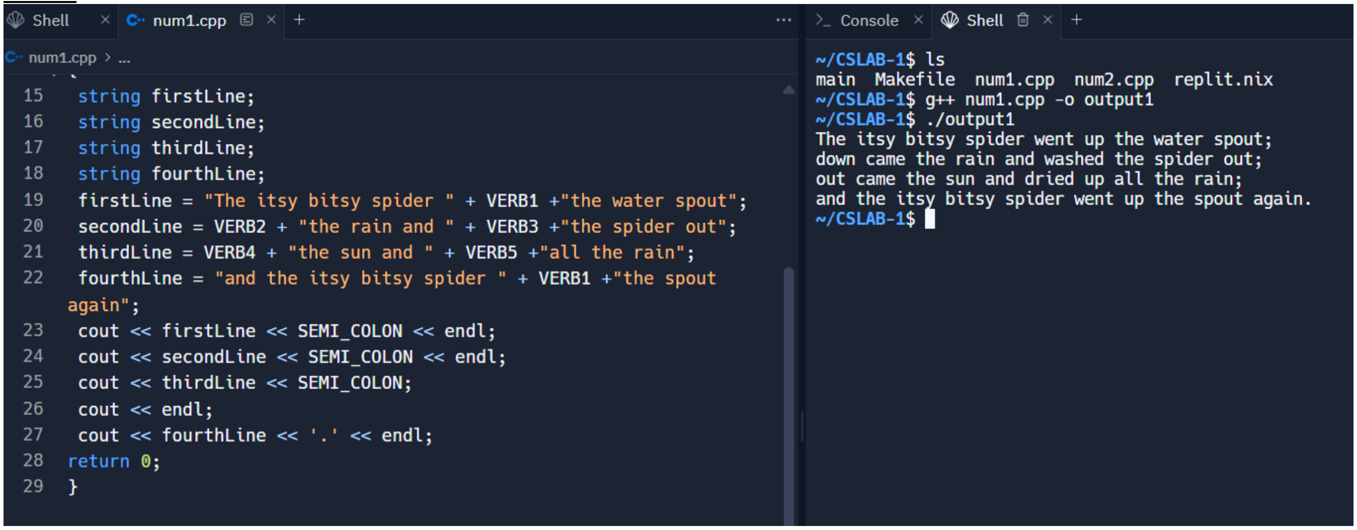
CS360L - Programming in C and C++ Lab
Lab Assignment #1

Due day: 2/13/2024

Kritika Regmi
ID:19702

1. Let’s examine / run the following C++ program regarding *string* data type and related operators.

ANS:



```cpp
 string firstLine;
 string secondLine;
 string thirdLine;
 string fourthLine;
 firstLine = "The itsy bitsy spider " + VERB1 +"the water spout";
 secondLine = VERB2 + "the rain and " + VERB3 +"the spider out";
 thirdLine = VERB4 + "the sun and " + VERB5 +"all the rain";
 fourthLine = "and the itsy bitsy spider " + VERB1 +"the spout
again";
 cout << firstLine << SEMI_COLON << endl;
 cout << secondLine << SEMI_COLON << endl;
 cout << thirdLine << SEMI_COLON;
 cout << endl;
 cout << fourthLine << '.' << endl;
return 0;
}
```

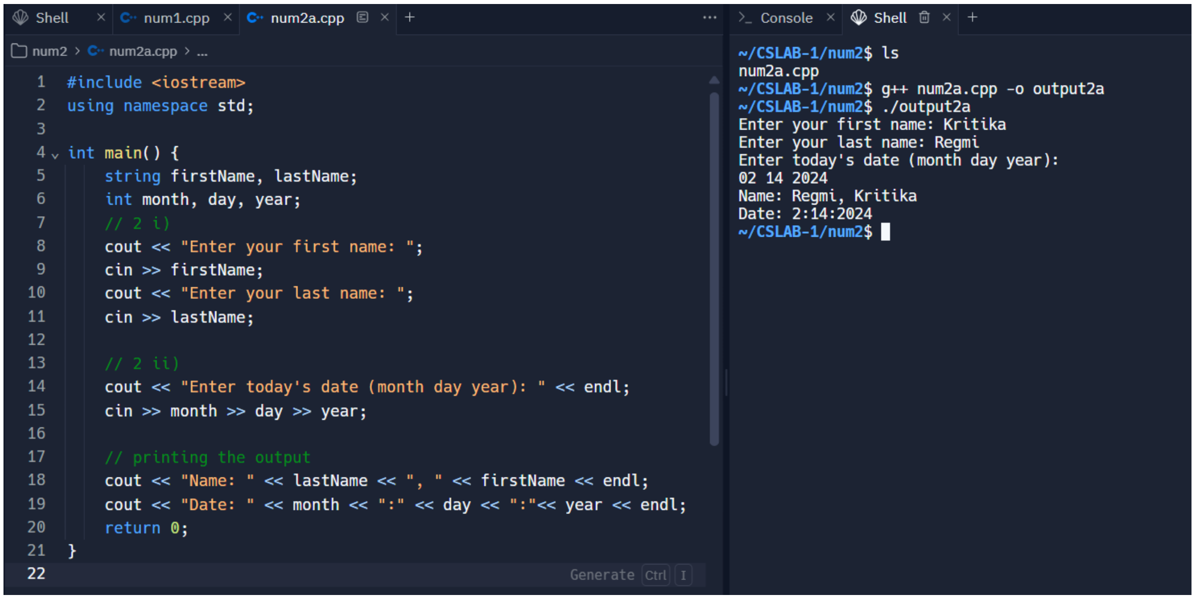
```
~/CSLAB-1$ ls
main  Makefile  num1.cpp  num2.cpp  replit.nix
~/CSLAB-1$ g++ num1.cpp -o output1
~/CSLAB-1$ ./output1
The itsy bitsy spider went up the water spout;
down came the rain and washed the spider out;
out came the sun and dried up all the rain;
and the itsy bitsy spider went up the spout again.
~/CSLAB-1$ 
```

2. Focuses on constructing output statements. Program Shell is the outline of a program. Use this shell for Question#1 through #3.

a. Question#1: Write a program to read-in from keyboard and print the following information single spaced on the screen. Use literal constants in the output statements for each of the data items to be written on the screen. Run your program to verify that the output is as specified.

i. your name (last name, comma, blank, first name)
ii. today’s date (month:day:year)

ANS:

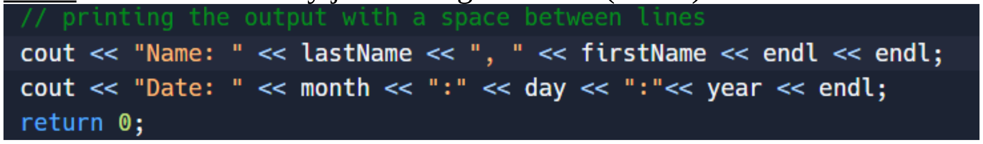


```cpp
#include <iostream>
using namespace std;

int main() {
    string firstName, lastName;
    int month, day, year;
    // 2 i)
    cout << "Enter your first name: ";
    cin >> firstName;
    cout << "Enter your last name: ";
    cin >> lastName;

    // 2 ii)
    cout << "Enter today's date (month day year): " << endl;
    cin >> month >> day >> year;

    // printing the output
    cout << "Name: " << lastName << ", " << firstName << endl;
    cout << "Date: " << month << ":" << day << ":"<< year << endl;
    return 0;
}
```

```
~/CSLAB-1/num2$ ls
num2a.cpp
~/CSLAB-1/num2$ g++ num2a.cpp -o output2a
~/CSLAB-1/num2$ ./output2a
Enter your first name: Kritika
Enter your last name: Regmi
Enter today's date (month day year):
02 14 2024
Name: Regmi, Kritika
Date: 2:14:2024
~/CSLAB-1/num2$
```

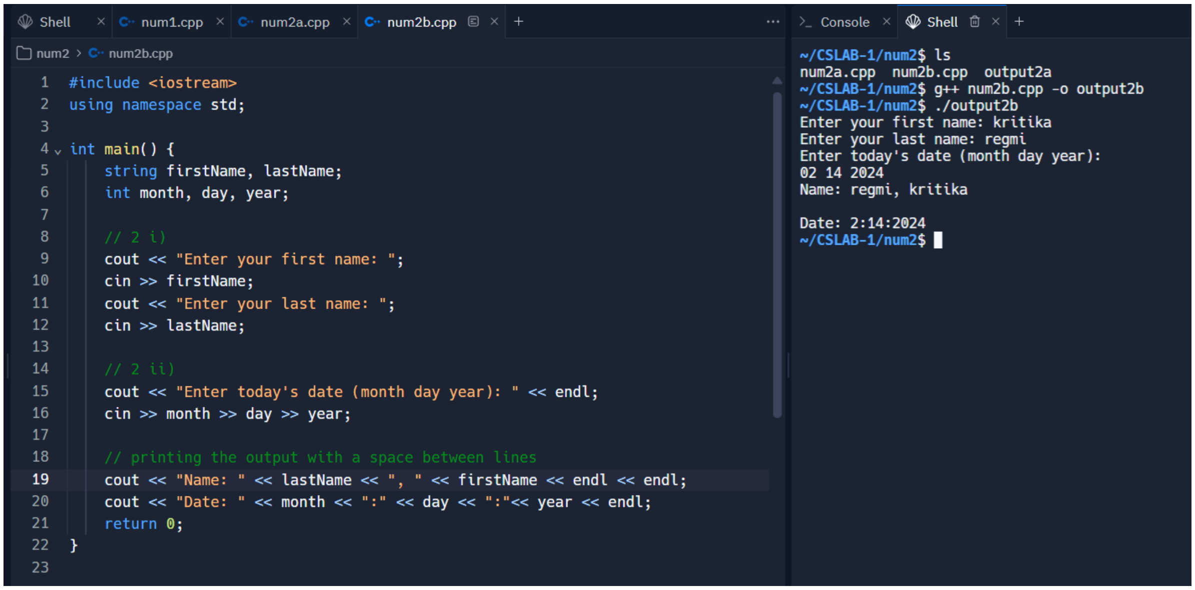
b. Question#2: Change your program so that there is a space between the two lines of output.

ANS: We can do this by just adding a newline('endl') after the first line of output.



```cpp
// printing the output with a space between lines
cout << "Name: " << lastName << ", " << firstName << endl << endl;
cout << "Date: " << month << ":" << day << ":"<< year << endl;
return 0;
```

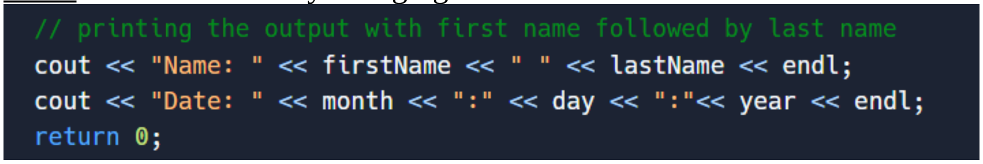
Full code:



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

int main() {
    string firstName, lastName;
    int month, day, year;

    // 2 i)
    cout << "Enter your first name: ";
    cin >> firstName;
    cout << "Enter your last name: ";
    cin >> lastName;

    // 2 ii)
    cout << "Enter today's date (month day year): " << endl;
    cin >> month >> day >> year;

    // printing the output with a space between lines
    cout << "Name: " << lastName << ", " << firstName << endl << endl;
    cout << "Date: " << month << ":" << day << ":"<< year << endl;
    return 0;
}
```

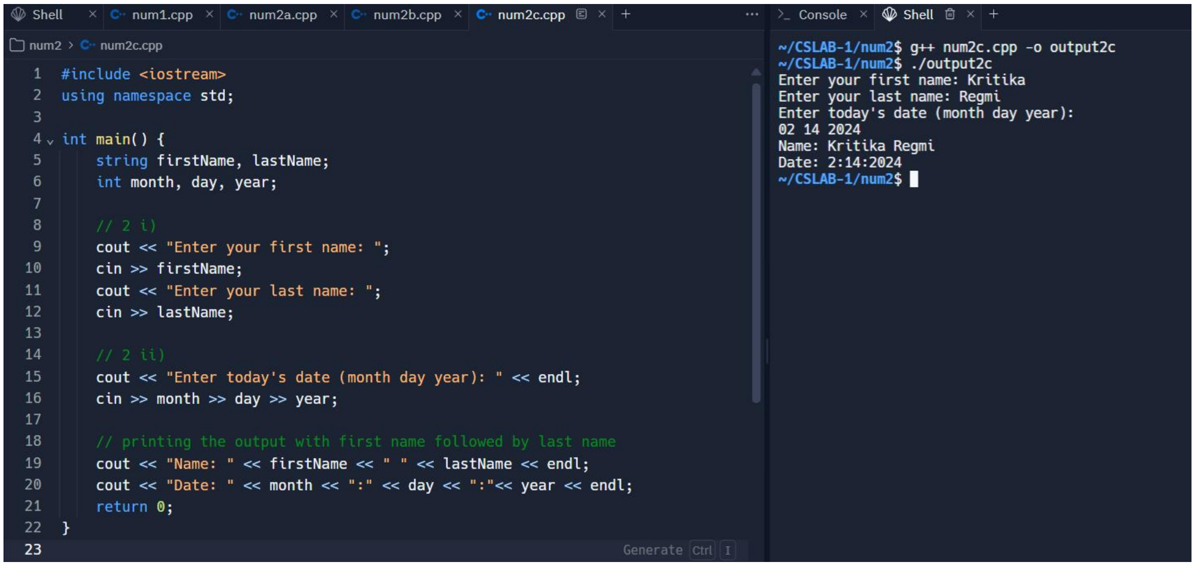
```
~/CSLAB-1/num2$ ls
num2a.cpp  num2b.cpp  output2a
~/CSLAB-1/num2$ g++ num2b.cpp -o output2b
~/CSLAB-1/num2$ ./output2b
Enter your first name: kritika
Enter your last name: regmi
Enter today's date (month day year):
02 14 2024
Name: regmi, kritika

Date: 2:14:2024
~/CSLAB-1/num2$
```

c. Question#3: Change your program so that your first name is printed followed by your last name, with a blank in between the names.
ANS: We can do this by changing the code like below:



```
// printing the output with first name followed by last name
cout << "Name: " << firstName << " " << lastName << endl;
cout << "Date: " << month << ":" << day << ":"<< year << endl;
return 0;
```

Full code:

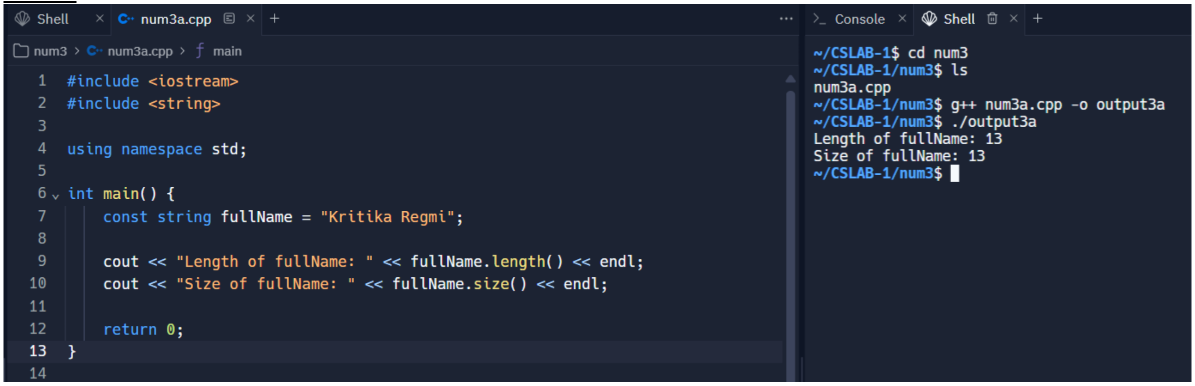


```
#include <iostream>
using namespace std;

int main() {
    string firstName, lastName;
    int month, day, year;

    // 2 i)
    cout << "Enter your first name: ";
    cin >> firstName;
    cout << "Enter your last name: ";
    cin >> lastName;

    // 2 ii)
    cout << "Enter today's date (month day year): " << endl;
    cin >> month >> day >> year;

    // printing the output with first name followed by last name
    cout << "Name: " << firstName << " " << lastName << endl;
    cout << "Date: " << month << ":" << day << ":"<< year << endl;
    return 0;
}
```

```
~/CSLAB-1/num2$ g++ num2c.cpp -o output2c
~/CSLAB-1/num2$ ./output2c
Enter your first name: Kritika
Enter your last name: Regmi
Enter today's date (month day year):
02 14 2024
Name: Kritika Regmi
Date: 2:14:2024
~/CSLAB-1/num2$
```

3. Use the following program shell for Question#1 through #3.

a. Question#1: Write a named string constant made up of your first and last names with a blank in between. Write the statements to print out the result of applying *length* and *size* to your named constant object. Compile and run your program.
ANS:



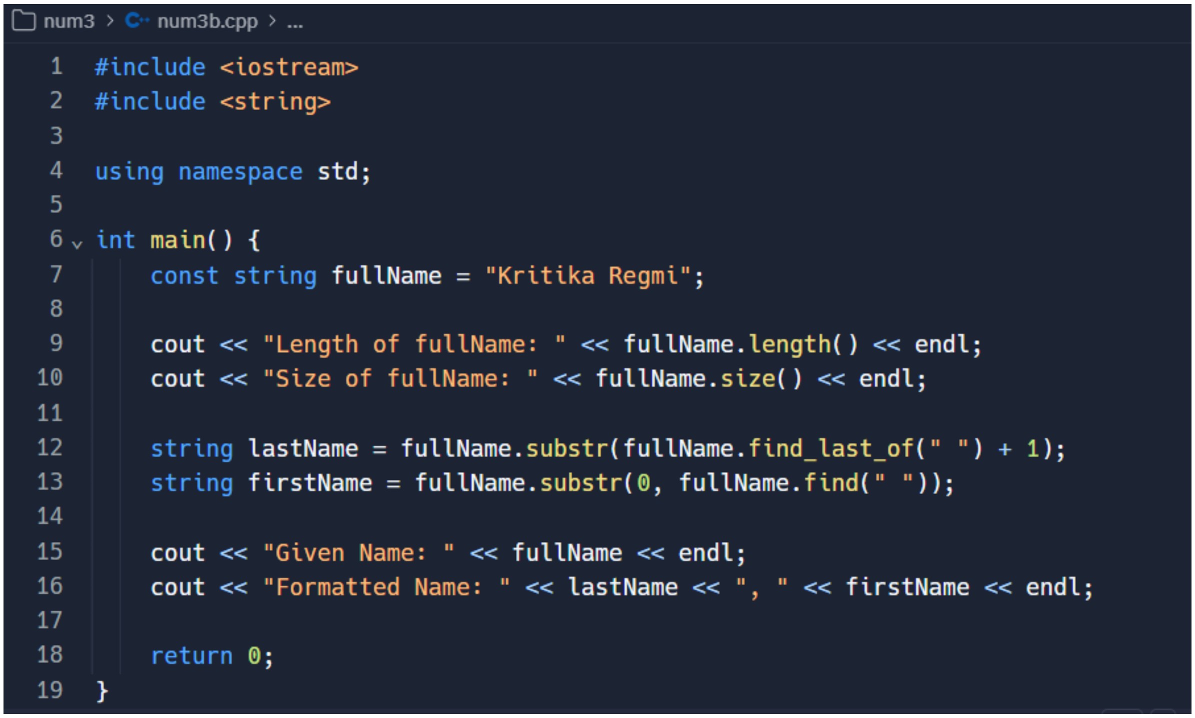
```
#include <iostream>
#include <string>

using namespace std;

int main() {
    const string fullName = "Kritika Regmi";

    cout << "Length of fullName: " << fullName.length() << endl;
    cout << "Size of fullName: " << fullName.size() << endl;

    return 0;
}
```

```
~/CSLAB-1$ cd num3
~/CSLAB-1/num3$ ls
num3a.cpp
~/CSLAB-1/num3$ g++ num3a.cpp -o output3a
~/CSLAB-1/num3$ ./output3a
Length of fullName: 13
Size of fullName: 13
~/CSLAB-1/num3$
```

b. Question#2: Add statements to your Question#1 program to print your name formatted as last name first, followed by a comma and your first name. Use function *substr* to accomplish this task. Compile and run your program.

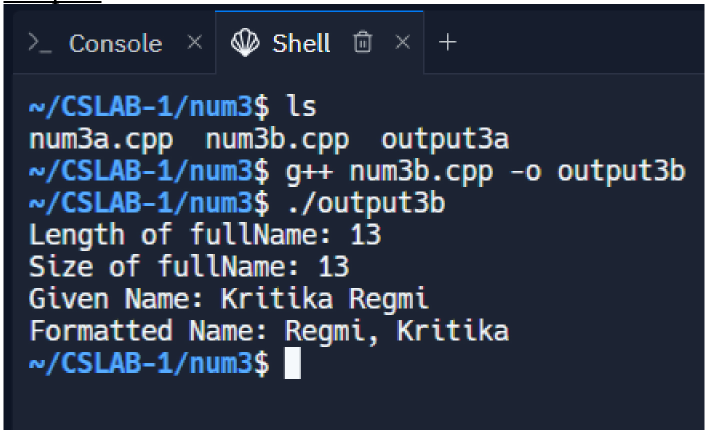
ANS:

Code:



```cpp
#include <iostream>
#include <string>

using namespace std;

int main() {
    const string fullName = "Kritika Regmi";

    cout << "Length of fullName: " << fullName.length() << endl;
    cout << "Size of fullName: " << fullName.size() << endl;

    string lastName = fullName.substr(fullName.find_last_of(" ") + 1);
    string firstName = fullName.substr(0, fullName.find(" "));

    cout << "Given Name: " << fullName << endl;
    cout << "Formatted Name: " << lastName << ", " << firstName << endl;

    return 0;
}
```

Output:

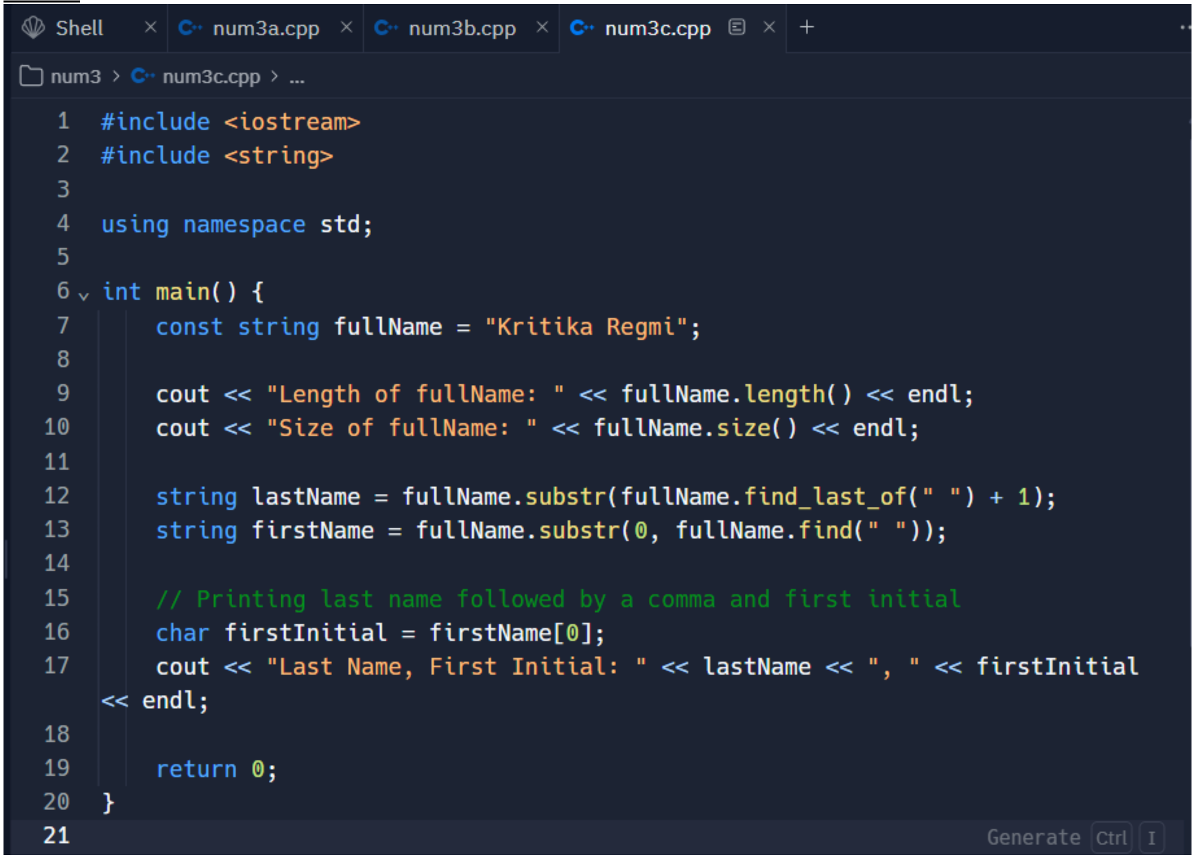


```
~/CSLAB-1/num3$ ls
num3a.cpp  num3b.cpp  output3a
~/CSLAB-1/num3$ g++ num3b.cpp -o output3b
~/CSLAB-1/num3$ ./output3b
Length of fullName: 13
Size of fullName: 13
Given Name: Kritika Regmi
Formatted Name: Regmi, Kritika
~/CSLAB-1/num3$ 
```

c. Question#3: Add the statements necessary to print your last name, followed by a comma and your first initial. Compile and run your program.

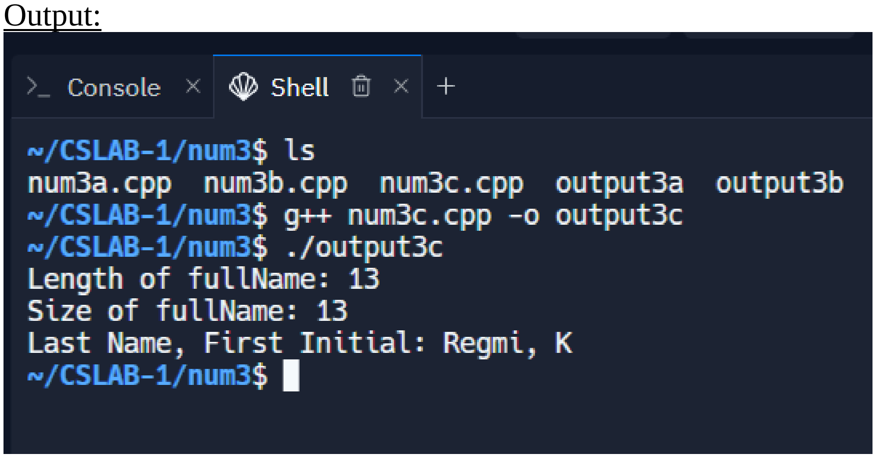
ANS:

Code:



```cpp
#include <iostream>
#include <string>

using namespace std;

int main() {
    const string fullName = "Kritika Regmi";

    cout << "Length of fullName: " << fullName.length() << endl;
    cout << "Size of fullName: " << fullName.size() << endl;

    string lastName = fullName.substr(fullName.find_last_of(" ") + 1);
    string firstName = fullName.substr(0, fullName.find(" "));

    // Printing last name followed by a comma and first initial
    char firstInitial = firstName[0];
    cout << "Last Name, First Initial: " << lastName << ", " << firstInitial << endl;

    return 0;
}
```

Output:

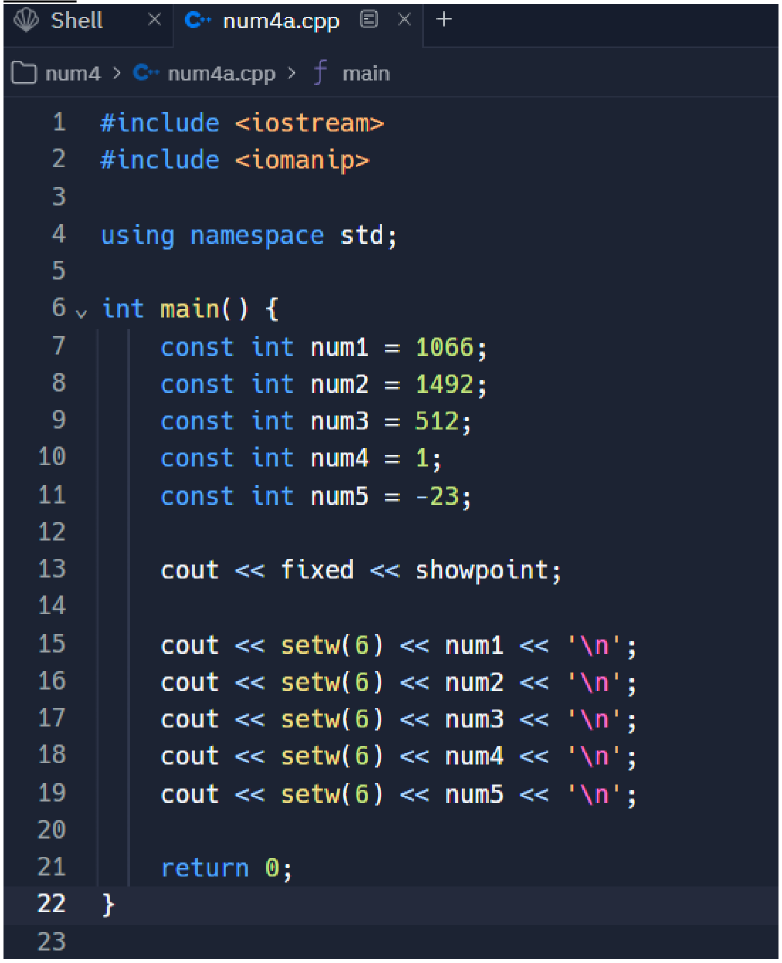


```
~/CSLAB-1/num3$ ls
num3a.cpp  num3b.cpp  num3c.cpp  output3a  output3b
~/CSLAB-1/num3$ g++ num3c.cpp -o output3c
~/CSLAB-1/num3$ ./output3c
Length of fullName: 13
Size of fullName: 13
Last Name, First Initial: Regmi, K
~/CSLAB-1/num3$ 
```

4. Use the following program shell for Question#1 through Question#4.

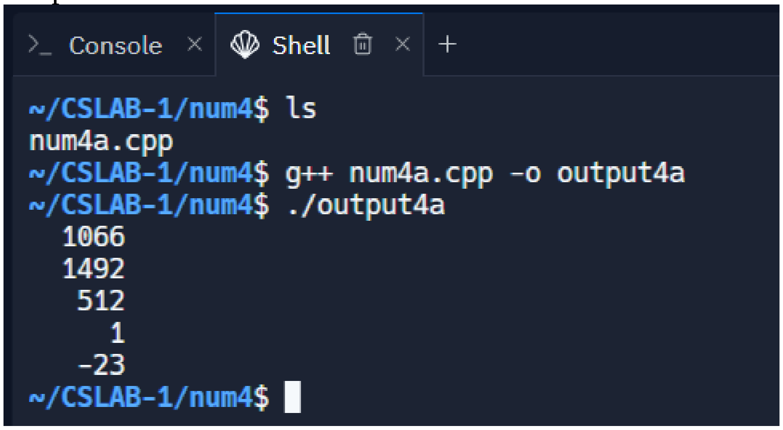
a. Question#1: Write a program to print the following numbers right justified in a column on the screen. Make the values named constants. *1066 1492 512 1 -23*

ANS:



```cpp
#include <iostream>
#include <iomanip>

using namespace std;

int main() {
    const int num1 = 1066;
    const int num2 = 1492;
    const int num3 = 512;
    const int num4 = 1;
    const int num5 = -23;

    cout << fixed << showpoint;

    cout << setw(6) << num1 << '\n';
    cout << setw(6) << num2 << '\n';
    cout << setw(6) << num3 << '\n';
    cout << setw(6) << num4 << '\n';
    cout << setw(6) << num5 << '\n';

    return 0;
}

```

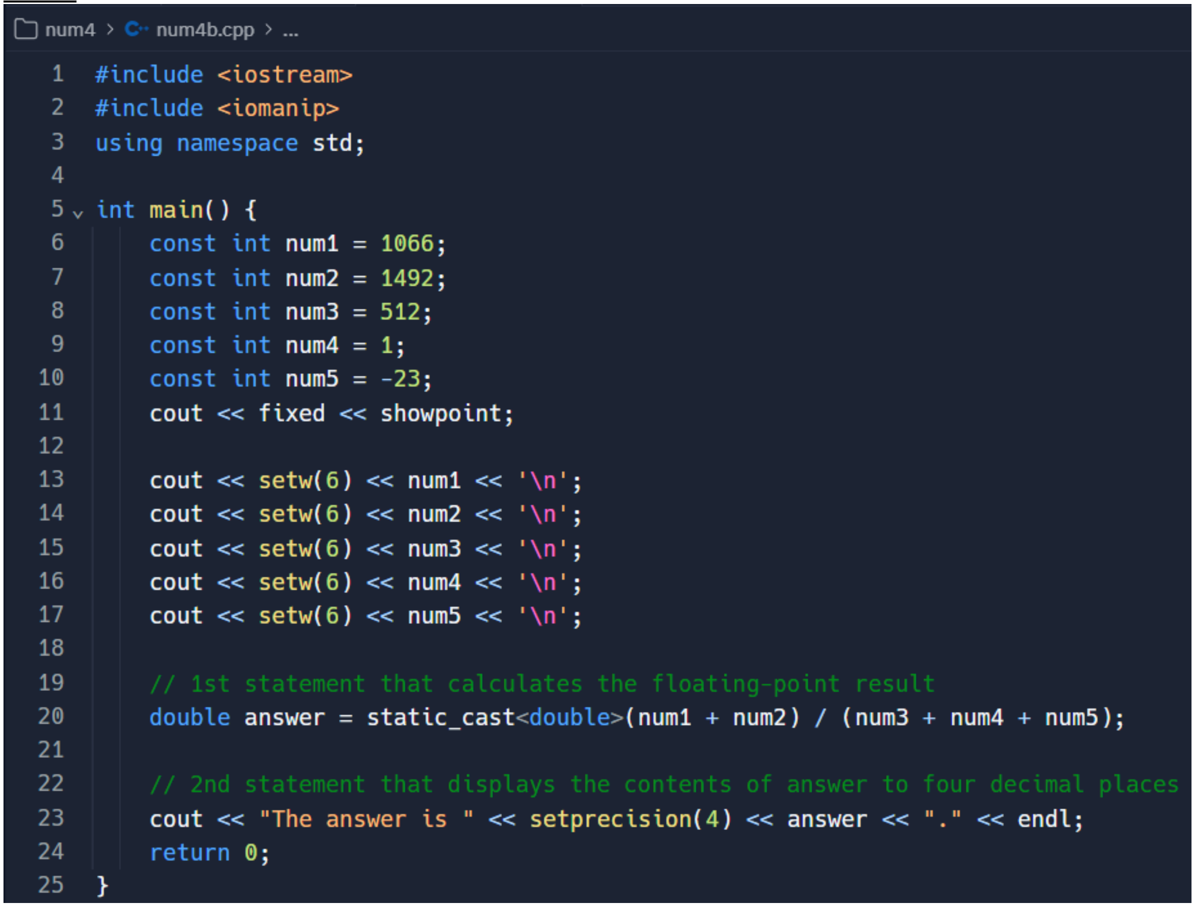
Output:



```
~/CSLAB-1/num4$ ls
num4a.cpp
~/CSLAB-1/num4$ g++ num4a.cpp -o output4a
~/CSLAB-1/num4$ ./output4a
  1066
  1492
   512
     1
   -23
~/CSLAB-1/num4$ 
```

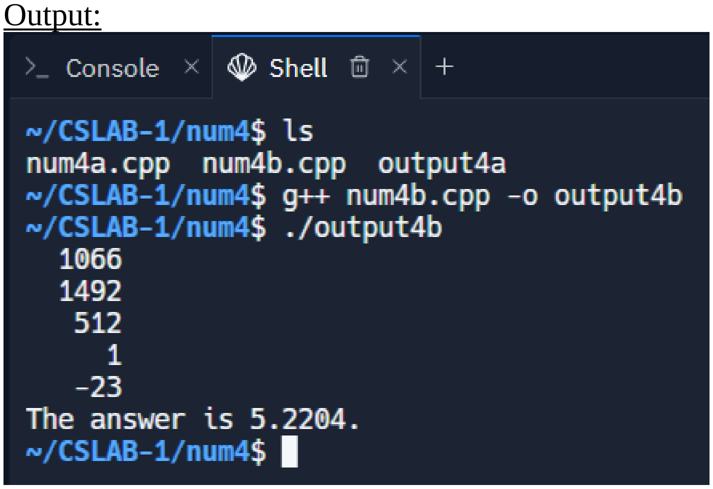
b. Question#2: Add two statements to your program. Calculate the floating-point result from dividing the sum of the first two values by the sum of the last three values and store it in answer. The second statement should write the contents of answer on the screen to four decimal places. (Do not forget to declare *answer*.)

<u>ANS:</u>



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

int main() {
    const int num1 = 1066;
    const int num2 = 1492;
    const int num3 = 512;
    const int num4 = 1;
    const int num5 = -23;
    cout << fixed << showpoint;

    cout << setw(6) << num1 << '\n';
    cout << setw(6) << num2 << '\n';
    cout << setw(6) << num3 << '\n';
    cout << setw(6) << num4 << '\n';
    cout << setw(6) << num5 << '\n';

    // 1st statement that calculates the floating-point result
    double answer = static_cast<double>(num1 + num2) / (num3 + num4 + num5);

    // 2nd statement that displays the contents of answer to four decimal places
    cout << "The answer is " << setprecision(4) << answer << "." << endl;
    return 0;
}
```

<u>Output:</u>

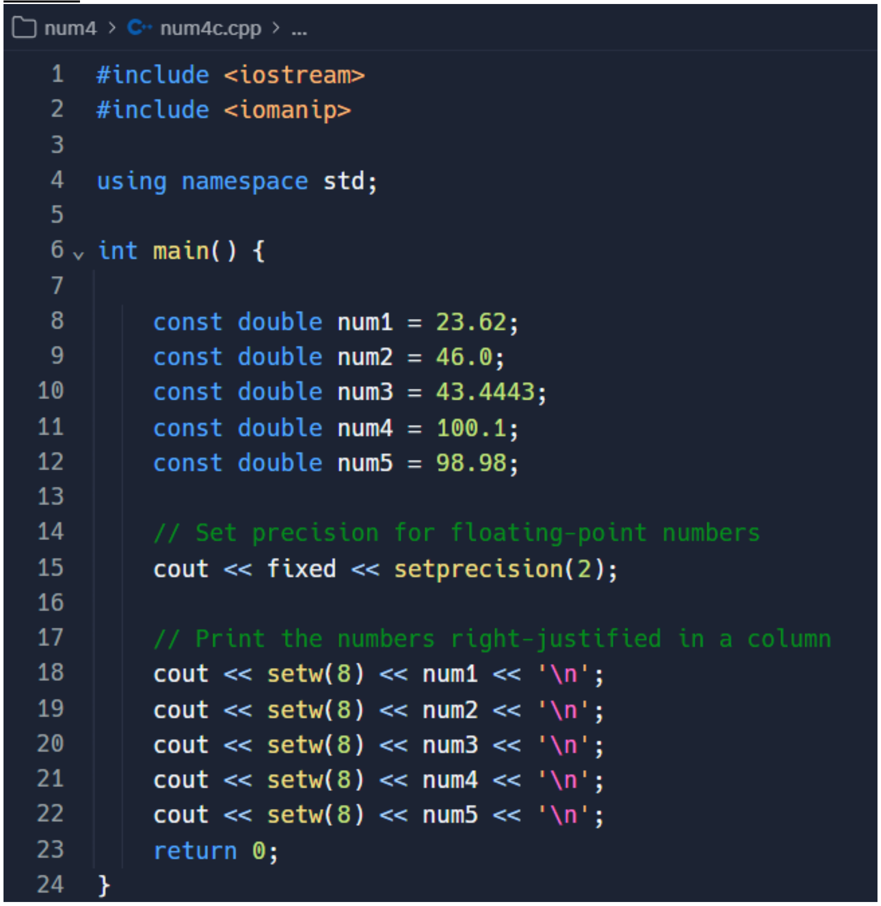


```
~/CSLAB-1/num4$ ls
num4a.cpp  num4b.cpp  output4a
~/CSLAB-1/num4$ g++ num4b.cpp -o output4b
~/CSLAB-1/num4$ ./output4b
  1066
  1492
   512
     1
   -23
The answer is 5.2204.
~/CSLAB-1/num4$ █
```

c. Question#3: Write the following numbers right-justified in a column on the screen. Each of the data values should be written in formatted floating-point notation with two decimal places. Use field width specifications rather than listing the numbers in your program with the proper formatting. You may use either literal constants or named constants.
23.62 46.0 43.4443 100.1 98.98

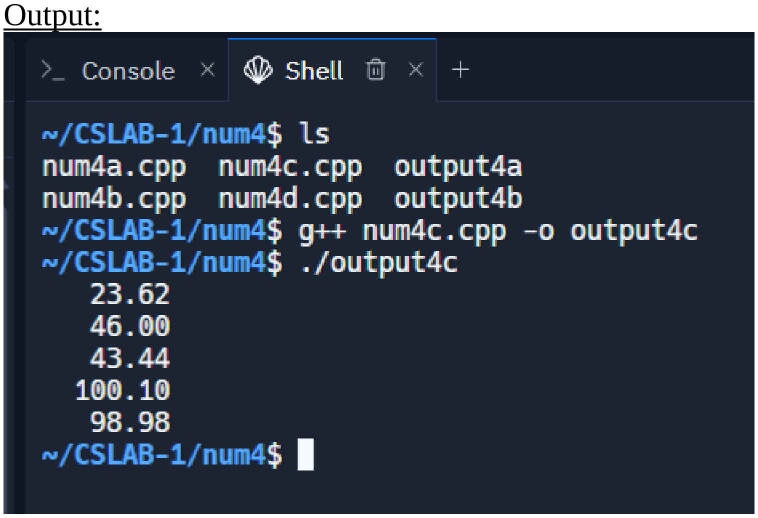
ANS:

Code:



```cpp
#include <iostream>
#include <iomanip>

using namespace std;

int main() {

    const double num1 = 23.62;
    const double num2 = 46.0;
    const double num3 = 43.4443;
    const double num4 = 100.1;
    const double num5 = 98.98;

    // Set precision for floating-point numbers
    cout << fixed << setprecision(2);

    // Print the numbers right-justified in a column
    cout << setw(8) << num1 << '\n';
    cout << setw(8) << num2 << '\n';
    cout << setw(8) << num3 << '\n';
    cout << setw(8) << num4 << '\n';
    cout << setw(8) << num5 << '\n';
    return 0;
}
```

Output:

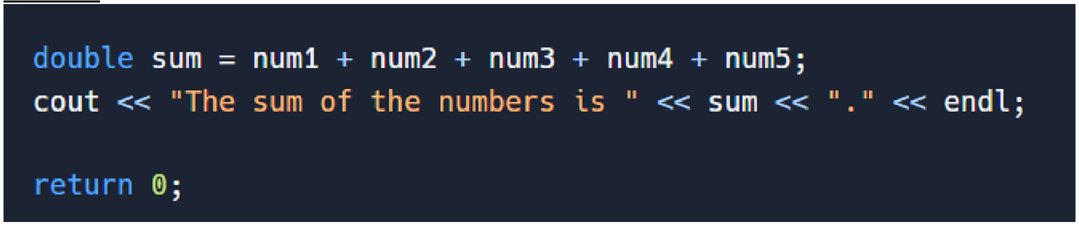


```
~/CSLAB-1/num4$ ls
num4a.cpp  num4c.cpp  output4a
num4b.cpp  num4d.cpp  output4b
~/CSLAB-1/num4$ g++ num4c.cpp -o output4c
~/CSLAB-1/num4$ ./output4c
   23.62
   46.00
   43.44
  100.10
   98.98
~/CSLAB-1/num4$ 
```

d. Question#4: Add two statements to your program for Question#3. The first statement should calculate the sum of the numbers and store the result in variable sum. The second statement should have *sum* on the screen, properly labeled.

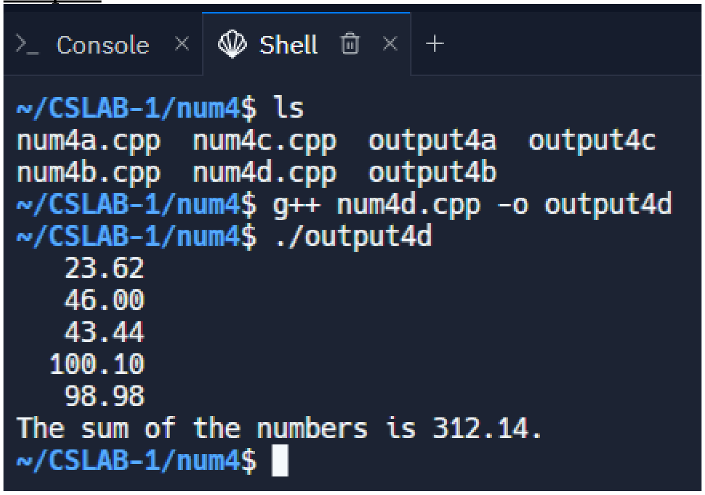
ANS: These two statements are added in the above program and the output is shown below.

CODE:



```
double sum = num1 + num2 + num3 + num4 + num5;
cout << "The sum of the numbers is " << sum << "." << endl;

return 0;
```

Output:



```
~/CSLAB-1/num4$ ls
num4a.cpp  num4c.cpp  output4a  output4c
num4b.cpp  num4d.cpp  output4b
~/CSLAB-1/num4$ g++ num4d.cpp -o output4d
~/CSLAB-1/num4$ ./output4d
   23.62
   46.00
   43.44
  100.10
   98.98
The sum of the numbers is 312.14.
~/CSLAB-1/num4$
```

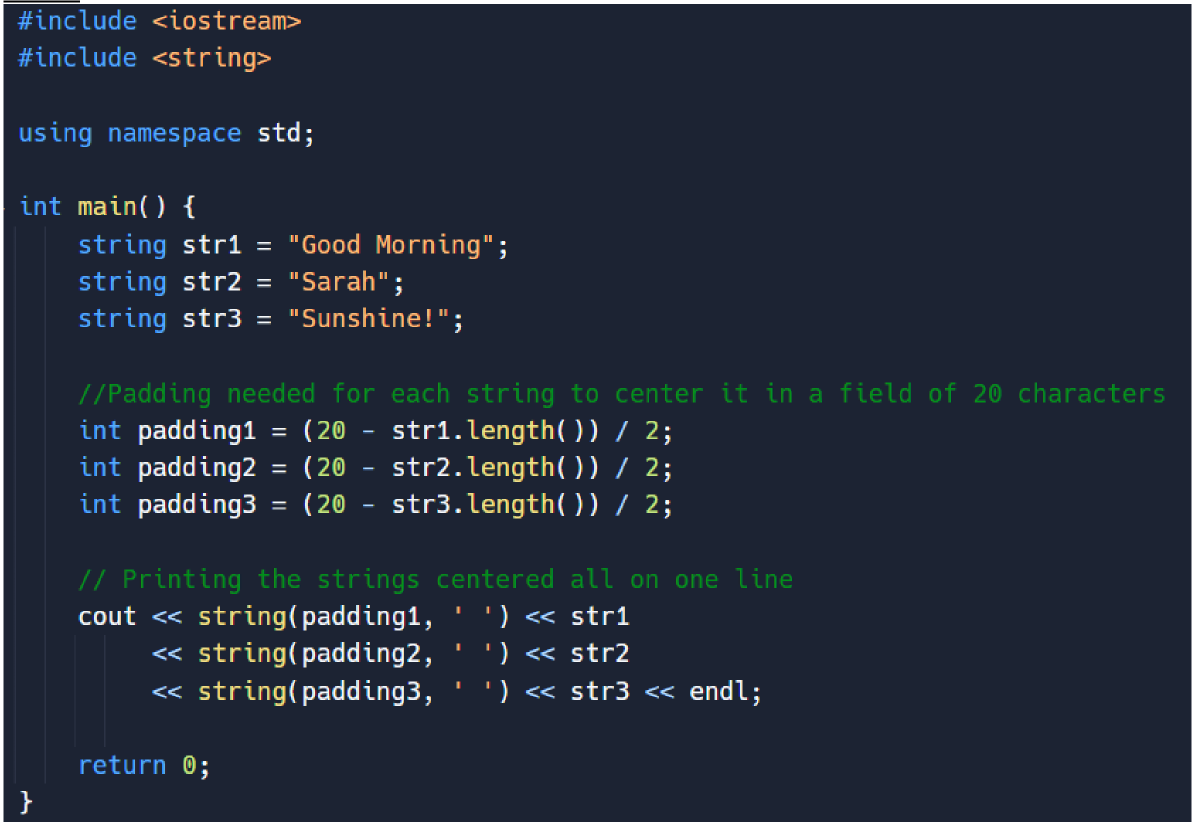
5. Use the following program shell for Question#1through #3.

```
// Program Center sends strings to the output stream in
// specified formats.

#include <iostream>
#include <iomanip>

using std::cout;

int main (void){
return 0;
}
```

a. Question#1: Add the statements necessary to print the following strings centered in fields of *20* characters, all on one line: *"Good Morning"*, *"Sarah"*, and *"Sunshine!"*. Do not use manipulators. Compile and run your program; show your output.

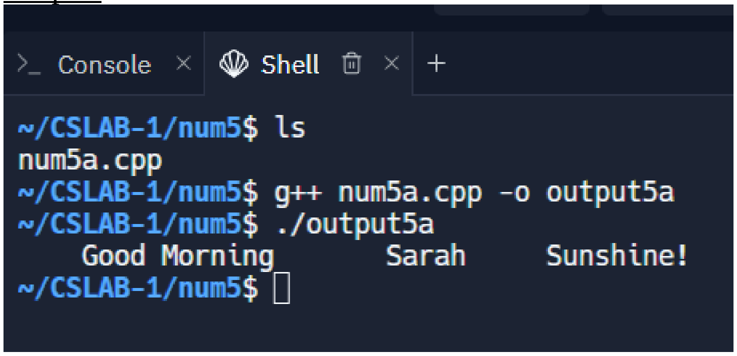
ANS:

Code:



```cpp
#include <iostream>
#include <string>

using namespace std;

int main() {
    string str1 = "Good Morning";
    string str2 = "Sarah";
    string str3 = "Sunshine!";

    //Padding needed for each string to center it in a field of 20 characters
    int padding1 = (20 - str1.length()) / 2;
    int padding2 = (20 - str2.length()) / 2;
    int padding3 = (20 - str3.length()) / 2;

    // Printing the strings centered all on one line
    cout << string(padding1, ' ') << str1
         << string(padding2, ' ') << str2
         << string(padding3, ' ') << str3 << endl;

    return 0;
}
```

Output:

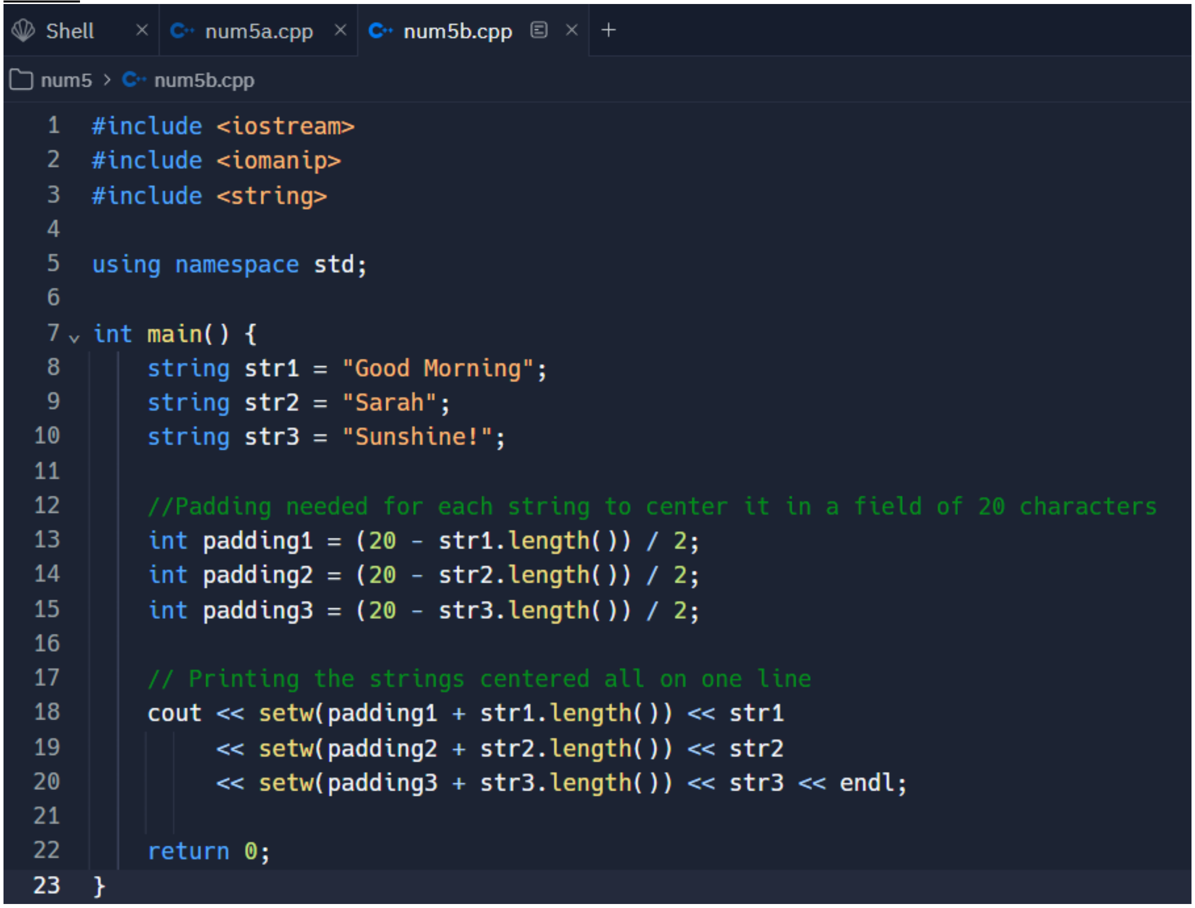


```
~/CSLAB-1/num5$ ls
num5a.cpp
~/CSLAB-1/num5$ g++ num5a.cpp -o output5a
~/CSLAB-1/num5$ ./output5a
    Good Morning       Sarah     Sunshine!
~/CSLAB-1/num5$ 
```

b. Question#2: Repeat Question#1using manipulators to help center your strings. Compile and run your program. Your output should be the same.

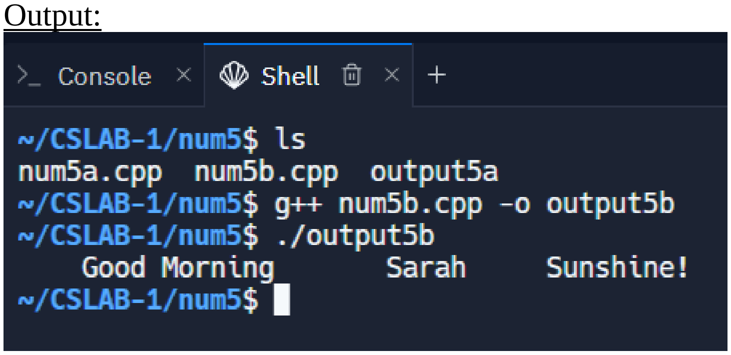
ANS:

Code:



```cpp
#include <iostream>
#include <iomanip>
#include <string>

using namespace std;

int main() {
    string str1 = "Good Morning";
    string str2 = "Sarah";
    string str3 = "Sunshine!";

    //Padding needed for each string to center it in a field of 20 characters
    int padding1 = (20 - str1.length()) / 2;
    int padding2 = (20 - str2.length()) / 2;
    int padding3 = (20 - str3.length()) / 2;

    // Printing the strings centered all on one line
    cout << setw(padding1 + str1.length()) << str1
         << setw(padding2 + str2.length()) << str2
         << setw(padding3 + str3.length()) << str3 << endl;

    return 0;
}
```

Output:

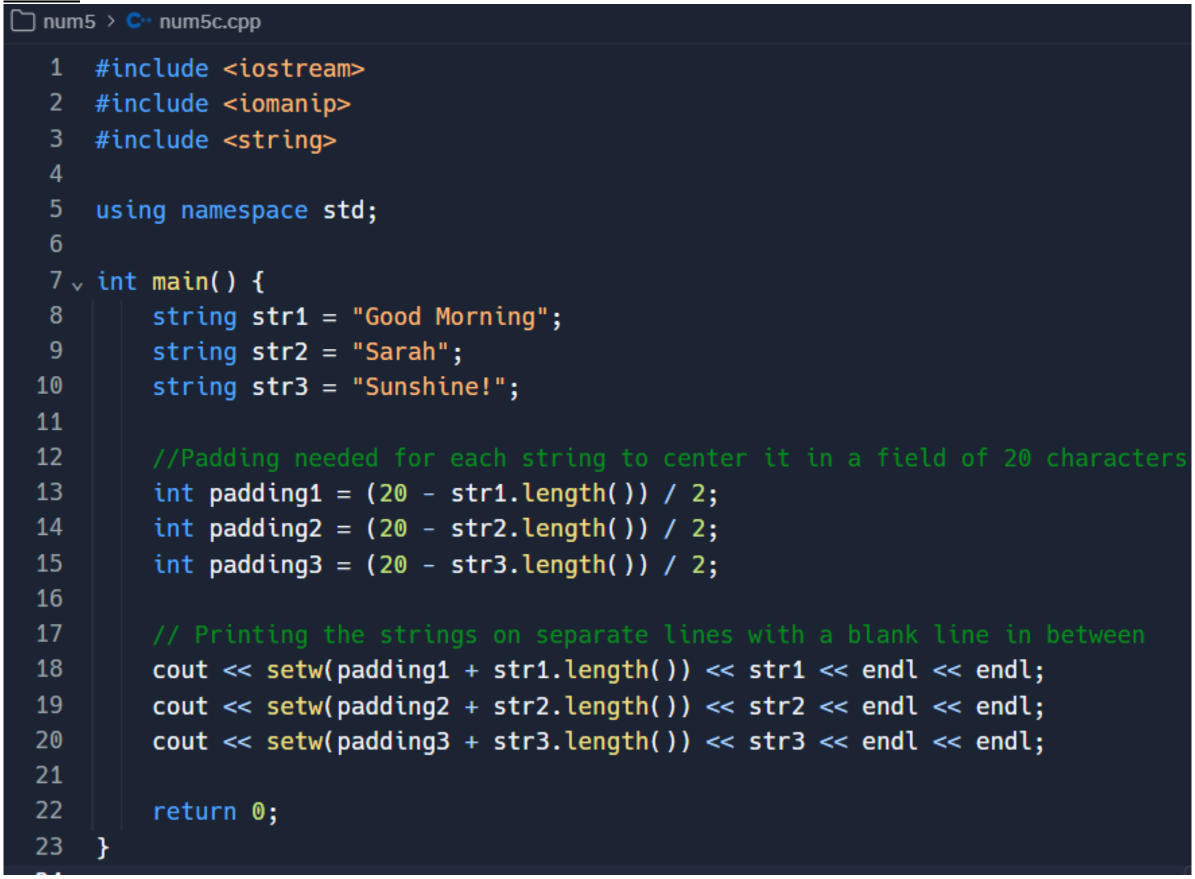


```
~/CSLAB-1/num5$ ls
num5a.cpp  num5b.cpp  output5a
~/CSLAB-1/num5$ g++ num5b.cpp -o output5b
~/CSLAB-1/num5$ ./output5b
    Good Morning       Sarah     Sunshine!
~/CSLAB-1/num5$ 
```

c. Question#3: Change the program in Question#2 so that the three strings are printed on three separate lines with a blank line in between each string.

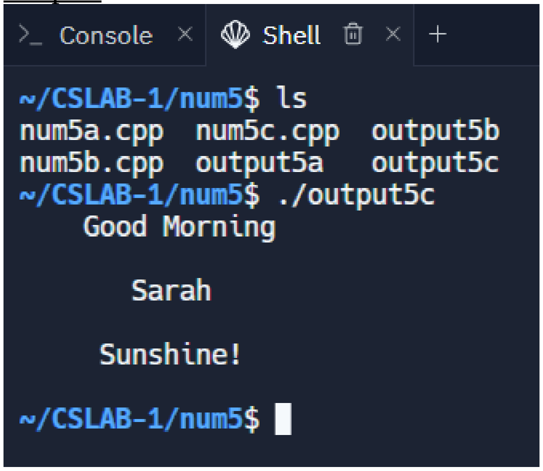
ANS:

Code:



```cpp
#include <iostream>
#include <iomanip>
#include <string>

using namespace std;

int main() {
    string str1 = "Good Morning";
    string str2 = "Sarah";
    string str3 = "Sunshine!";

    //Padding needed for each string to center it in a field of 20 characters
    int padding1 = (20 - str1.length()) / 2;
    int padding2 = (20 - str2.length()) / 2;
    int padding3 = (20 - str3.length()) / 2;

    // Printing the strings on separate lines with a blank line in between
    cout << setw(padding1 + str1.length()) << str1 << endl << endl;
    cout << setw(padding2 + str2.length()) << str2 << endl << endl;
    cout << setw(padding3 + str3.length()) << str3 << endl << endl;

    return 0;
}
```

Output:



```
~/CSLAB-1/num5$ ls
num5a.cpp  num5c.cpp  output5b
num5b.cpp  output5a   output5c
~/CSLAB-1/num5$ ./output5c
    Good Morning

       Sarah

     Sunshine!

~/CSLAB-1/num5$ 
```