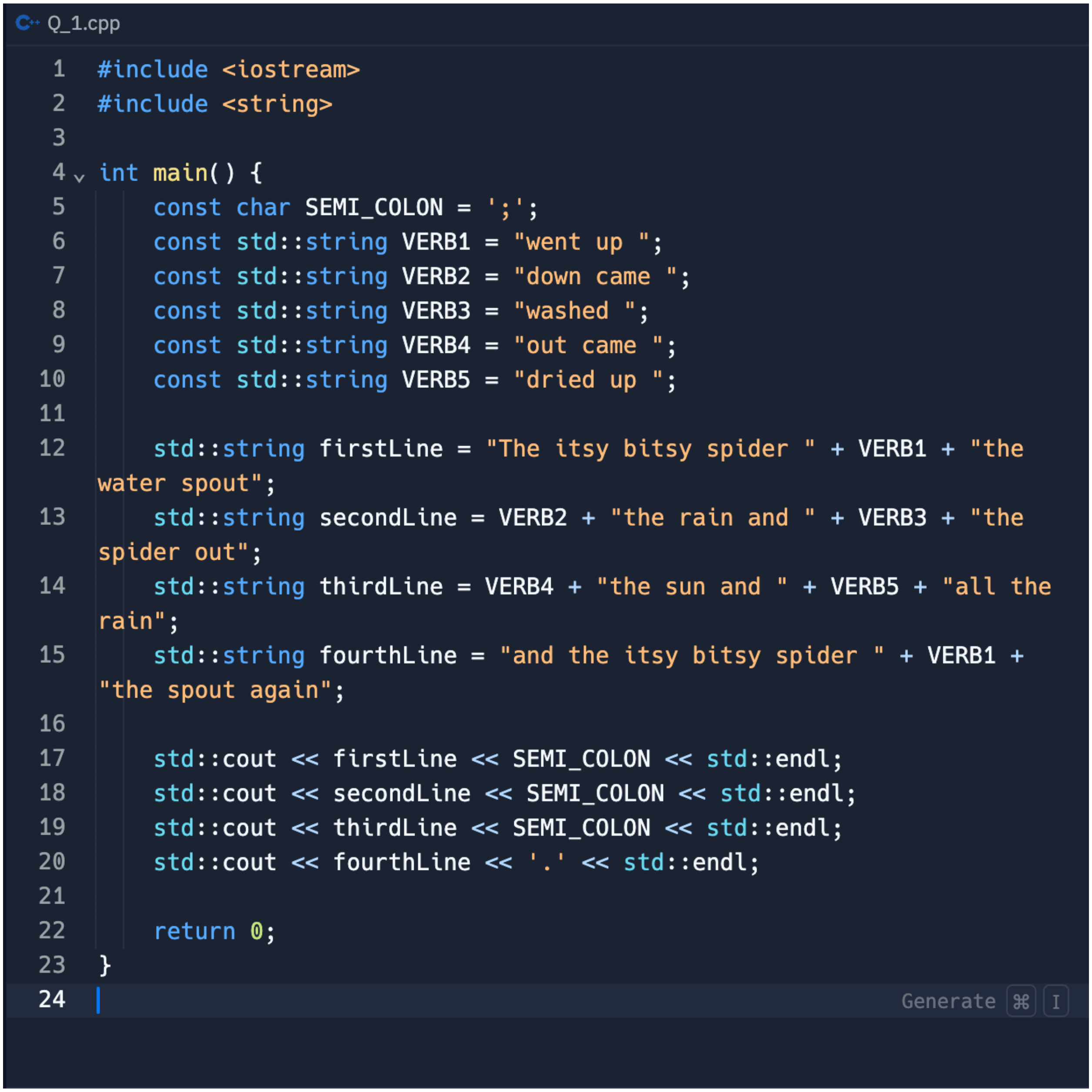
Question no.1

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

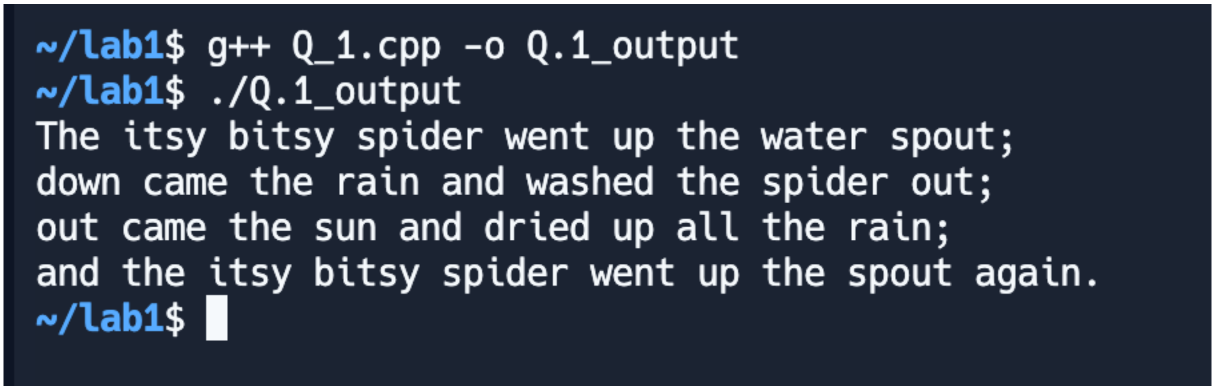
- ⇨ Answer:

  Code:



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

int main() {
    const char SEMI_COLON = ';';
    const std::string VERB1 = "went up ";
    const std::string VERB2 = "down came ";
    const std::string VERB3 = "washed ";
    const std::string VERB4 = "out came ";
    const std::string VERB5 = "dried up ";

    std::string firstLine = "The itsy bitsy spider " + VERB1 + "the water spout";
    std::string secondLine = VERB2 + "the rain and " + VERB3 + "the spider out";
    std::string thirdLine = VERB4 + "the sun and " + VERB5 + "all the rain";
    std::string fourthLine = "and the itsy bitsy spider " + VERB1 + "the spout again";

    std::cout << firstLine << SEMI_COLON << std::endl;
    std::cout << secondLine << SEMI_COLON << std::endl;
    std::cout << thirdLine << SEMI_COLON << std::endl;
    std::cout << fourthLine << '.' << std::endl;

    return 0;
}
```

Output:



```
~/lab1$ g++ Q_1.cpp -o Q.1_output
~/lab1$ ./Q.1_output
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.
~/lab1$
```

Question no.2

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

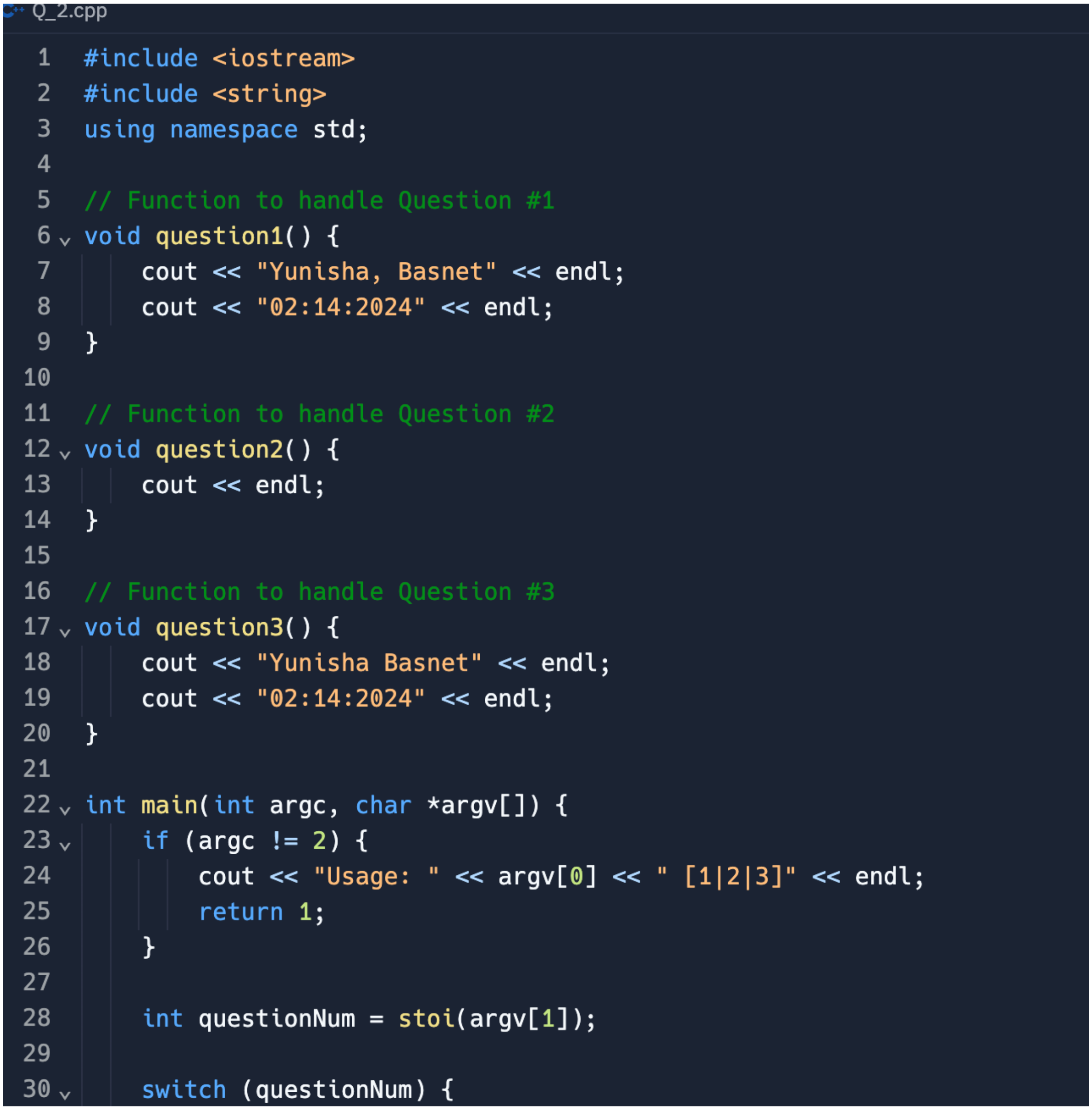
```
// Program Shell

#include <iostream>

using namespace std;

int main (){

return 0;
}
```

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)

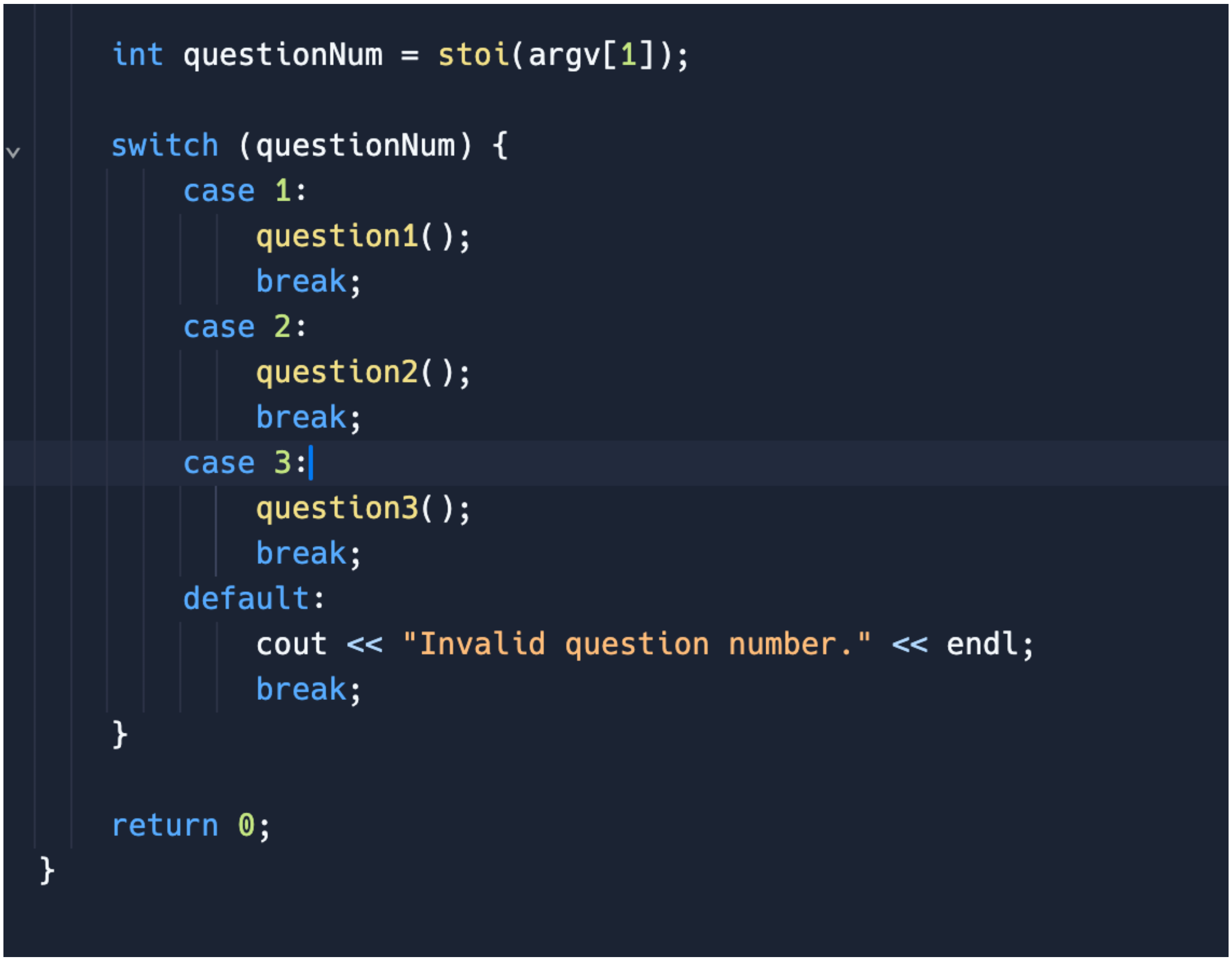
b. Question#2: Change your program so that there is a space between the two lines of output.

c. Question#3: Change your program so that your first name is printed followed by your last name, with a blank in between the name.

⇨ Answer:
Code:

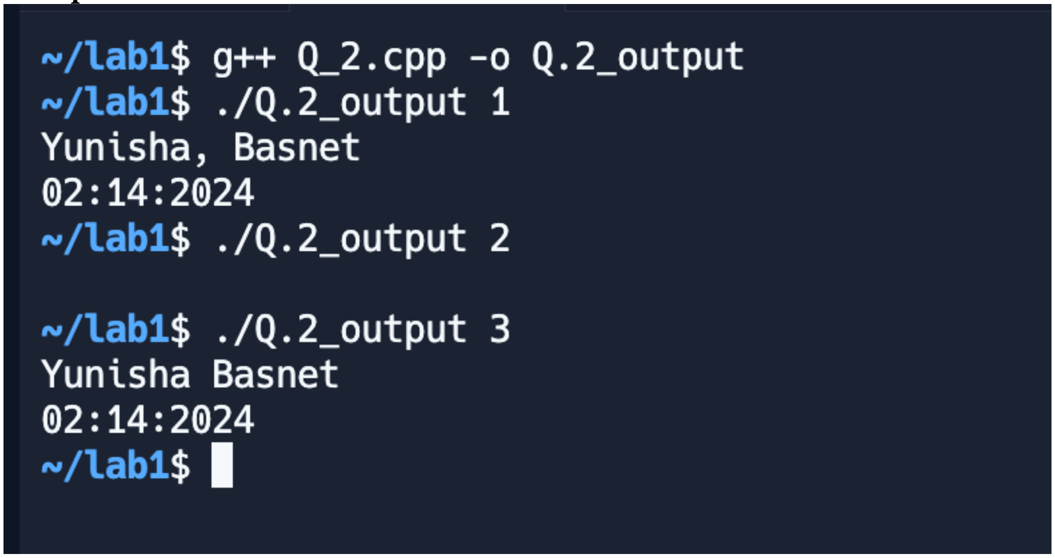


```
#include <iostream>
#include <string>
using namespace std;

// Function to handle Question #1
void question1() {
    cout << "Yunisha, Basnet" << endl;
    cout << "02:14:2024" << endl;
}

// Function to handle Question #2
void question2() {
    cout << endl;
}

// Function to handle Question #3
void question3() {
    cout << "Yunisha Basnet" << endl;
    cout << "02:14:2024" << endl;
}

int main(int argc, char *argv[]) {
    if (argc != 2) {
        cout << "Usage: " << argv[0] << " [1|2|3]" << endl;
        return 1;
    }

    int questionNum = stoi(argv[1]);

    switch (questionNum) {
```



```cpp
    int questionNum = stoi(argv[1]);

    switch (questionNum) {
        case 1:
            question1();
            break;
        case 2:
            question2();
            break;
        case 3:
            question3();
            break;
        default:
            cout << "Invalid question number." << endl;
            break;
    }

    return 0;
}
```

Output:



```
~/lab1$ g++ Q_2.cpp -o Q.2_output
~/lab1$ ./Q.2_output 1
Yunisha, Basnet
02:14:2024
~/lab1$ ./Q.2_output 2

~/lab1$ ./Q.2_output 3
Yunisha Basnet
02:14:2024
~/lab1$ 
```

Question no.3

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

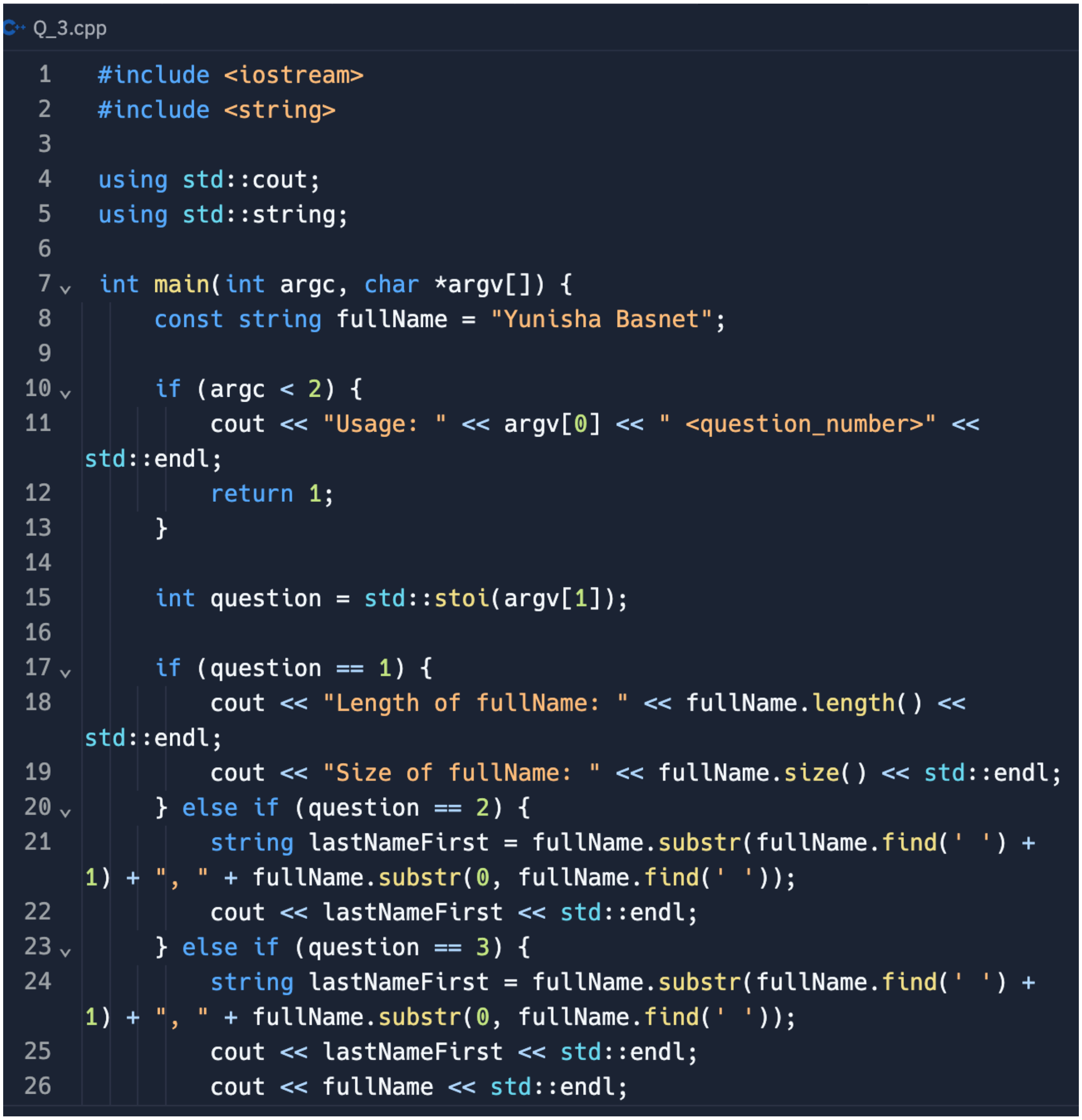
```
// Program Strings applies string functions.

#include <iostream>

using std::cout, std::string;

int main (void){

return 0;
}
```

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.

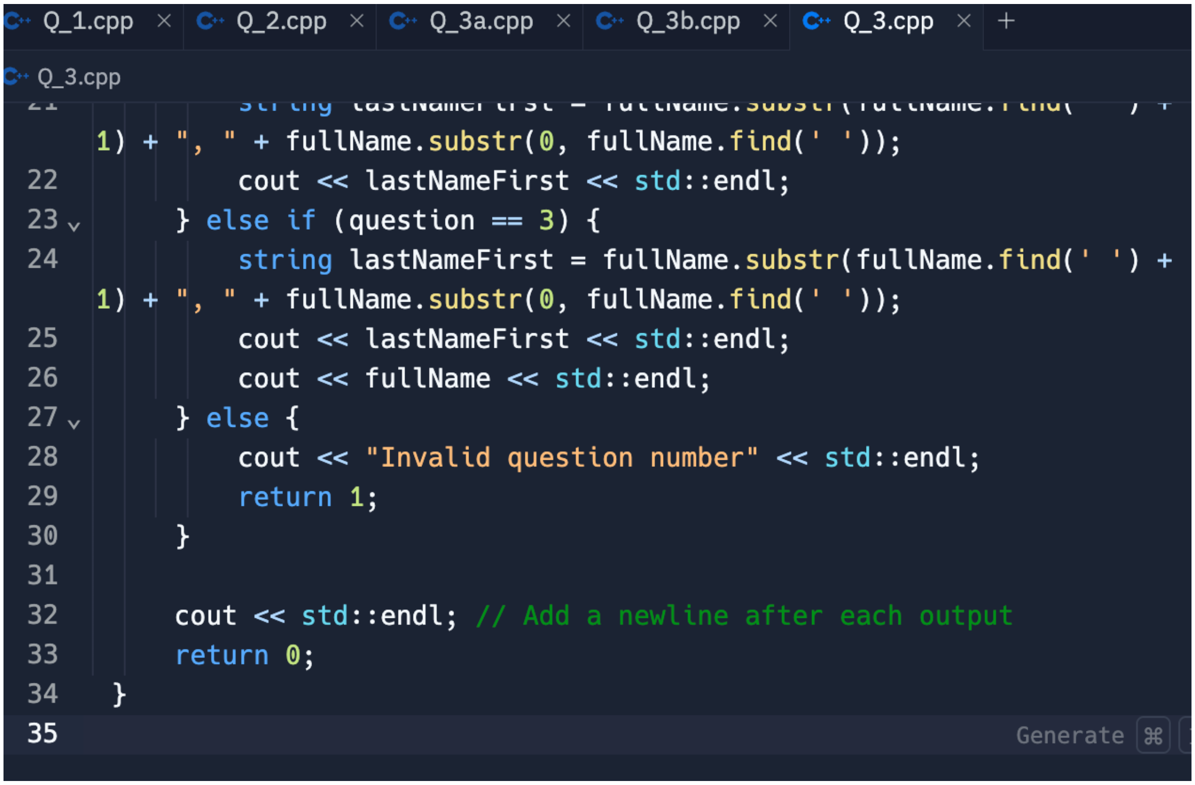
b. Question#2: Add statements to your Question#1 program to print your name formatted as last name first, followed by a comma and your first name. Use function substr to accomplish this task. Compile and run your program.

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.

⇨ Answer:
Code:

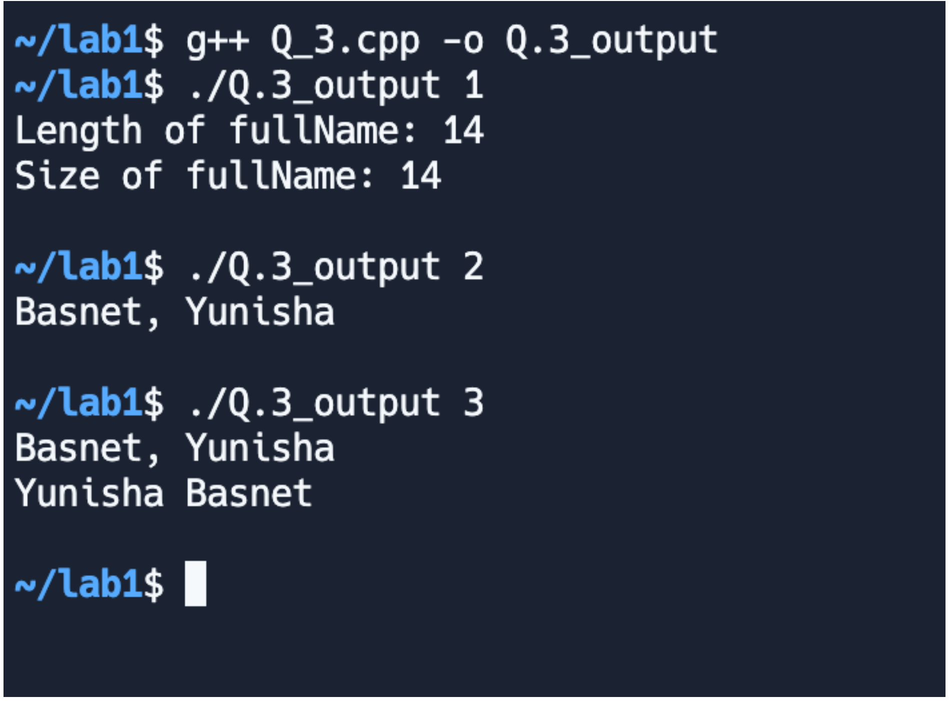


```cpp
#include <iostream>
#include <string>

using std::cout;
using std::string;

int main(int argc, char *argv[]) {
    const string fullName = "Yunisha Basnet";

    if (argc < 2) {
        cout << "Usage: " << argv[0] << " <question_number>" << std::endl;
        return 1;
    }

    int question = std::stoi(argv[1]);

    if (question == 1) {
        cout << "Length of fullName: " << fullName.length() << std::endl;
        cout << "Size of fullName: " << fullName.size() << std::endl;
    } else if (question == 2) {
        string lastNameFirst = fullName.substr(fullName.find(' ') + 1) + ", " + fullName.substr(0, fullName.find(' '));
        cout << lastNameFirst << std::endl;
    } else if (question == 3) {
        string lastNameFirst = fullName.substr(fullName.find(' ') + 1) + ", " + fullName.substr(0, fullName.find(' '));
        cout << lastNameFirst << std::endl;
        cout << fullName << std::endl;
```



```
        string lastNameFirst = fullName.substr(fullName.find(' ') +
1) + ", " + fullName.substr(0, fullName.find(' '));
        cout << lastNameFirst << std::endl;
    } else if (question == 3) {
        string lastNameFirst = fullName.substr(fullName.find(' ') +
1) + ", " + fullName.substr(0, fullName.find(' '));
        cout << lastNameFirst << std::endl;
        cout << fullName << std::endl;
    } else {
        cout << "Invalid question number" << std::endl;
        return 1;
    }

    cout << std::endl; // Add a newline after each output
    return 0;
}
```

Output:



```
~/lab1$ g++ Q_3.cpp -o Q.3_output
~/lab1$ ./Q.3_output 1
Length of fullName: 14
Size of fullName: 14

~/lab1$ ./Q.3_output 2
Basnet, Yunisha

~/lab1$ ./Q.3_output 3
Basnet, Yunisha
Yunisha Basnet

~/lab1$ 
```

Question no.4

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

```
// Program Numbers sends numbers to the output stream in
// specified formats.

#include <iostream>
#include <iomanip>

using std::cout;

int main (void){

cout << fixed  << showpoint;
return 0;
}
```

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

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.)

   The answer is ______________.

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.

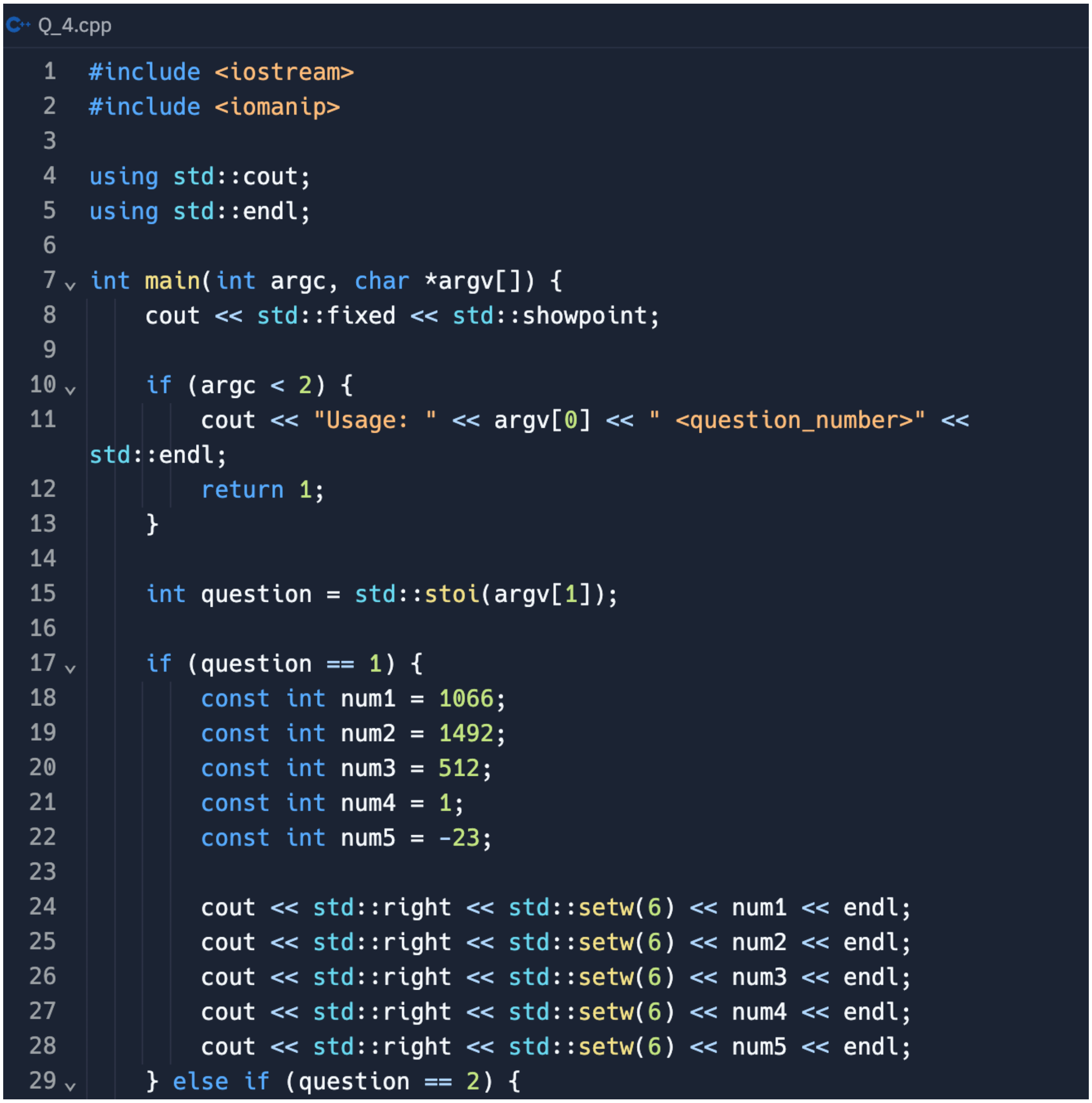
Use field width specifications rather than listing the numbers in your program with the proper formatting. You may use either literal constants or named constants.

23.62 46.0 43.4443 100.1 98.98

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 write sum on the screen, properly labeled.

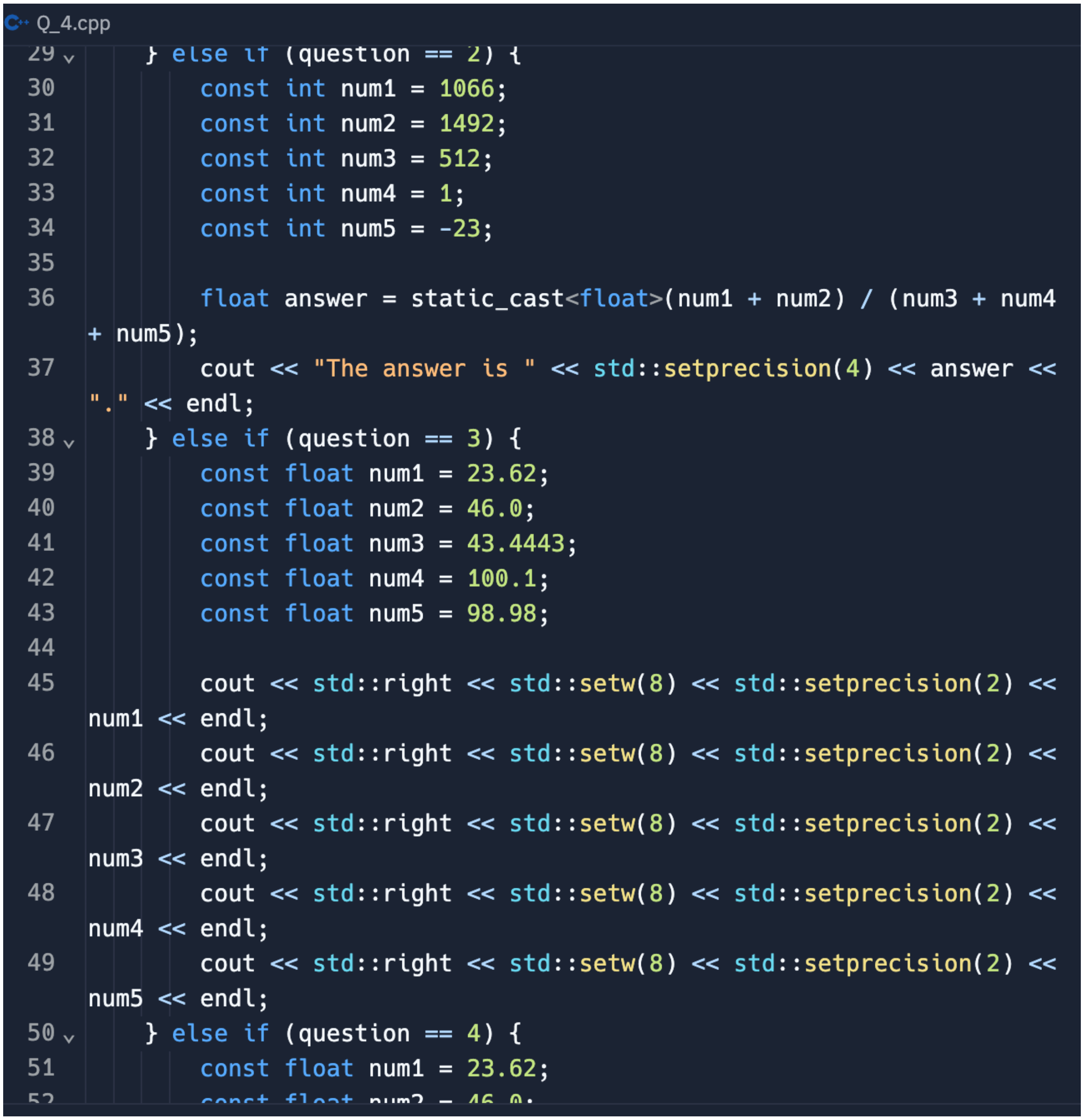
The sum of the numbers is ________________.

⇨ Answer:
Code:

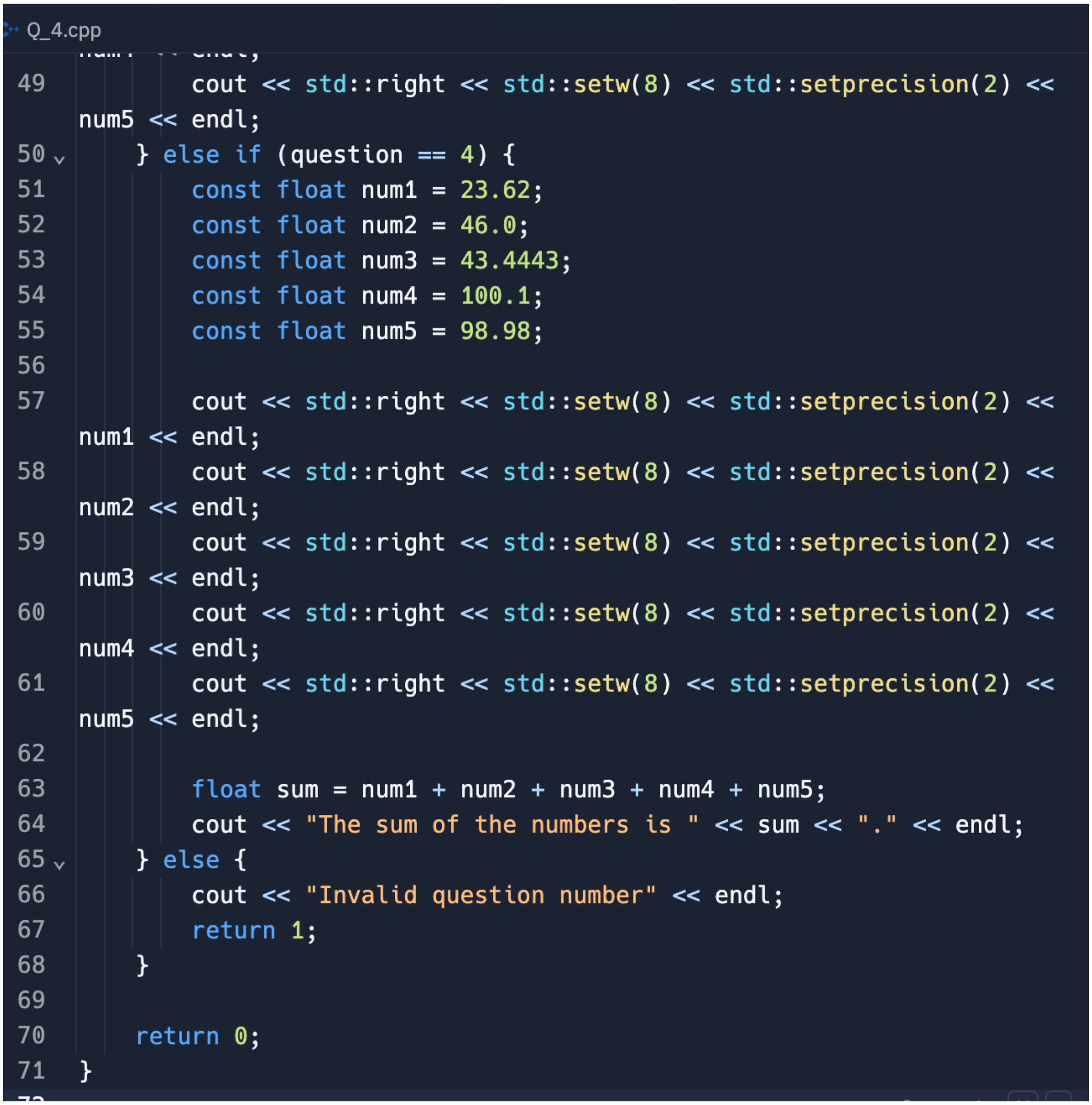


Q_4.cpp

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

using std::cout;
using std::endl;

int main(int argc, char *argv[]) {
    cout << std::fixed << std::showpoint;

    if (argc < 2) {
        cout << "Usage: " << argv[0] << " <question_number>" << std::endl;
        return 1;
    }

    int question = std::stoi(argv[1]);

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

        cout << std::right << std::setw(6) << num1 << endl;
        cout << std::right << std::setw(6) << num2 << endl;
        cout << std::right << std::setw(6) << num3 << endl;
        cout << std::right << std::setw(6) << num4 << endl;
        cout << std::right << std::setw(6) << num5 << endl;
    } else if (question == 2) {
```

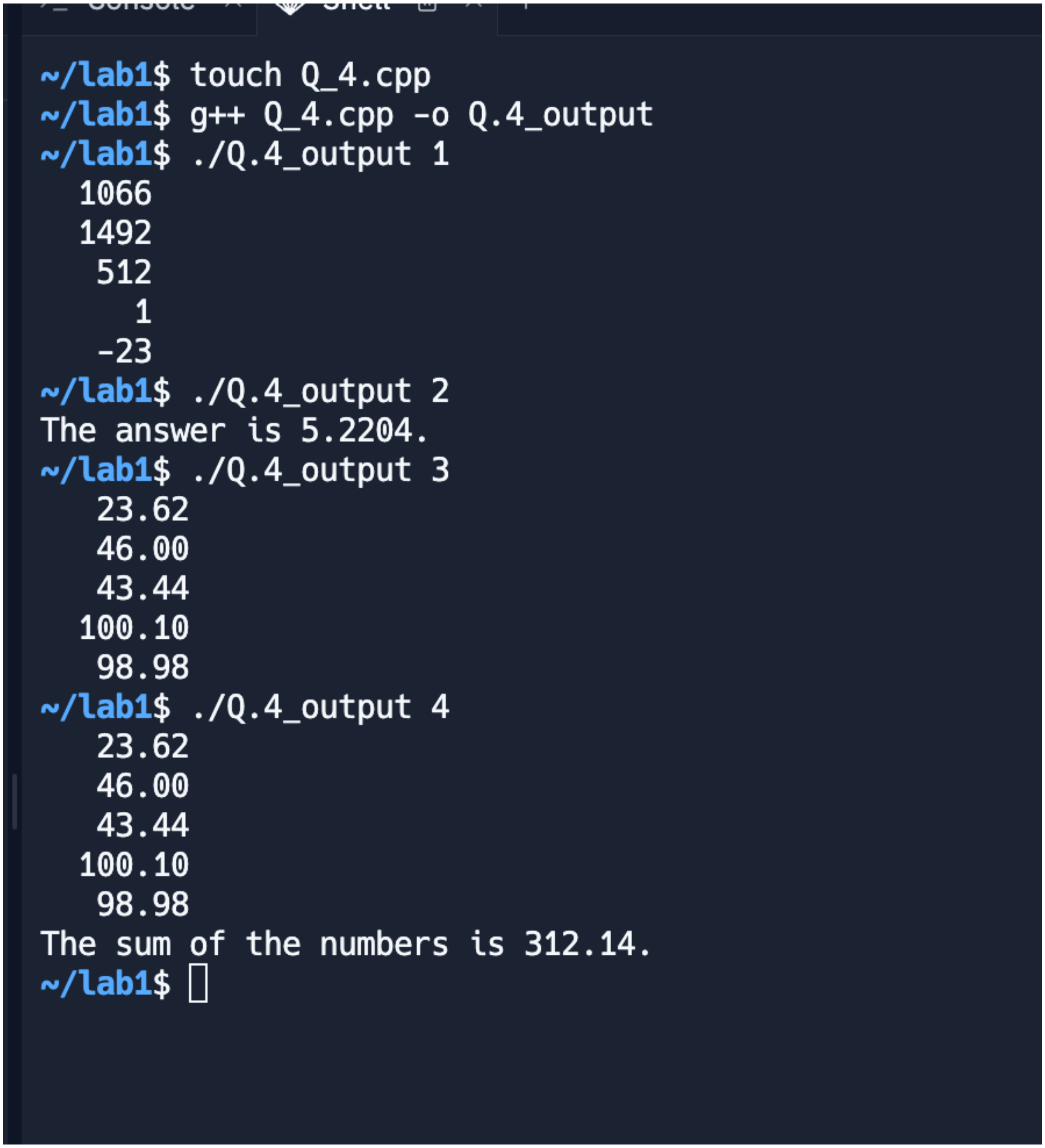


```cpp
        } else if (question == 2) {
            const int num1 = 1066;
            const int num2 = 1492;
            const int num3 = 512;
            const int num4 = 1;
            const int num5 = -23;

            float answer = static_cast<float>(num1 + num2) / (num3 + num4 + num5);
            cout << "The answer is " << std::setprecision(4) << answer << "." << endl;
        } else if (question == 3) {
            const float num1 = 23.62;
            const float num2 = 46.0;
            const float num3 = 43.4443;
            const float num4 = 100.1;
            const float num5 = 98.98;

            cout << std::right << std::setw(8) << std::setprecision(2) << num1 << endl;
            cout << std::right << std::setw(8) << std::setprecision(2) << num2 << endl;
            cout << std::right << std::setw(8) << std::setprecision(2) << num3 << endl;
            cout << std::right << std::setw(8) << std::setprecision(2) << num4 << endl;
            cout << std::right << std::setw(8) << std::setprecision(2) << num5 << endl;
        } else if (question == 4) {
            const float num1 = 23.62;
            const float num2 = 46.0;
```



```cpp
num4 << endl;
        cout << std::right << std::setw(8) << std::setprecision(2) << num5 << endl;
    } else if (question == 4) {
        const float num1 = 23.62;
        const float num2 = 46.0;
        const float num3 = 43.4443;
        const float num4 = 100.1;
        const float num5 = 98.98;

        cout << std::right << std::setw(8) << std::setprecision(2) << num1 << endl;
        cout << std::right << std::setw(8) << std::setprecision(2) << num2 << endl;
        cout << std::right << std::setw(8) << std::setprecision(2) << num3 << endl;
        cout << std::right << std::setw(8) << std::setprecision(2) << num4 << endl;
        cout << std::right << std::setw(8) << std::setprecision(2) << num5 << endl;

        float sum = num1 + num2 + num3 + num4 + num5;
        cout << "The sum of the numbers is " << sum << "." << endl;
    } else {
        cout << "Invalid question number" << endl;
        return 1;
    }

    return 0;
}
```

Output:



```
~/lab1$ touch Q_4.cpp
~/lab1$ g++ Q_4.cpp -o Q.4_output
~/lab1$ ./Q.4_output 1
  1066
  1492
   512
     1
   -23
~/lab1$ ./Q.4_output 2
The answer is 5.2204.
~/lab1$ ./Q.4_output 3
   23.62
   46.00
   43.44
  100.10
   98.98
~/lab1$ ./Q.4_output 4
   23.62
   46.00
   43.44
  100.10
   98.98
The sum of the numbers is 312.14.
~/lab1$ 
```

Question no.5
Use the following program shell for Question#1through #3.

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

#include <iostream>
```

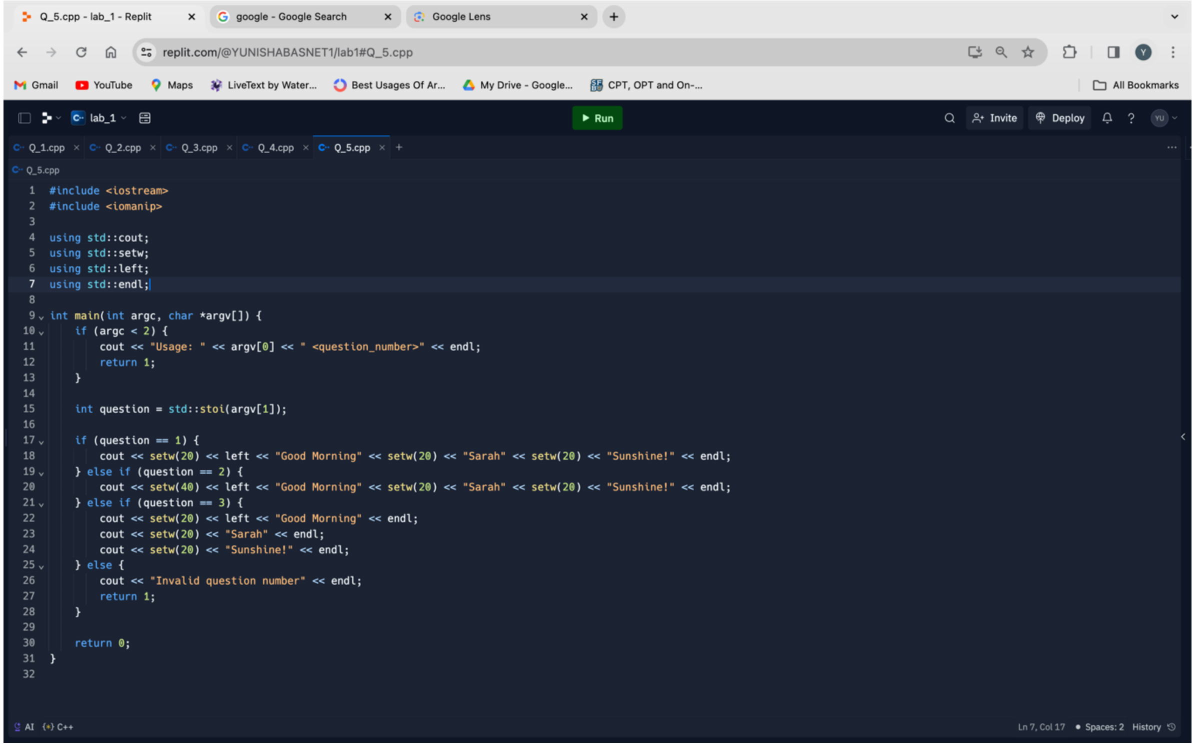
```
#include <iomanip>

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.

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

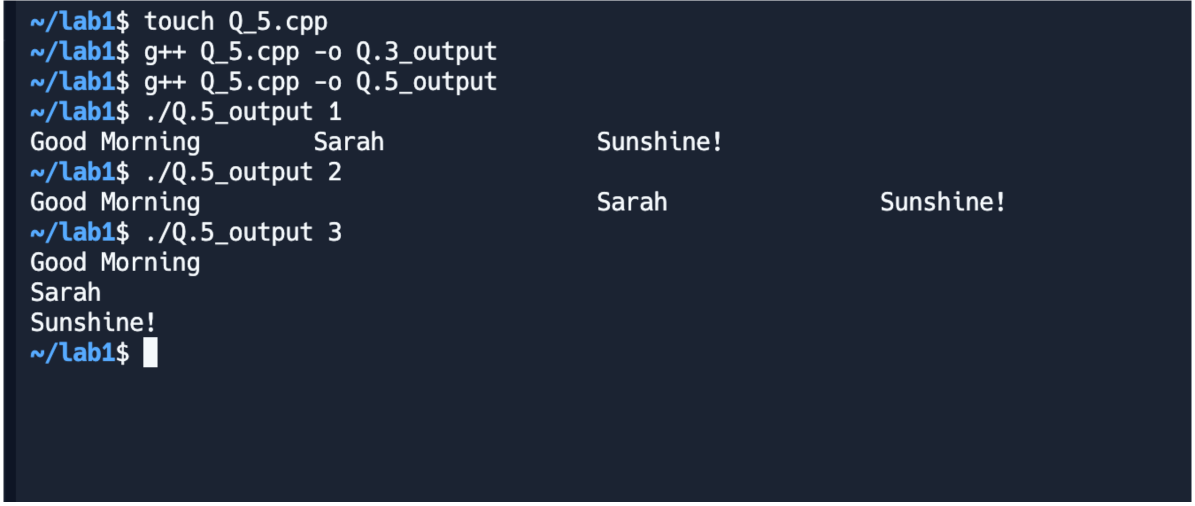
c. Question#3: Change the program in Question#2 so that the three strings are printed on three separate lines with a blank line in between each string.

⇨ Answer:
Code:



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

using std::cout;
using std::setw;
using std::left;
using std::endl;

int main(int argc, char *argv[]) {
    if (argc < 2) {
        cout << "Usage: " << argv[0] << " <question_number>" << endl;
        return 1;
    }

    int question = std::stoi(argv[1]);

    if (question == 1) {
        cout << setw(20) << left << "Good Morning" << setw(20) << "Sarah" << setw(20) << "Sunshine!" << endl;
    } else if (question == 2) {
        cout << setw(40) << left << "Good Morning" << setw(20) << "Sarah" << setw(20) << "Sunshine!" << endl;
    } else if (question == 3) {
        cout << setw(20) << left << "Good Morning" << endl;
        cout << setw(20) << "Sarah" << endl;
        cout << setw(20) << "Sunshine!" << endl;
    } else {
        cout << "Invalid question number" << endl;
        return 1;
    }

    return 0;
}
```

Output:



```
~/lab1$ touch Q_5.cpp
~/lab1$ g++ Q_5.cpp -o Q.3_output
~/lab1$ g++ Q_5.cpp -o Q.5_output
~/lab1$ ./Q.5_output 1
Good Morning        Sarah               Sunshine!
~/lab1$ ./Q.5_output 2
Good Morning                            Sarah               Sunshine!
~/lab1$ ./Q.5_output 3
Good Morning
Sarah
Sunshine!
~/lab1$ 
```