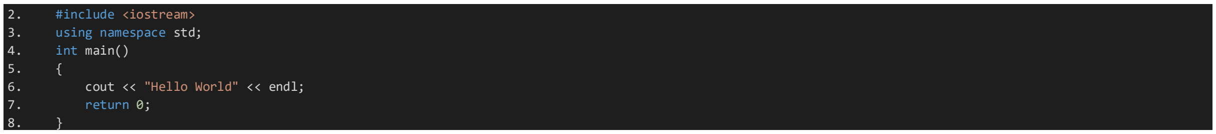
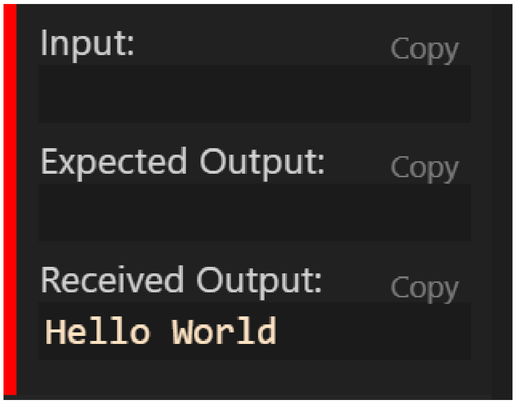
1. Write a program to print “Hello World” on the screen.



```
#include <iostream>
using namespace std;
int main()
{
    cout << "Hello World" << endl;
    return 0;
}
```

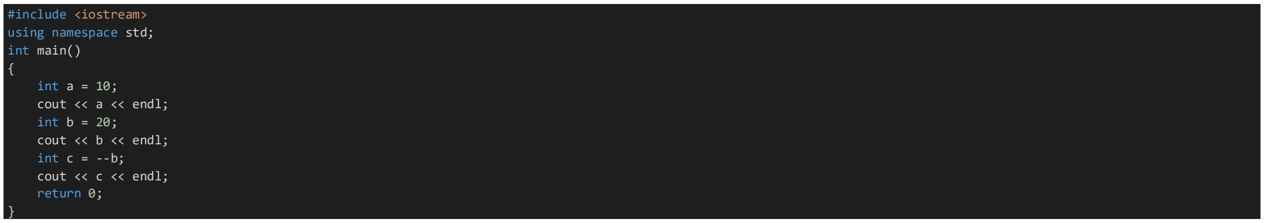


```
Hello World
```

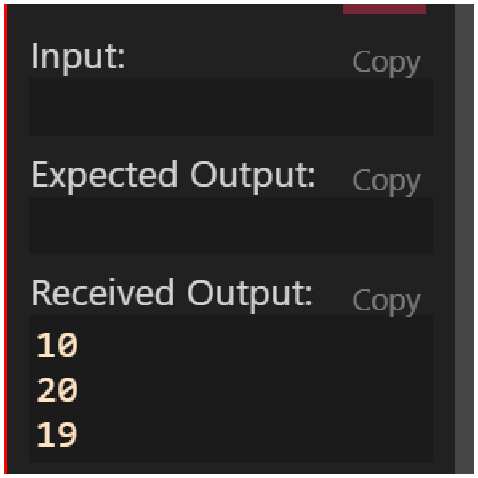
2. Write a program that generate the following output

10, 20, 19

Use an integer constant for 10, an arithmetic C++ ASSIGNMENT operator to generate the 20, and a decrement operator to generate 19.

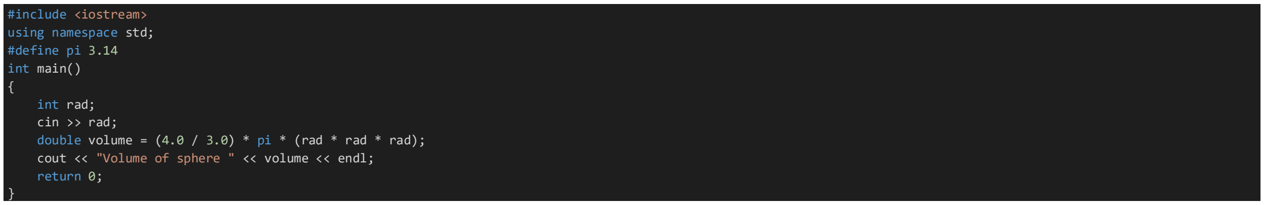


```
#include <iostream>
using namespace std;
int main()
{
    int a = 10;
    cout << a << endl;
    int b = 20;
    cout << b << endl;
    int c = --b;
    cout << c << endl;
    return 0;
}
```

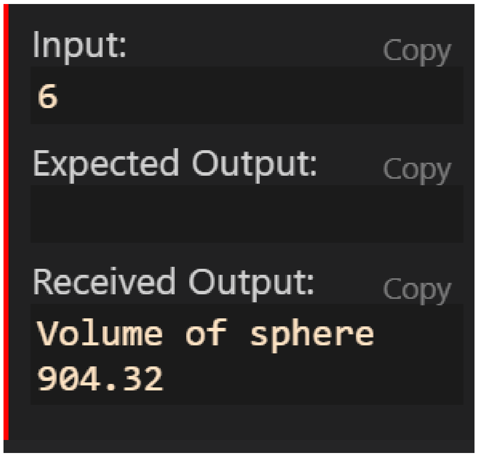


```
10
20
19
```

3. Write a program that asks the user to enter a radius value and then compute the volume of a sphere with the input radius.



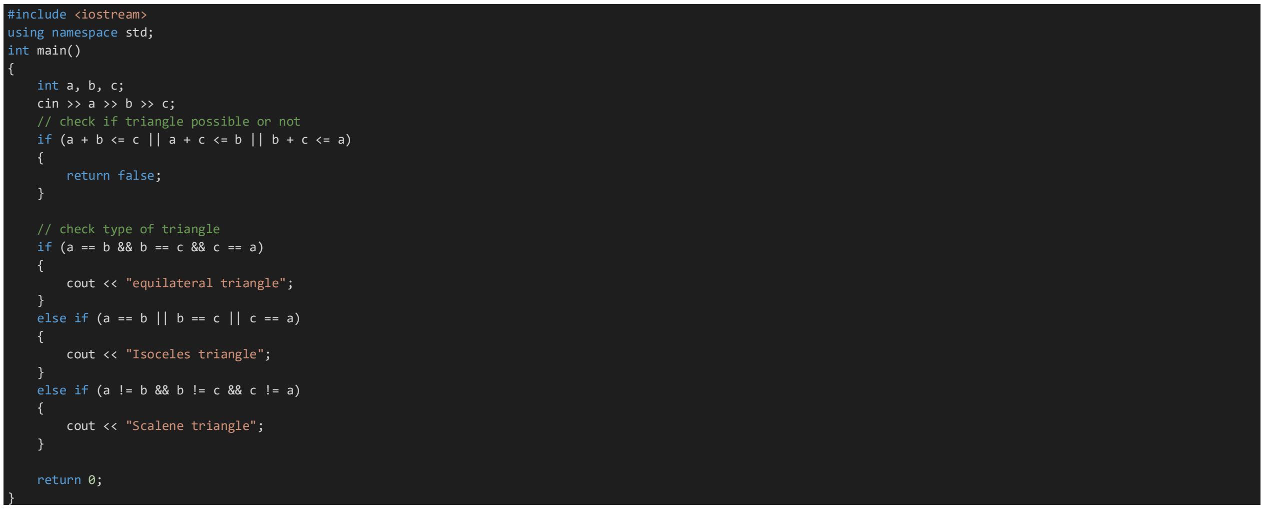
```
#include <iostream>
using namespace std;
#define pi 3.14
int main()
{
    int rad;
    cin >> rad;
    double volume = (4.0 / 3.0) * pi * (rad * rad * rad);
    cout << "Volume of sphere " << volume << endl;
    return 0;
}
```



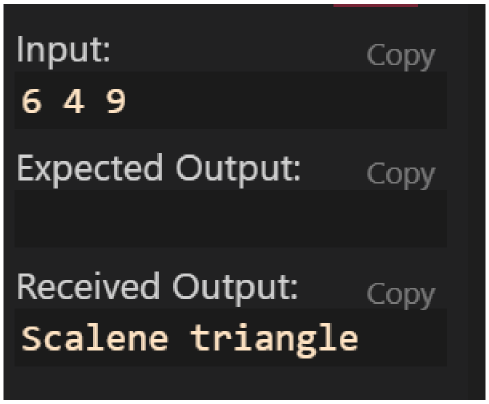
```
6
```

```
Volume of sphere
904.32
```

4. Write a program that takes three input of sides of a triangle. The program should indicate whether the triangle would be formed or not. If it can be formed it also indicates the type.



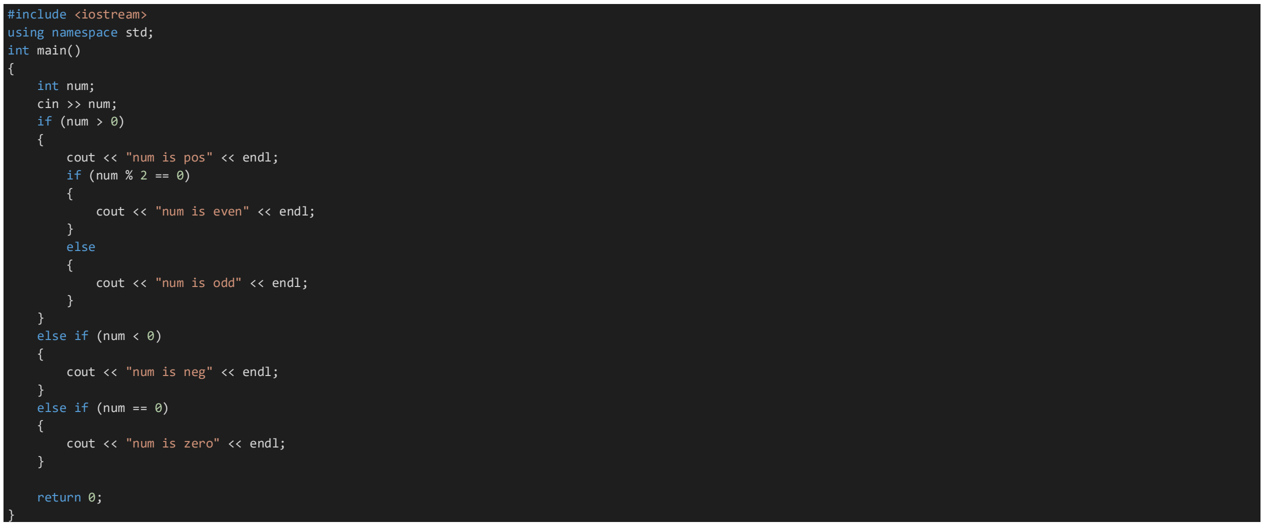
```
#include <iostream>
using namespace std;
int main()
{
    int a, b, c;
    cin >> a >> b >> c;
    // check if triangle possible or not
    if (a + b <= c || a + c <= b || b + c <= a)
    {
        return false;
    }

    // check type of triangle
    if (a == b && b == c && c == a)
    {
        cout << "equilateral triangle";
    }
    else if (a == b || b == c || c == a)
    {
        cout << "Isoceles triangle";
    }
    else if (a != b && b != c && c != a)
    {
        cout << "Scalene triangle";
    }

    return 0;
}
```



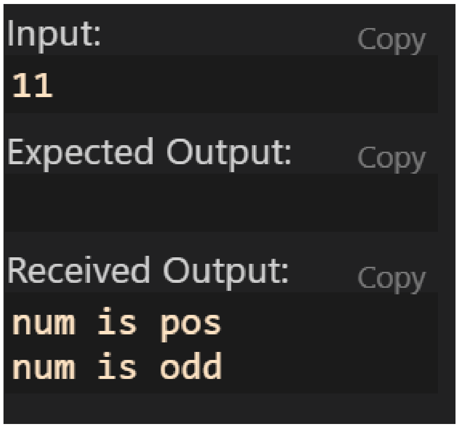
```
6 4 9
```

```
Scalene triangle
```

5. Write a program that takes one input as number and it will display whether the number is +ve, -ve or zero. If the number is +ve, then it will display whether the number is odd or even.



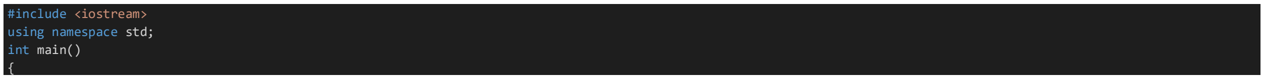
```cpp
#include <iostream>
using namespace std;
int main()
{
    int num;
    cin >> num;
    if (num > 0)
    {
        cout << "num is pos" << endl;
        if (num % 2 == 0)
        {
            cout << "num is even" << endl;
        }
        else
        {
            cout << "num is odd" << endl;
        }
    }
    else if (num < 0)
    {
        cout << "num is neg" << endl;
    }
    else if (num == 0)
    {
        cout << "num is zero" << endl;
    }

    return 0;
}
```



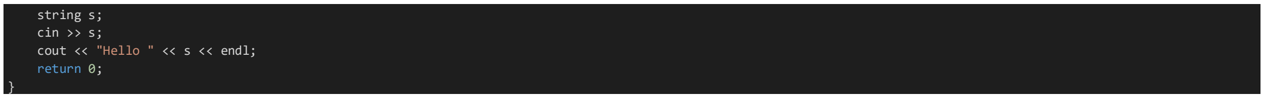
```
11
```

```
num is pos
num is odd
```

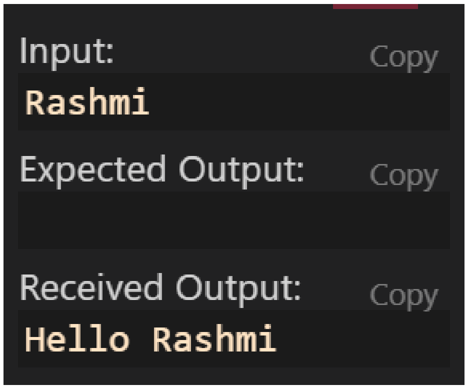
6. Write a program which takes username as input and it greets to user with his name.



```cpp
#include <iostream>
using namespace std;
int main()
{
```



```
    string s;
    cin >> s;
    cout << "Hello " << s << endl;
    return 0;
}
```



Input: Copy

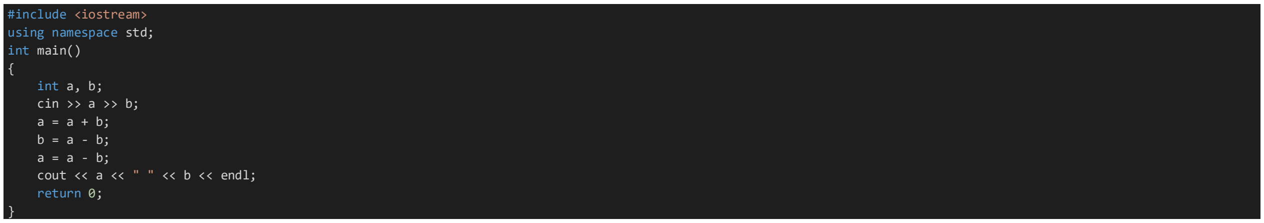
```
Rashmi
```

Expected Output: Copy

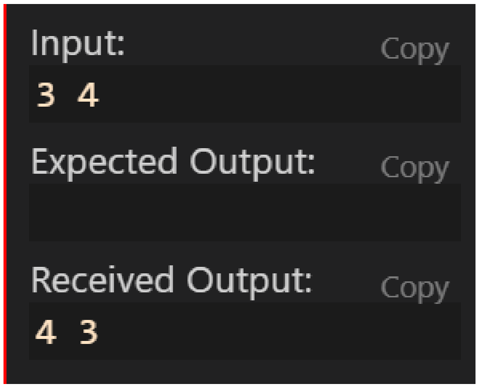
Received Output: Copy

```
Hello Rashmi
```

7. Write a program, which takes two integer numbers as input and it shows their exchanged value. (Don’t use third variable)



```
#include <iostream>
using namespace std;
int main()
{
    int a, b;
    cin >> a >> b;
    a = a + b;
    b = a - b;
    a = a - b;
    cout << a << " " << b << endl;
    return 0;
}
```



Input: Copy

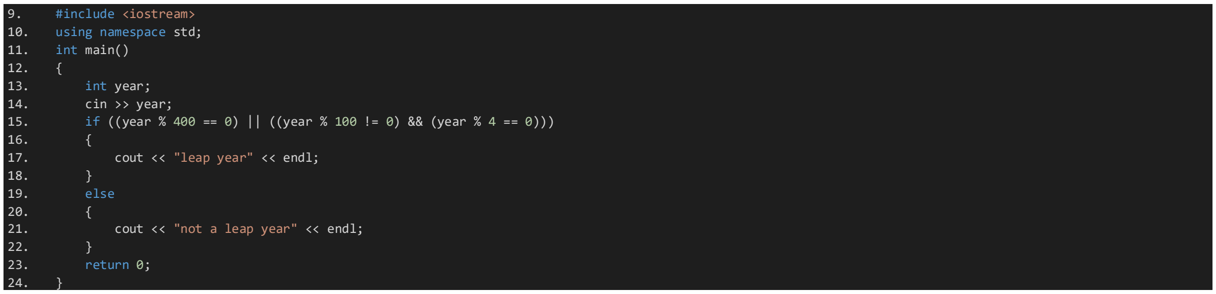
```
3 4
```

Expected Output: Copy

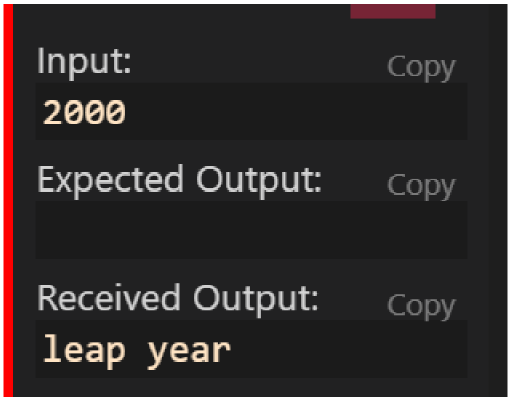
Received Output: Copy

```
4 3
```

8. WAP to check Leap Year.



```
#include <iostream>
using namespace std;
int main()
{
    int year;
    cin >> year;
    if ((year % 400 == 0) || ((year % 100 != 0) && (year % 4 == 0)))
    {
        cout << "leap year" << endl;
    }
    else
    {
        cout << "not a leap year" << endl;
    }
    return 0;
}
```



Input:

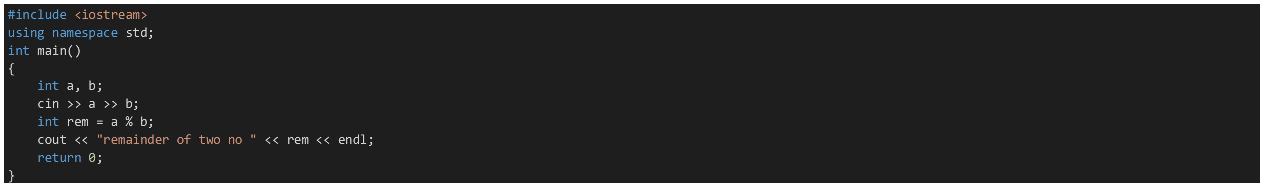
```
2000
```

Expected Output:

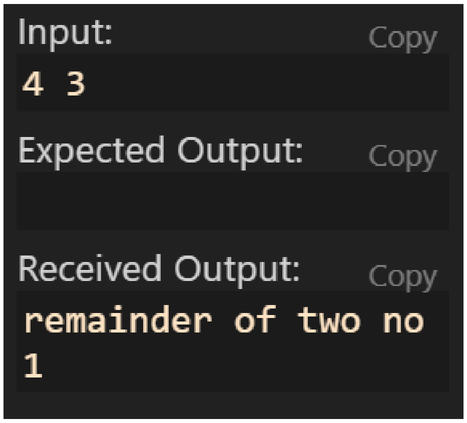
Received Output:

```
leap year
```

9. WAP for finding remainder of division of 2 numbers.



```
#include <iostream>
using namespace std;
int main()
{
    int a, b;
    cin >> a >> b;
    int rem = a % b;
    cout << "remainder of two no " << rem << endl;
    return 0;
}
```



Input:

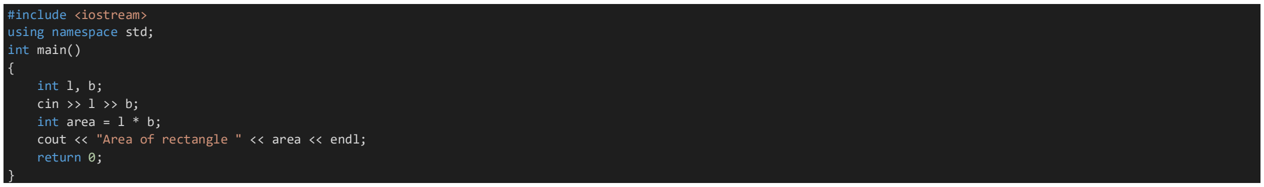
```
4 3
```

Expected Output:

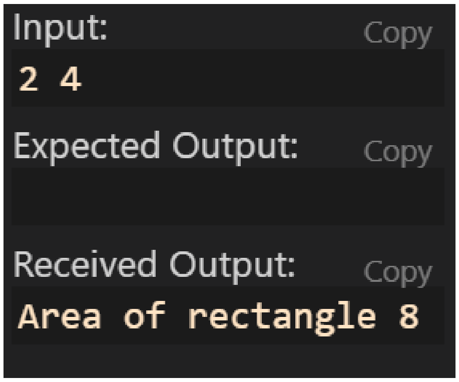
Received Output:

```
remainder of two no
1
```

10. WAP to calculate Area of Rectangle.



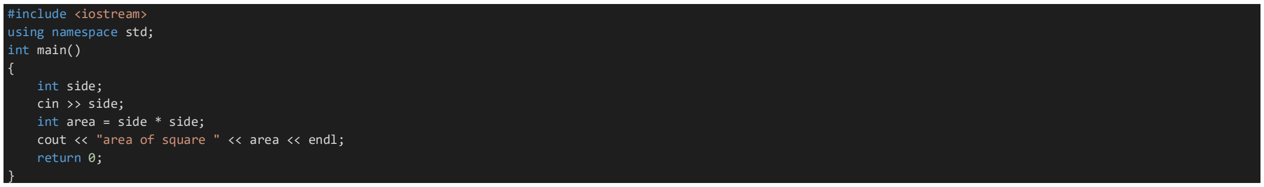
```
#include <iostream>
using namespace std;
int main()
{
    int l, b;
    cin >> l >> b;
    int area = l * b;
    cout << "Area of rectangle " << area << endl;
    return 0;
}
```



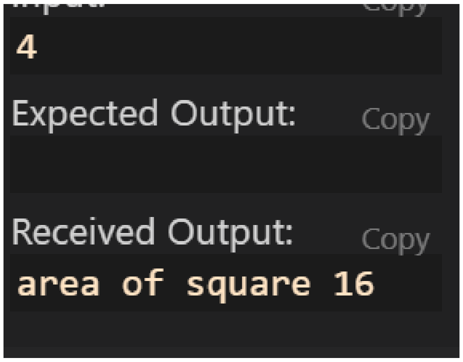
```
2 4
```

```
Area of rectangle 8
```

11. WAP to calculate Area of Square.



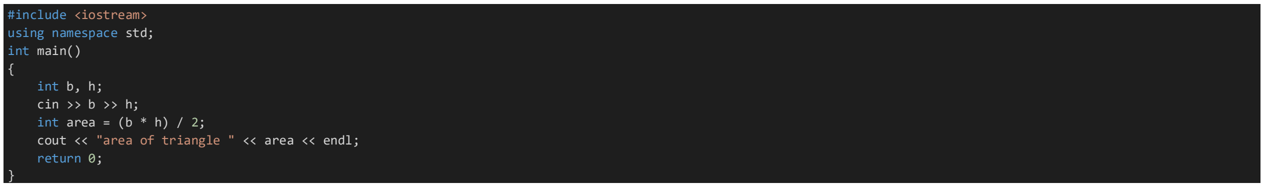
```
#include <iostream>
using namespace std;
int main()
{
    int side;
    cin >> side;
    int area = side * side;
    cout << "area of square " << area << endl;
    return 0;
}
```



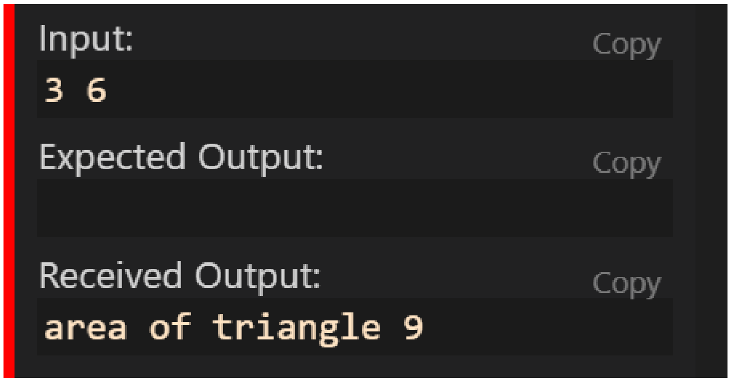
```
4
```

```
area of square 16
```

12. WAP to calculate the area of Triangle.



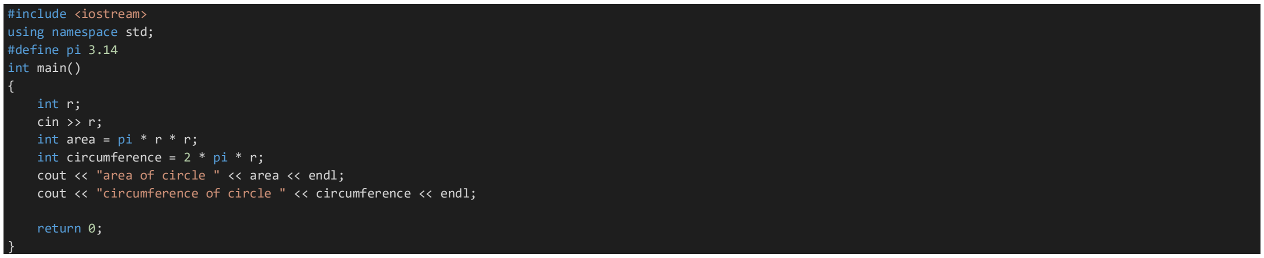
```
#include <iostream>
using namespace std;
int main()
{
    int b, h;
    cin >> b >> h;
    int area = (b * h) / 2;
    cout << "area of triangle " << area << endl;
    return 0;
}
```



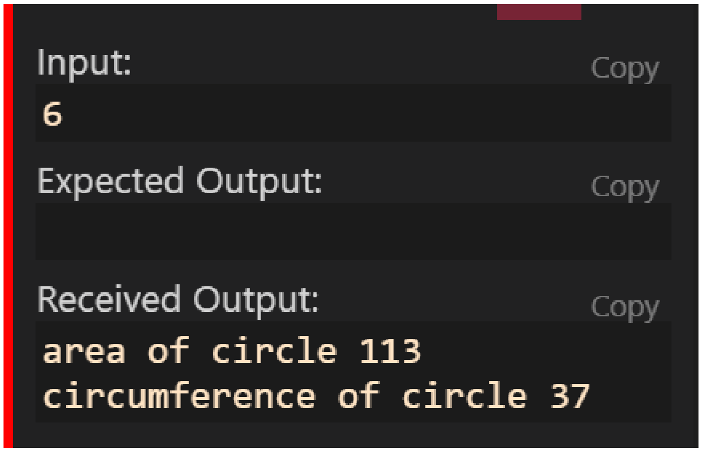
```
3 6
```

```
area of triangle 9
```

13. WAP to calculate Area and Circumference of Circle.



```cpp
#include <iostream>
using namespace std;
#define pi 3.14
int main()
{
    int r;
    cin >> r;
    int area = pi * r * r;
    int circumference = 2 * pi * r;
    cout << "area of circle " << area << endl;
    cout << "circumference of circle " << circumference << endl;

    return 0;
}
```



```
6
```

```
area of circle 113
circumference of circle 37
```

14. WAP for two item’s weight (floating points&#39; values) and number of purchase (floating points&#39; values) and

calculate the average value of the items.

Test Data:

Weight - Item1: 15

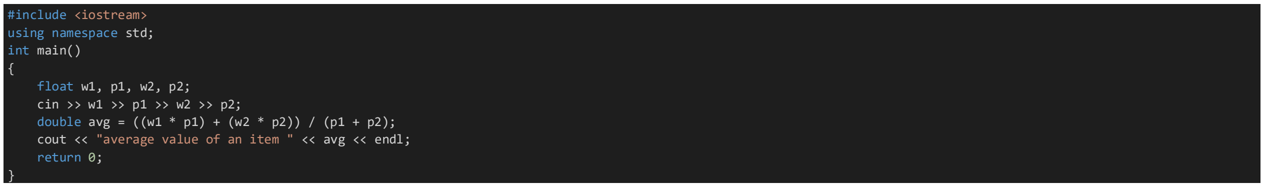
No. of item1: 5

Weight - Item2: 25

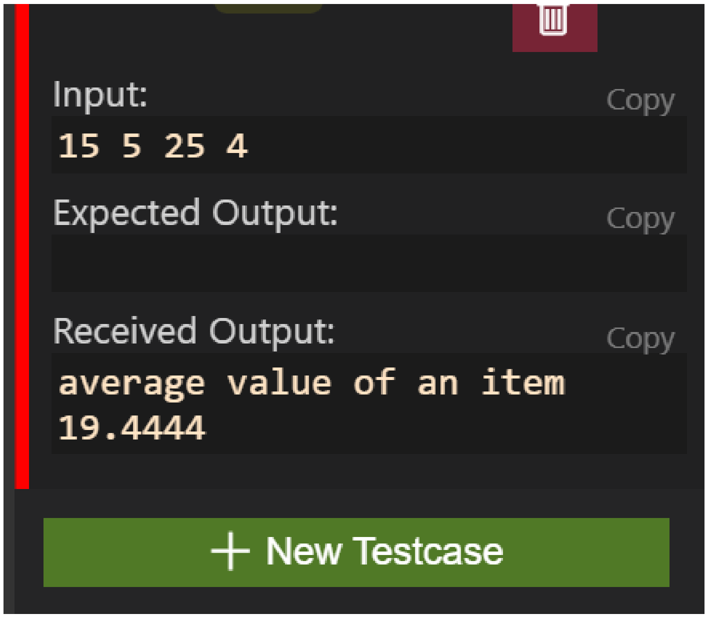
No. of item2: 4

Expected Output:

Average Value = 19.444444



```
#include <iostream>
using namespace std;
int main()
{
    float w1, p1, w2, p2;
    cin >> w1 >> p1 >> w2 >> p2;
    double avg = ((w1 * p1) + (w2 * p2)) / (p1 + p2);
    cout << "average value of an item " << avg << endl;
    return 0;
}
```



Input:

```
15 5 25 4
```

Expected Output:

Received Output:

```
average value of an item
19.4444
```

15. WAP to calculate a bike’s average consumption from the given total distance (integer value) travelled (in km)

and spent fuel.

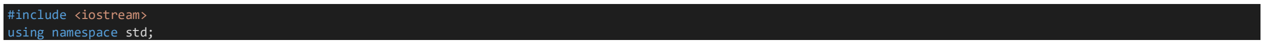
Test Data:

Input total distance in km: 350

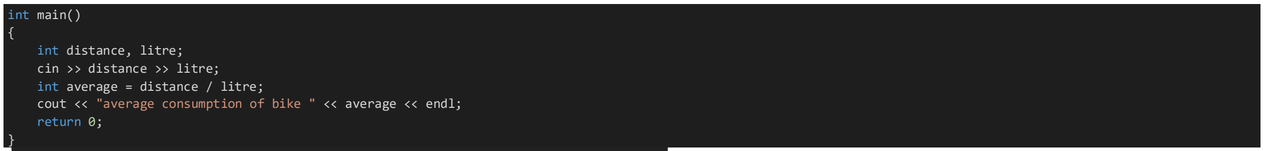
Input total fuel spent in litres: 5

Expected Output:

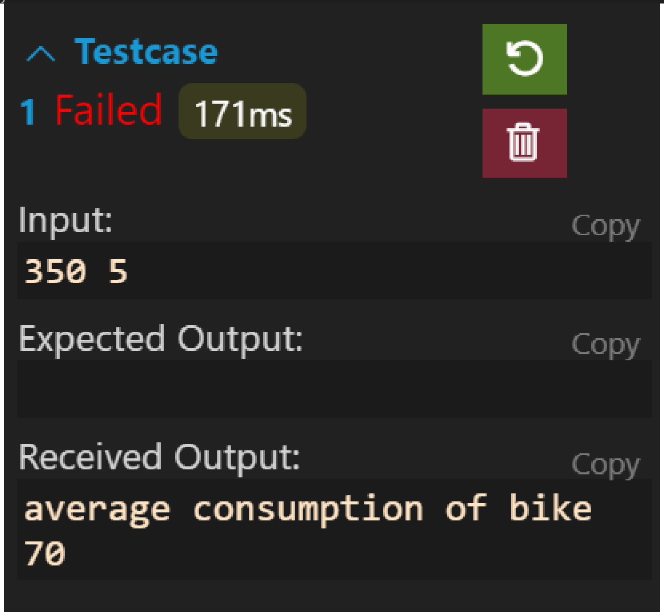
Average consumption (km/lt) 70.00



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



```
int main()
{
    int distance, litre;
    cin >> distance >> litre;
    int average = distance / litre;
    cout << "average consumption of bike " << average << endl;
    return 0;
}
```



Input:

```
350 5
```

Expected Output:

Received Output:

```
average consumption of bike
70
```

16. Write a program that will give the grade of the student based on the percentage he got in the course.

Use the following criteria for assigning grades:

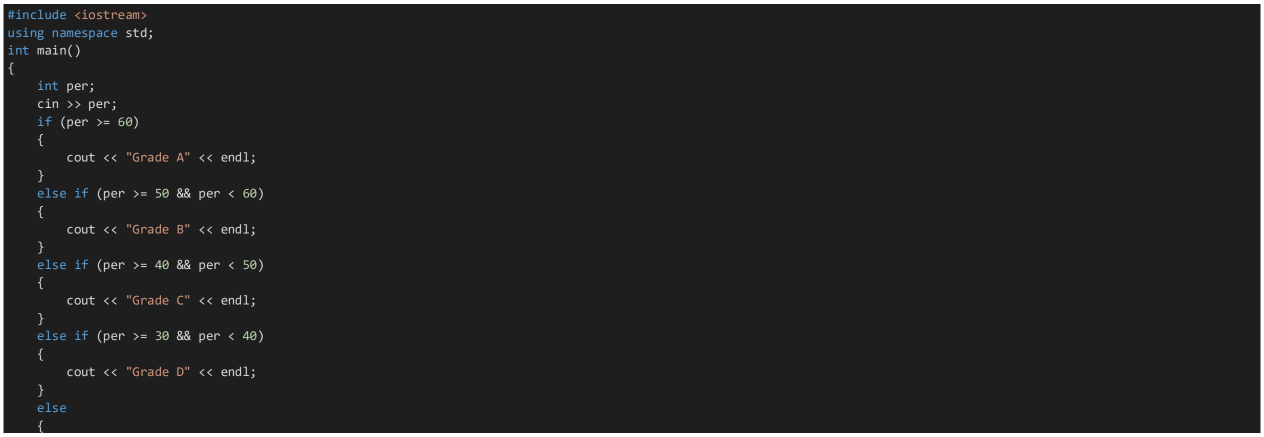
Grade = A (when percentage &gt;= 60)

Grade = B (when percentage &gt;= 50 and percentage &lt; 60)

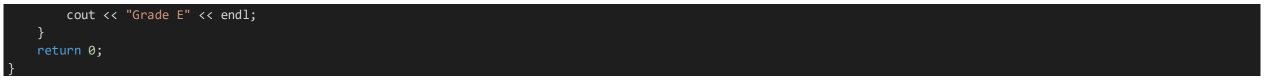
Grade = C (when percentage &gt;= 40 and percentage &lt; 50)

Grade = D (when percentage &gt;= 30 and percentage &lt; 40)

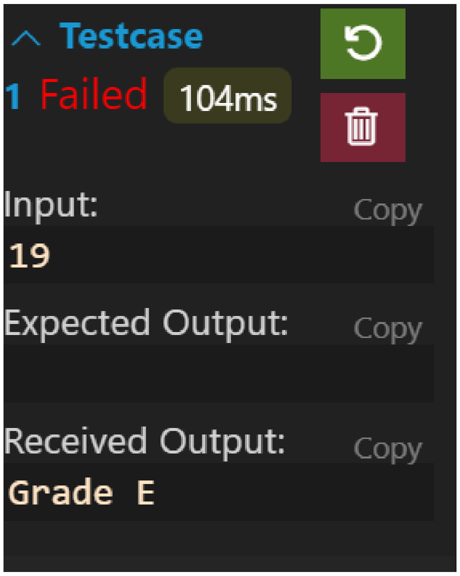
Grade = E (when percentage &gt;= 20 and percentage &lt; 30)



```
#include <iostream>
using namespace std;
int main()
{
    int per;
    cin >> per;
    if (per >= 60)
    {
        cout << "Grade A" << endl;
    }
    else if (per >= 50 && per < 60)
    {
        cout << "Grade B" << endl;
    }
    else if (per >= 40 && per < 50)
    {
        cout << "Grade C" << endl;
    }
    else if (per >= 30 && per < 40)
    {
        cout << "Grade D" << endl;
    }
    else
    {
```



```cpp
        cout << "Grade E" << endl;
    }
    return 0;
}
```



1 Failed 104ms

Input:

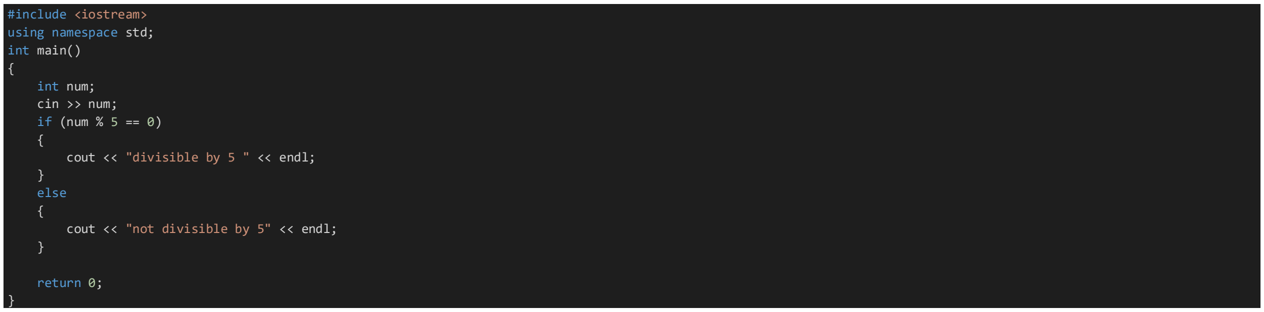
```
19
```

Expected Output:

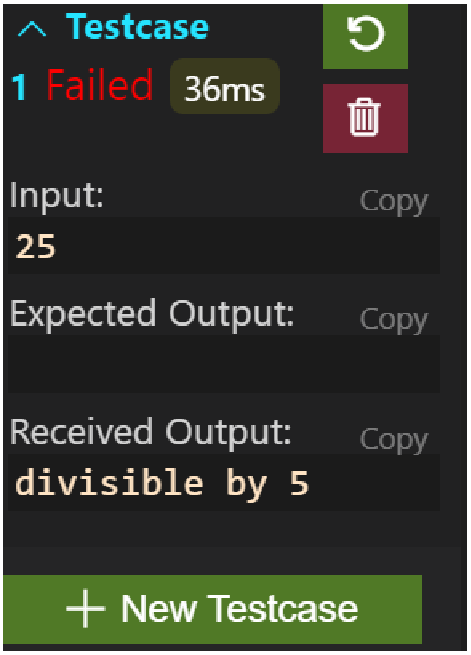
Received Output:

```
Grade E
```

17. WAP to check whether a number is divisible by 5.



```cpp
#include <iostream>
using namespace std;
int main()
{
    int num;
    cin >> num;
    if (num % 5 == 0)
    {
        cout << "divisible by 5 " << endl;
    }
    else
    {
        cout << "not divisible by 5" << endl;
    }

    return 0;
}
```



Input:
```
25
```

Expected Output:

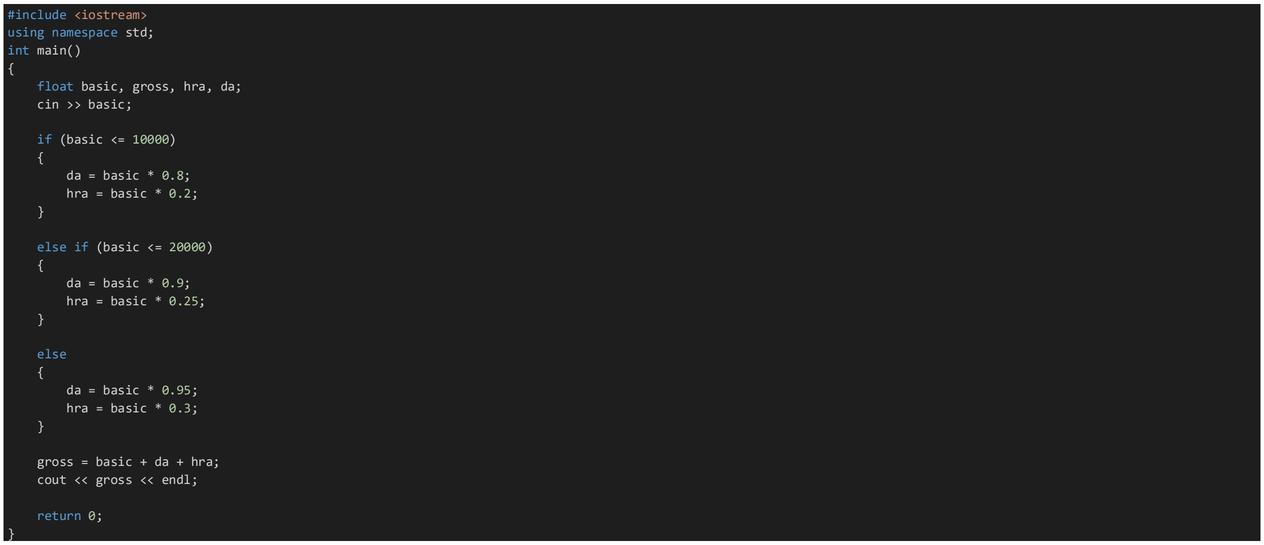
Received Output:
```
divisible by 5
```

18. WAP to input basic salary of an employee and calculate its Gross salary according to following:

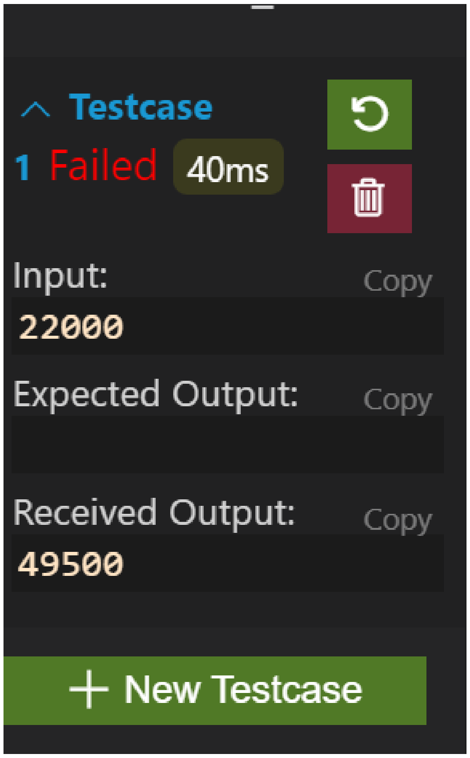
Basic Salary &lt;= 10000 : HRA = 20%, DA = 80%

Basic Salary &lt;= 20000 : HRA = 25%, DA = 90%

Basic Salary &gt; 20000 : HRA = 30%, DA = 95%



```
#include <iostream>
using namespace std;
int main()
{
    float basic, gross, hra, da;
    cin >> basic;

    if (basic <= 10000)
    {
        da = basic * 0.8;
        hra = basic * 0.2;
    }

    else if (basic <= 20000)
    {
        da = basic * 0.9;
        hra = basic * 0.25;
    }

    else
    {
        da = basic * 0.95;
        hra = basic * 0.3;
    }

    gross = basic + da + hra;
    cout << gross << endl;

    return 0;
}
```



1 Failed 40ms

Input:

```
22000
```

Expected Output:

Received Output:

```
49500
```

19. WAP to input electricity unit charges and calculate total electricity bill according to the given condition:

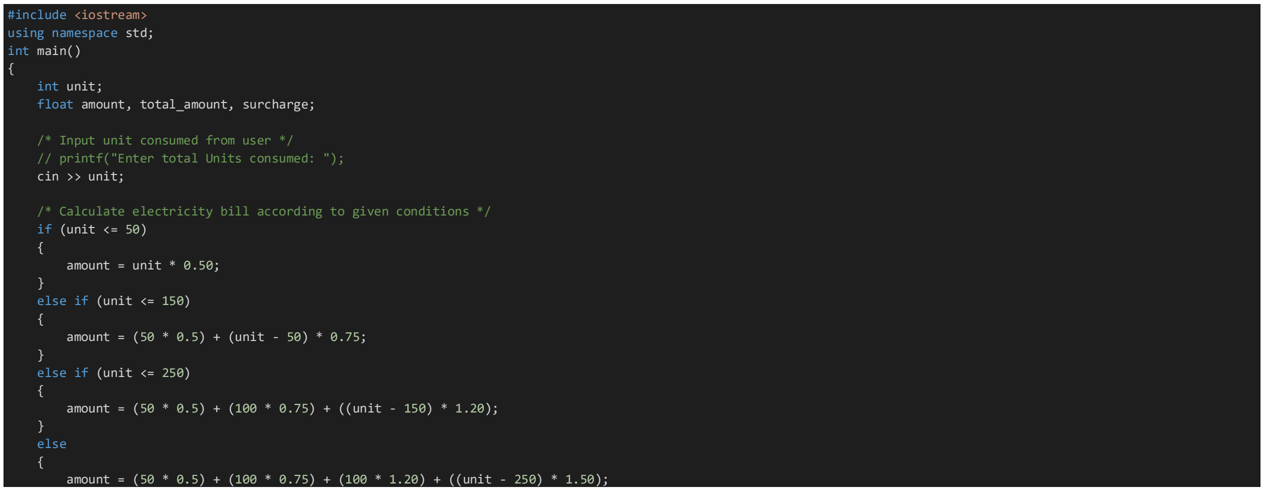
For first 50 units Rs. 0.50/unit

For next 100 units Rs. 0.75/unit

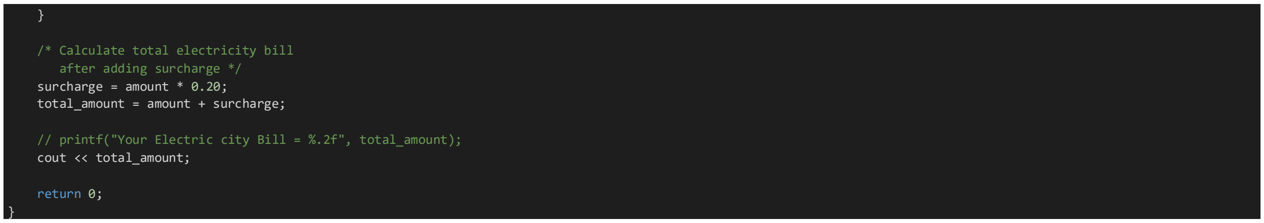
For next 100 units Rs. 1.20/unit

For unit above 250 Rs. 1.50/unit

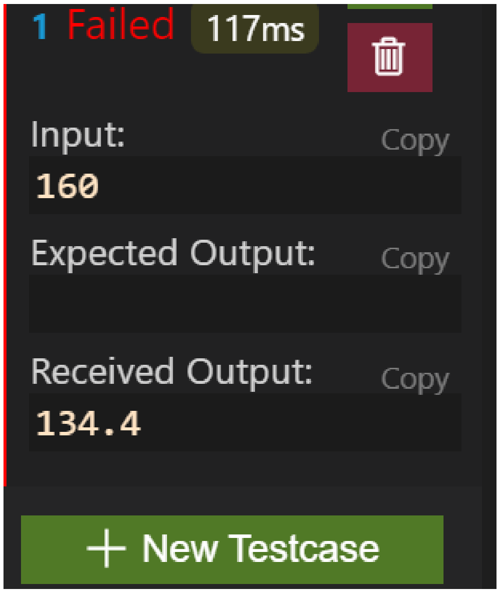
An additional surcharge of 20% is added to the bill



```
#include <iostream>
using namespace std;
int main()
{
    int unit;
    float amount, total_amount, surcharge;

    /* Input unit consumed from user */
    // printf("Enter total Units consumed: ");
    cin >> unit;

    /* Calculate electricity bill according to given conditions */
    if (unit <= 50)
    {
        amount = unit * 0.50;
    }
    else if (unit <= 150)
    {
        amount = (50 * 0.5) + (unit - 50) * 0.75;
    }
    else if (unit <= 250)
    {
        amount = (50 * 0.5) + (100 * 0.75) + ((unit - 150) * 1.20);
    }
    else
    {
        amount = (50 * 0.5) + (100 * 0.75) + (100 * 1.20) + ((unit - 250) * 1.50);
```



```
    }

    /* Calculate total electricity bill
       after adding surcharge */
    surcharge = amount * 0.20;
    total_amount = amount + surcharge;

    // printf("Your Electric city Bill = %.2f", total_amount);
    cout << total_amount;

    return 0;
}
```



Input:

```
160
```

Expected Output:

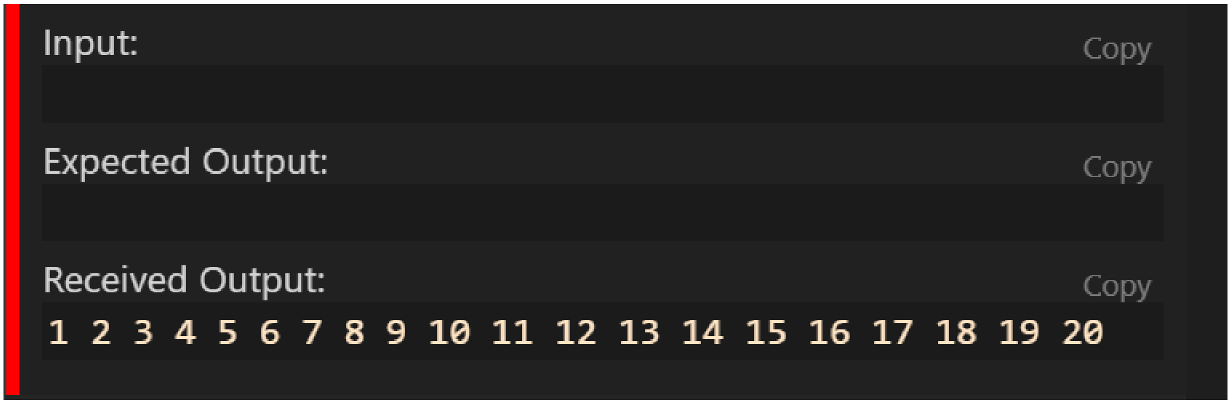
```
```

Received Output:

```
134.4
```

1. WAP for printing all natural numbers till 20.

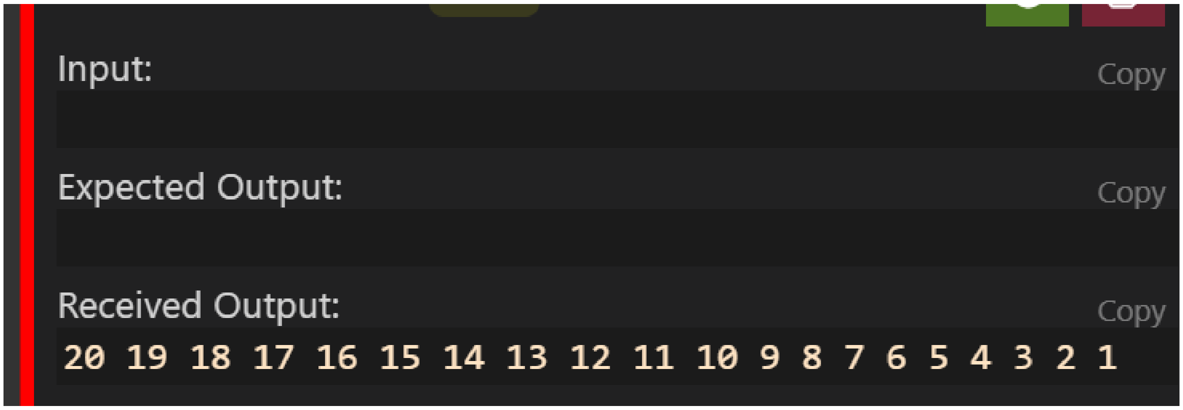
```
#include <iostream>
using namespace std;
int main()
{
    for (int i = 1; i <= 20; i++)
    {
        cout << i << " ";
    }
    return 0;
}
```



```
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
```

2. WAP for printing all natural numbers in reverse order starting from 20.

```
#include <iostream>
using namespace std;
int main()
{
    for (int i = 20; i >= 1; i--)
    {
        cout << i << endl;
    }
    return 0;
}
```

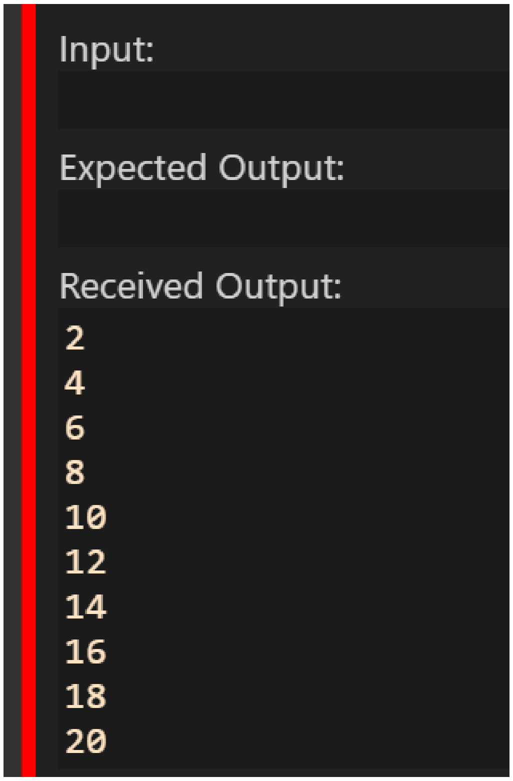


```
20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1
```

3. WAP for printing all even numbers from 1 to 20.

```
#include <iostream>
using namespace std;
int main()
{
    for (int i = 1; i <= 20; i++)
    {
        if (i % 2 == 0)
        {
            cout << i << endl;
        }
    }
```

```
    return 0;
}
```



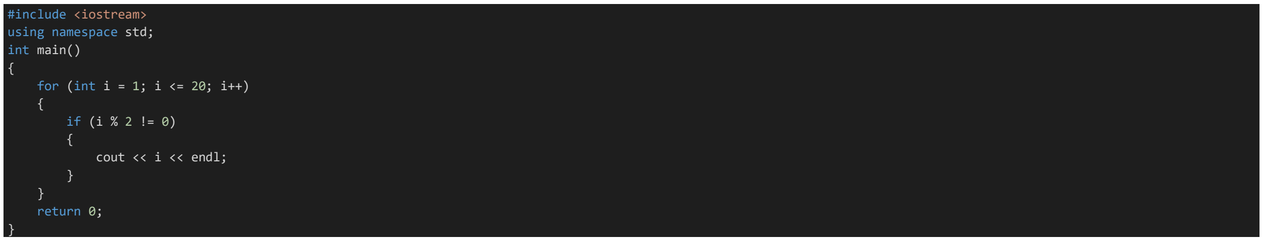
Input:

Expected Output:

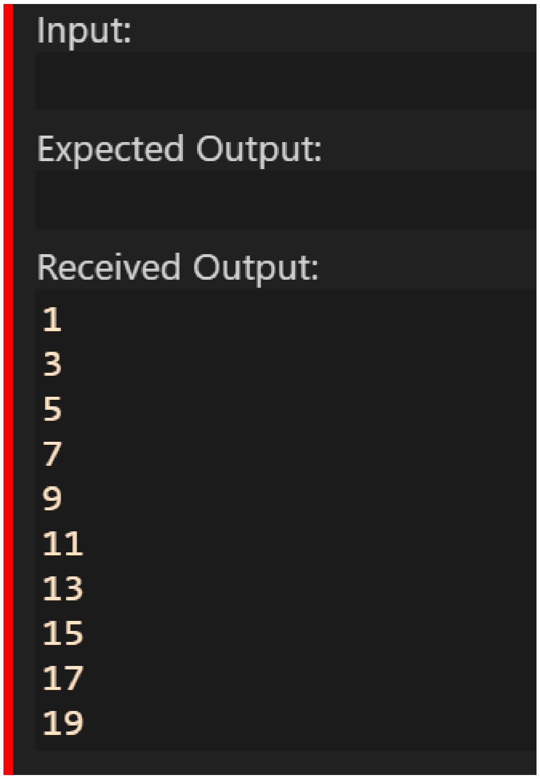
Received Output:

```
2
4
6
8
10
12
14
16
18
20
```

4. WAP for printing all odd numbers from 1 to 20.

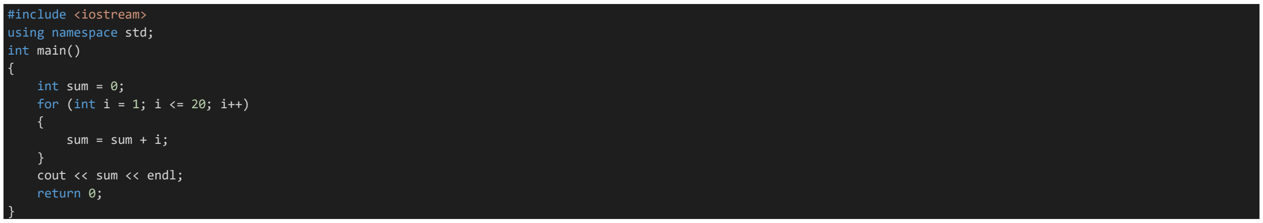


```
#include <iostream>
using namespace std;
int main()
{
    for (int i = 1; i <= 20; i++)
    {
        if (i % 2 != 0)
        {
            cout << i << endl;
        }
    }
    return 0;
}
```

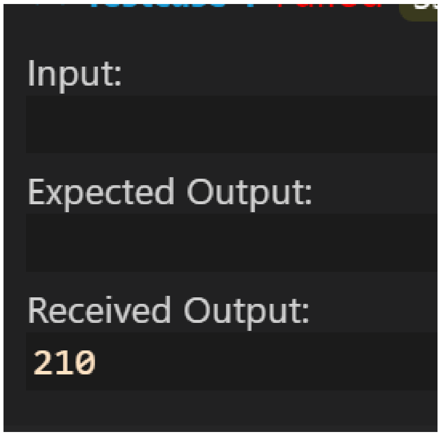


```
Input:

Expected Output:

Received Output:
1
3
5
7
9
11
13
15
17
19
```

5. WAP for adding all numbers from 1 to 20.

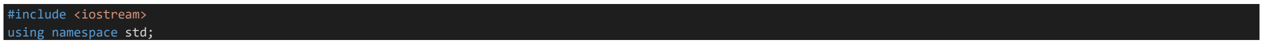


```
#include <iostream>
using namespace std;
int main()
{
    int sum = 0;
    for (int i = 1; i <= 20; i++)
    {
        sum = sum + i;
    }
    cout << sum << endl;
    return 0;
}
```

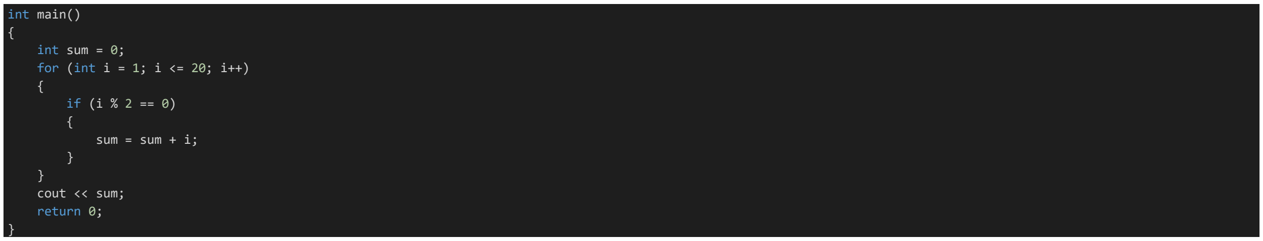


```
Input:

Expected Output:

Received Output:
210
```

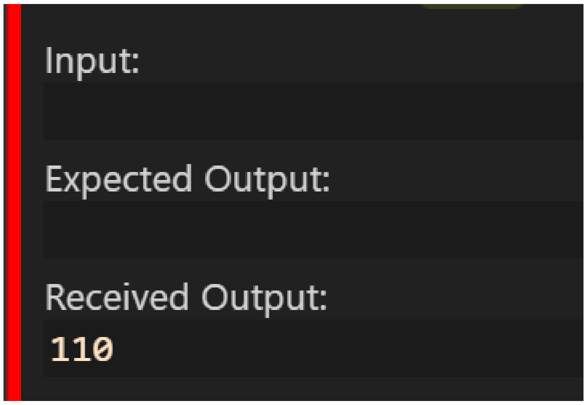
6. WAP for finding sum of all even numbers till 20.



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



```
int main()
{
    int sum = 0;
    for (int i = 1; i <= 20; i++)
    {
        if (i % 2 == 0)
        {
            sum = sum + i;
        }
    }
    cout << sum;
    return 0;
}
```



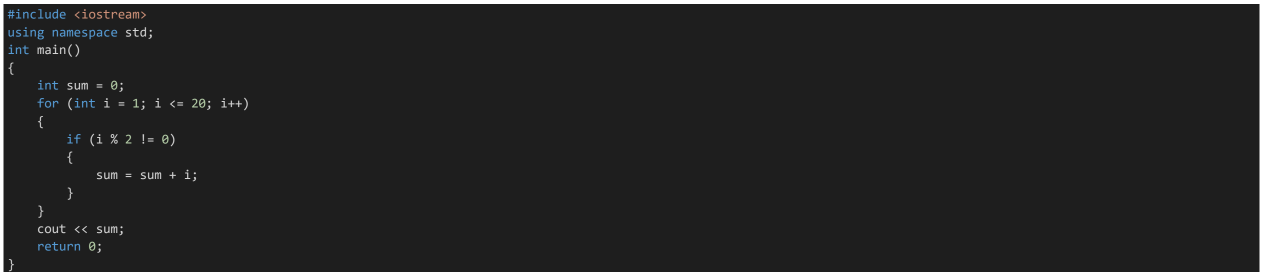
Input:

Expected Output:

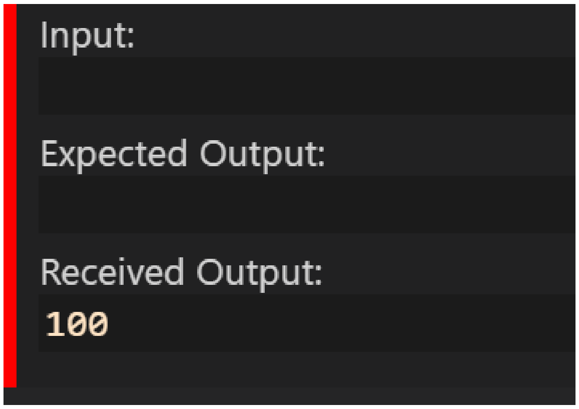
Received Output:

```
110
```

7. WAP for finding sum of all odd numbers till 20.



```
#include <iostream>
using namespace std;
int main()
{
    int sum = 0;
    for (int i = 1; i <= 20; i++)
    {
        if (i % 2 != 0)
        {
            sum = sum + i;
        }
    }
    cout << sum;
    return 0;
}
```



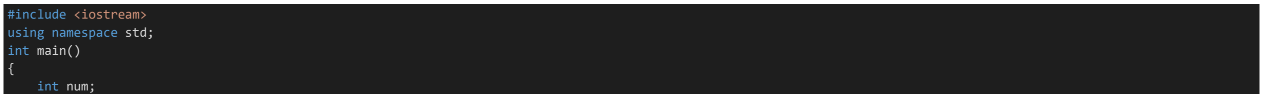
Input:

Expected Output:

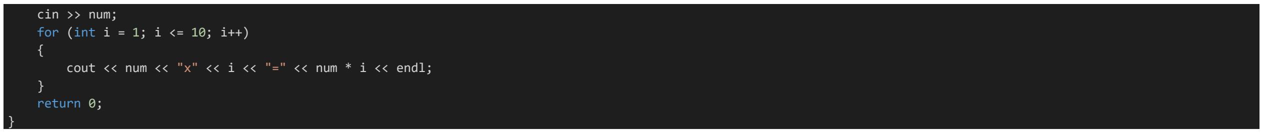
Received Output:

```
100
```

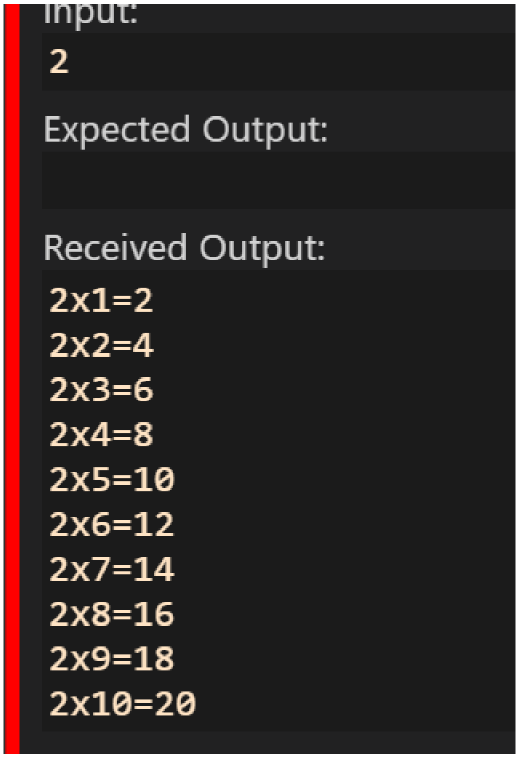
8. WAP for printing multiplication table of a number. For eg. Display should be “ 2 X 1 = 2”



```
#include <iostream>
using namespace std;
int main()
{
    int num;
```

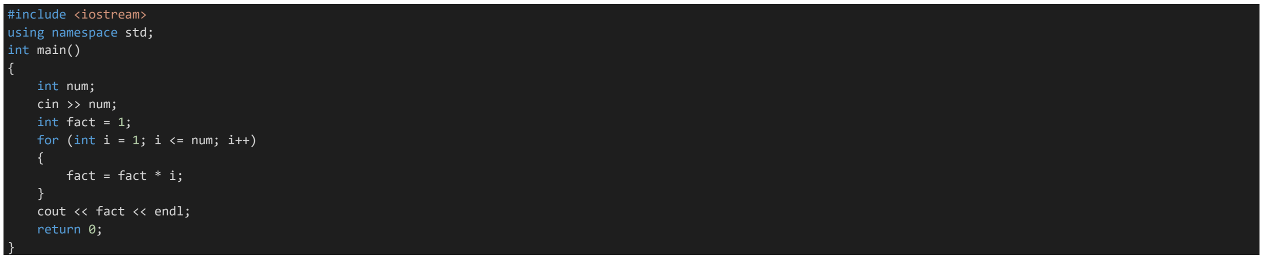


```
    cin >> num;
    for (int i = 1; i <= 10; i++)
    {
        cout << num << "x" << i << "=" << num * i << endl;
    }
    return 0;
}
```

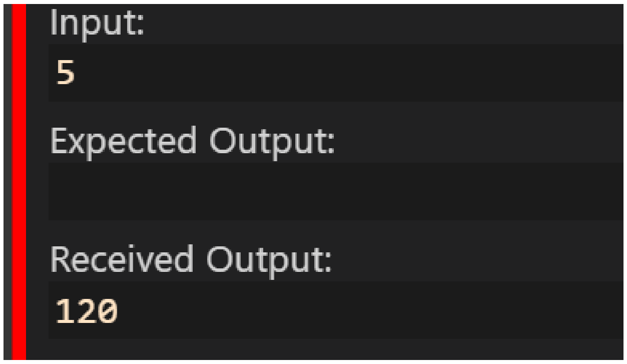


```
Input:
2

Expected Output:


Received Output:
2x1=2
2x2=4
2x3=6
2x4=8
2x5=10
2x6=12
2x7=14
2x8=16
2x9=18
2x10=20
```

9. WAP to calculate factorial of a number.

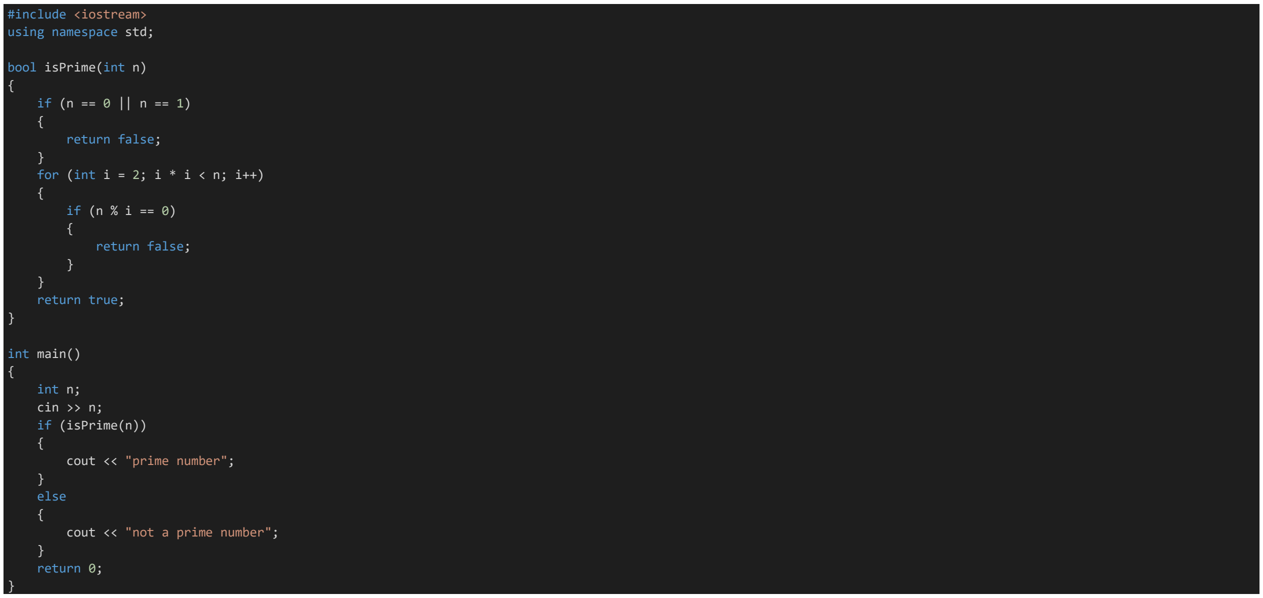


```
#include <iostream>
using namespace std;
int main()
{
    int num;
    cin >> num;
    int fact = 1;
    for (int i = 1; i <= num; i++)
    {
        fact = fact * i;
    }
    cout << fact << endl;
    return 0;
}
```

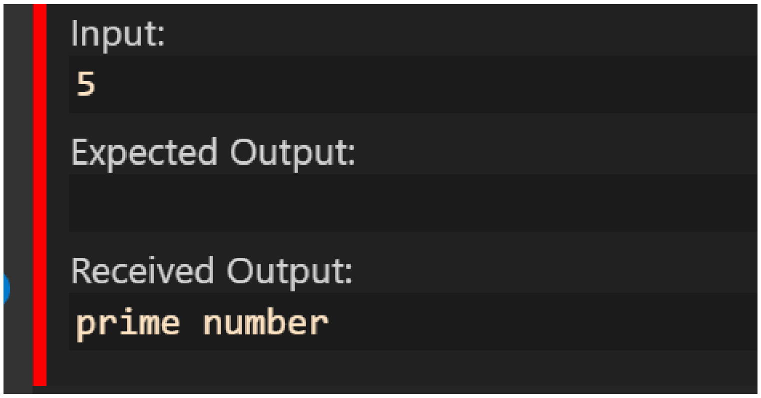


```
Input:
5

Expected Output:


Received Output:
120
```

10. WAP to check whether a number is prime or not.



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

bool isPrime(int n)
{
    if (n == 0 || n == 1)
    {
        return false;
    }
    for (int i = 2; i * i < n; i++)
    {
        if (n % i == 0)
        {
            return false;
        }
    }
    return true;
}

int main()
{
    int n;
    cin >> n;
    if (isPrime(n))
    {
        cout << "prime number";
    }
    else
    {
        cout << "not a prime number";
    }
    return 0;
}
```



Input:

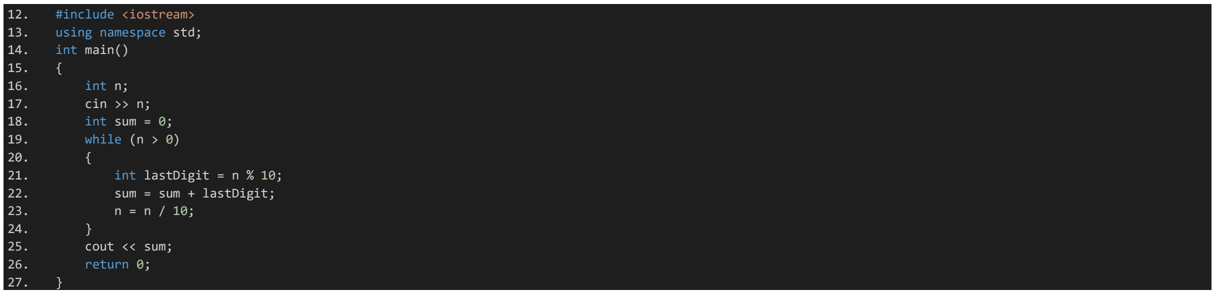
```
5
```

Expected Output:

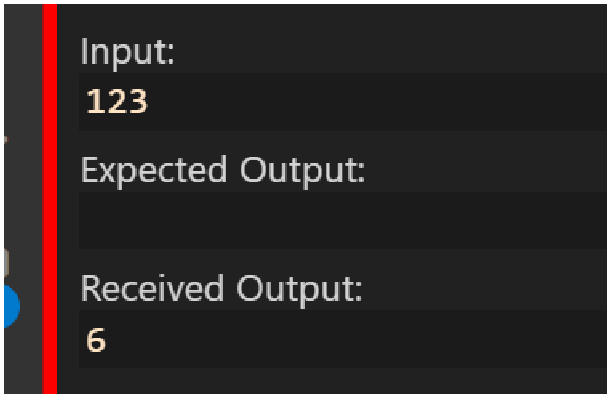
Received Output:

```
prime number
```

11. WAP to print all digits of a number and their sum.

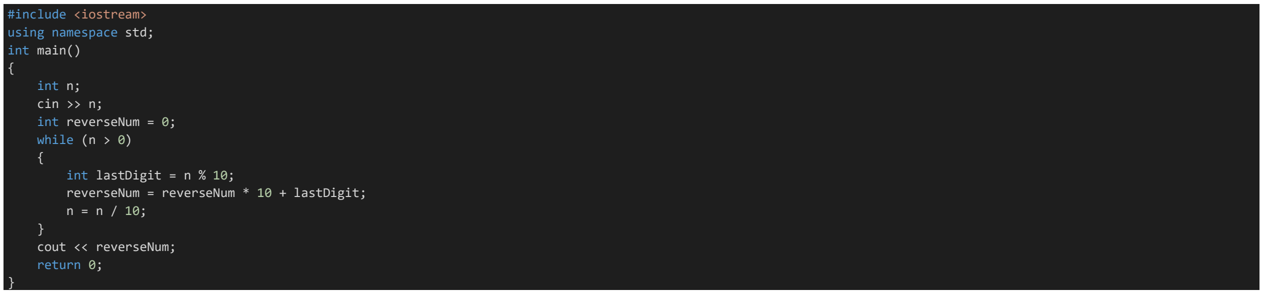


```cpp
#include <iostream>
using namespace std;
int main()
{
    int n;
    cin >> n;
    int sum = 0;
    while (n > 0)
    {
        int lastDigit = n % 10;
        sum = sum + lastDigit;
        n = n / 10;
    }
    cout << sum;
    return 0;
}
```

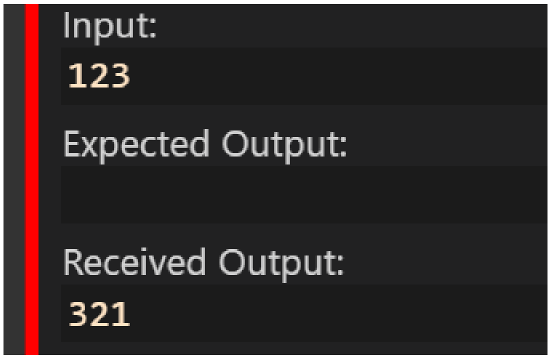


```
Input:
123
Expected Output:

Received Output:
6
```

12. WAP to print reverse of a number.

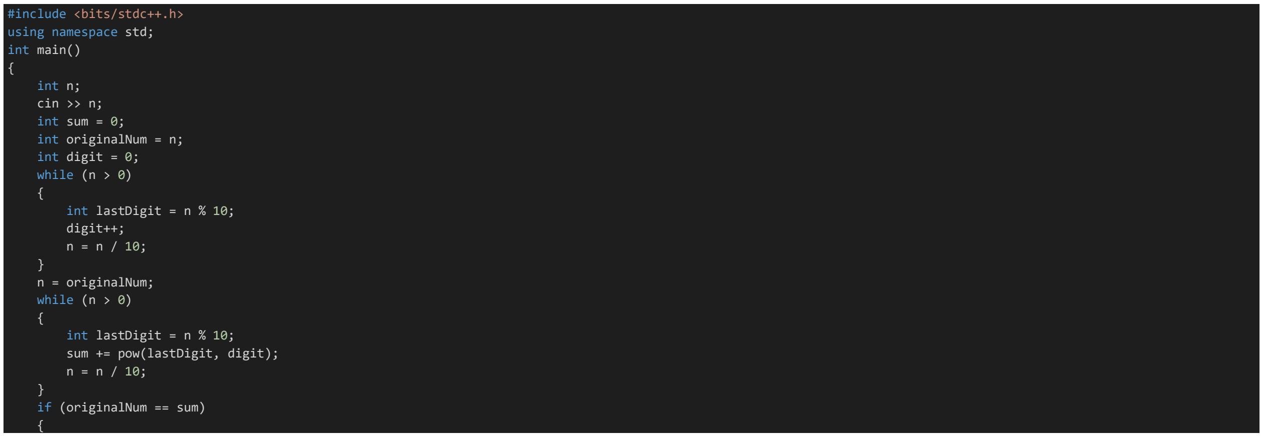


```
#include <iostream>
using namespace std;
int main()
{
    int n;
    cin >> n;
    int reverseNum = 0;
    while (n > 0)
    {
        int lastDigit = n % 10;
        reverseNum = reverseNum * 10 + lastDigit;
        n = n / 10;
    }
    cout << reverseNum;
    return 0;
}
```

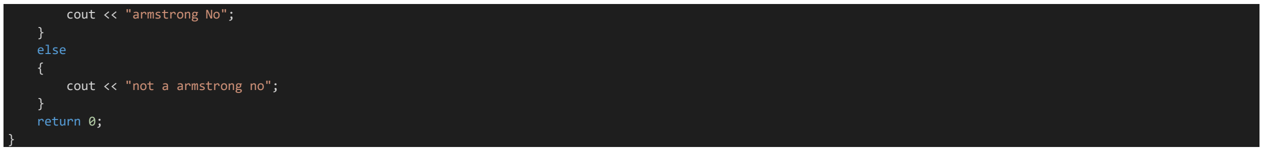


```
Input:
123
Expected Output:

Received Output:
321
```

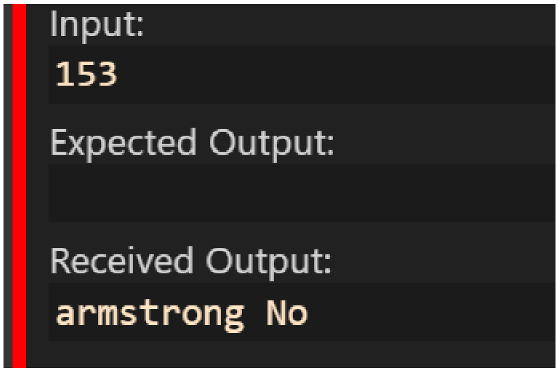
13. WAP to check whether the number is Armstrong or not.



```
#include <bits/stdc++.h>
using namespace std;
int main()
{
    int n;
    cin >> n;
    int sum = 0;
    int originalNum = n;
    int digit = 0;
    while (n > 0)
    {
        int lastDigit = n % 10;
        digit++;
        n = n / 10;
    }
    n = originalNum;
    while (n > 0)
    {
        int lastDigit = n % 10;
        sum += pow(lastDigit, digit);
        n = n / 10;
    }
    if (originalNum == sum)
    {
```

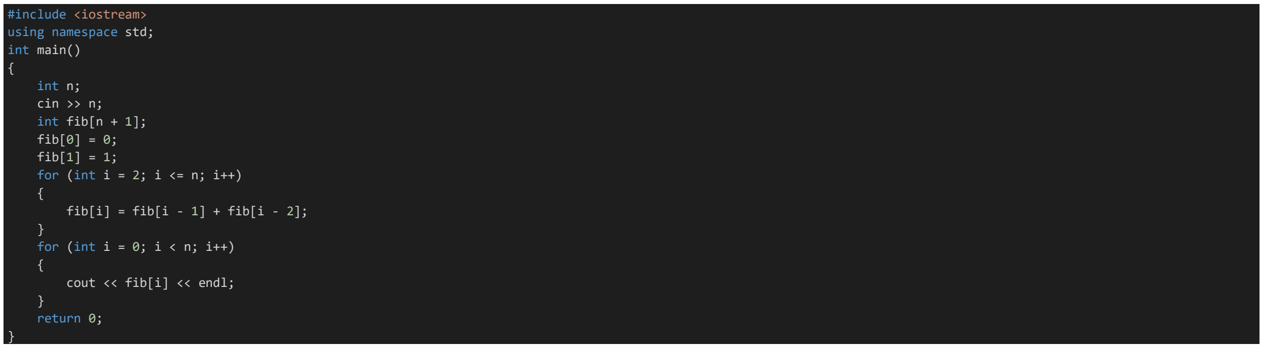


```
        cout << "armstrong No";
    }
    else
    {
        cout << "not a armstrong no";
    }
    return 0;
}
```

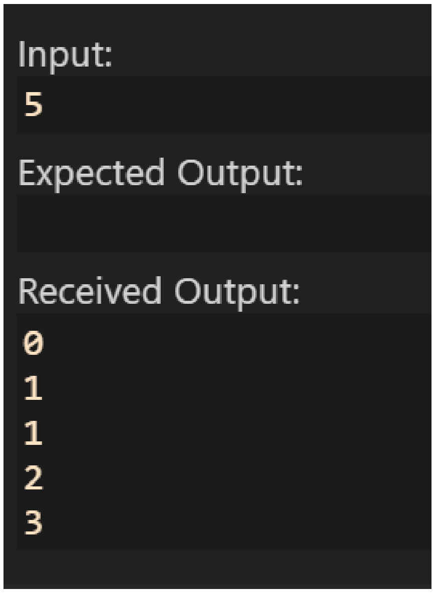


```
Input:
153
Expected Output:

Received Output:
armstrong No
```

14. WAP to print the Fibonacci series in a given range.

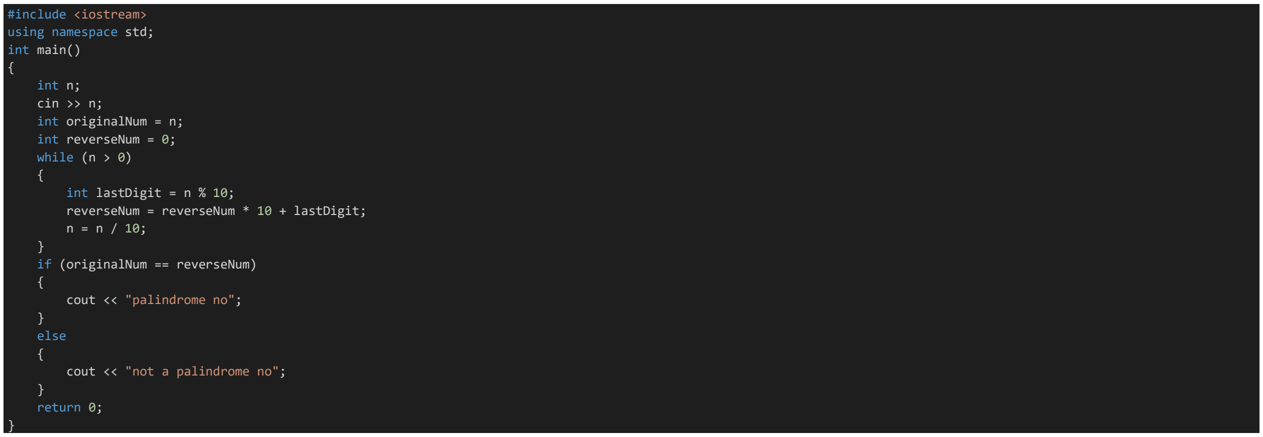


```
#include <iostream>
using namespace std;
int main()
{
    int n;
    cin >> n;
    int fib[n + 1];
    fib[0] = 0;
    fib[1] = 1;
    for (int i = 2; i <= n; i++)
    {
        fib[i] = fib[i - 1] + fib[i - 2];
    }
    for (int i = 0; i < n; i++)
    {
        cout << fib[i] << endl;
    }
    return 0;
}
```

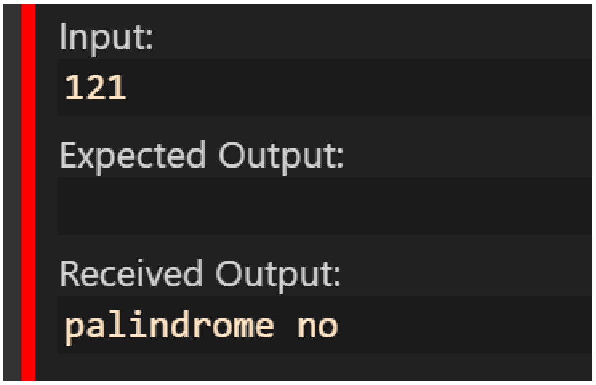


```
Input:
5
Expected Output:

Received Output:
0
1
1
2
3
```

15. WAP to check whether the number entered is palindrome or not.

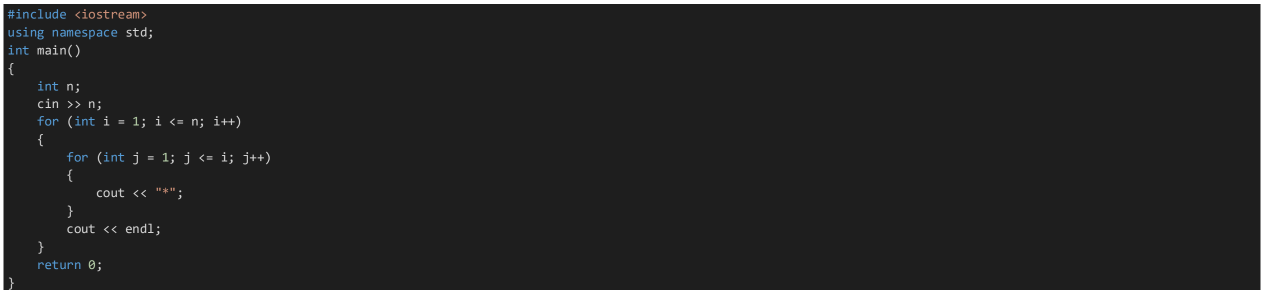


```cpp
#include <iostream>
using namespace std;
int main()
{
    int n;
    cin >> n;
    int originalNum = n;
    int reverseNum = 0;
    while (n > 0)
    {
        int lastDigit = n % 10;
        reverseNum = reverseNum * 10 + lastDigit;
        n = n / 10;
    }
    if (originalNum == reverseNum)
    {
        cout << "palindrome no";
    }
    else
    {
        cout << "not a palindrome no";
    }
    return 0;
}
```

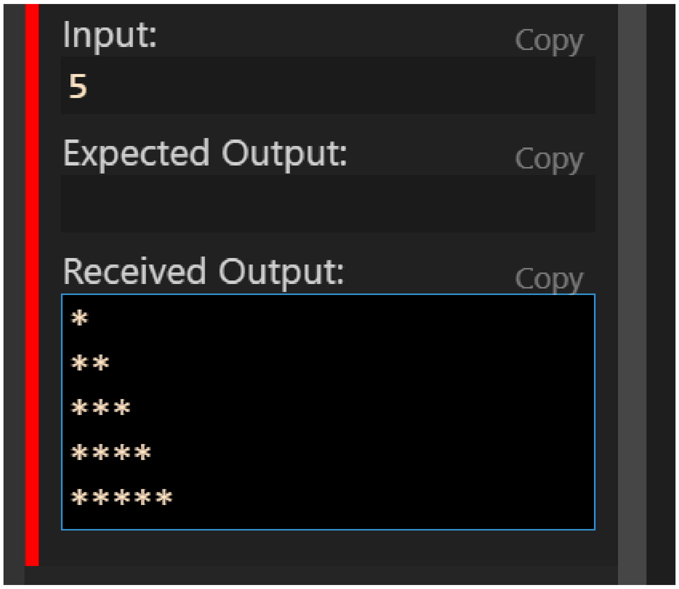


```
Input:
121
Expected Output:

Received Output:
palindrome no
```

```
1.  *

    **

    ***

    ****

    ******
```



```cpp
#include <iostream>
using namespace std;
int main()
{
    int n;
    cin >> n;
    for (int i = 1; i <= n; i++)
    {
        for (int j = 1; j <= i; j++)
        {
            cout << "*";
        }
        cout << endl;
    }
    return 0;
}
```



Input:

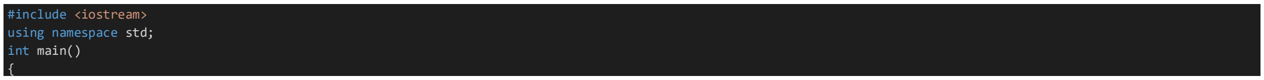
```
5
```

Expected Output:

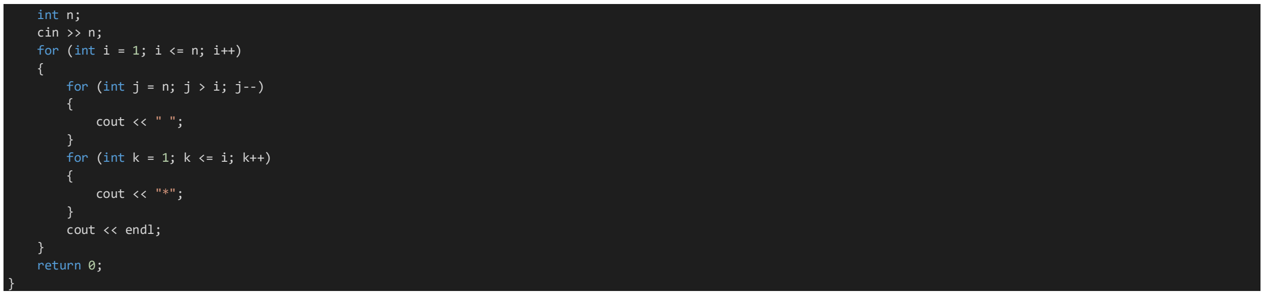
Received Output:

```
*
**
***
****
*****
```

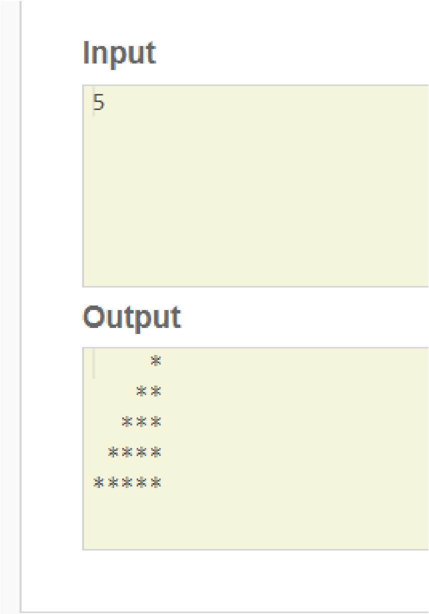
```
2.              *

               **

              ***

             ****

           ******
```



```cpp
#include <iostream>
using namespace std;
int main()
{
```



```
    int n;
    cin >> n;
    for (int i = 1; i <= n; i++)
    {
        for (int j = n; j > i; j--)
        {
            cout << " ";
        }
        for (int k = 1; k <= i; k++)
        {
            cout << "*";
        }
        cout << endl;
    }
    return 0;
}
```

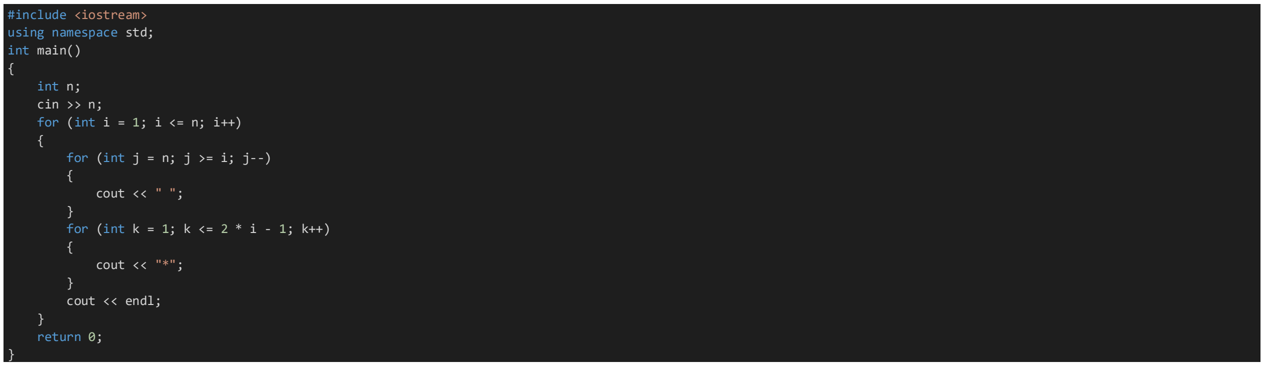


```
5
```

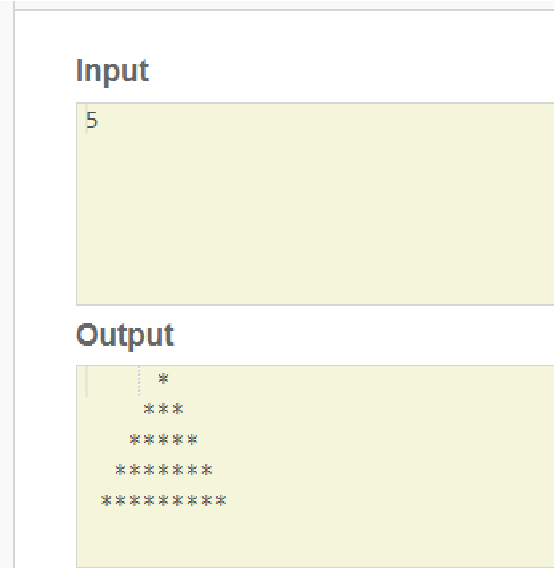
```
    *
   **
  ***
 ****
*****
```

3.

```
    *
   ***
  *****
 *******
*********
```



```
#include <iostream>
using namespace std;
int main()
{
    int n;
    cin >> n;
    for (int i = 1; i <= n; i++)
    {
        for (int j = n; j >= i; j--)
        {
            cout << " ";
        }
        for (int k = 1; k <= 2 * i - 1; k++)
        {
            cout << "*";
        }
        cout << endl;
    }
    return 0;
}
```



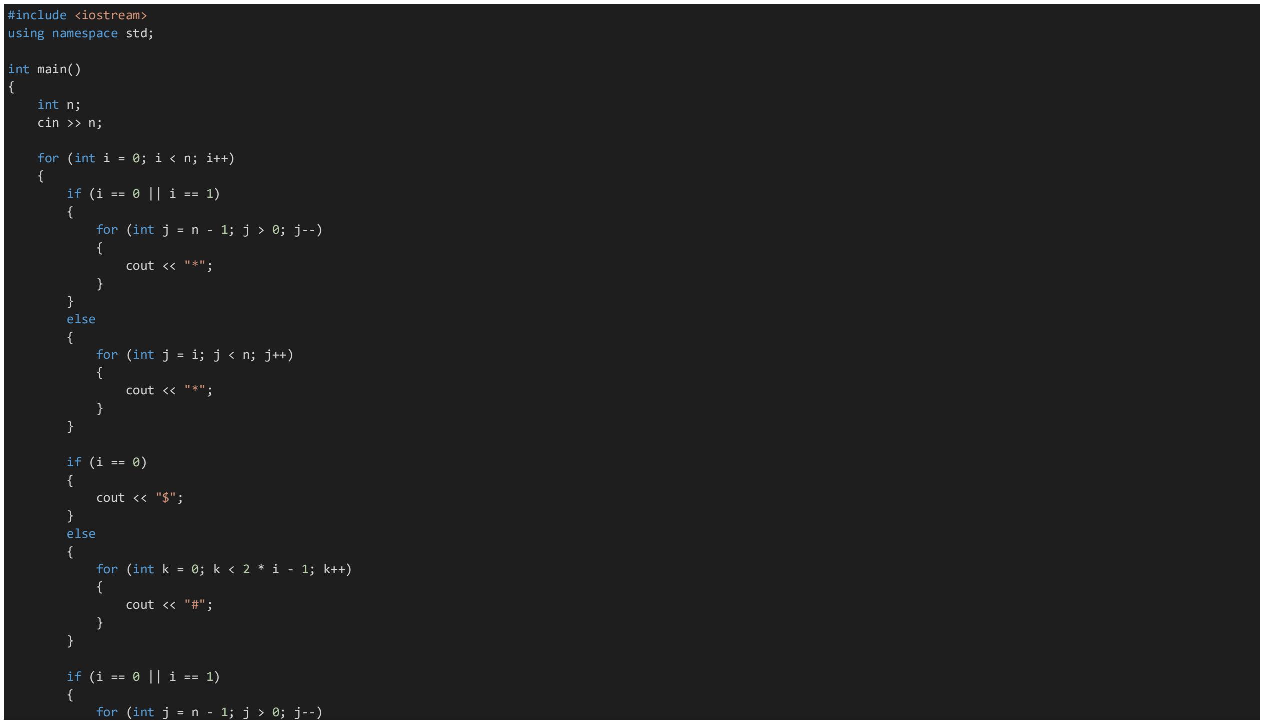
Input

```
5
```

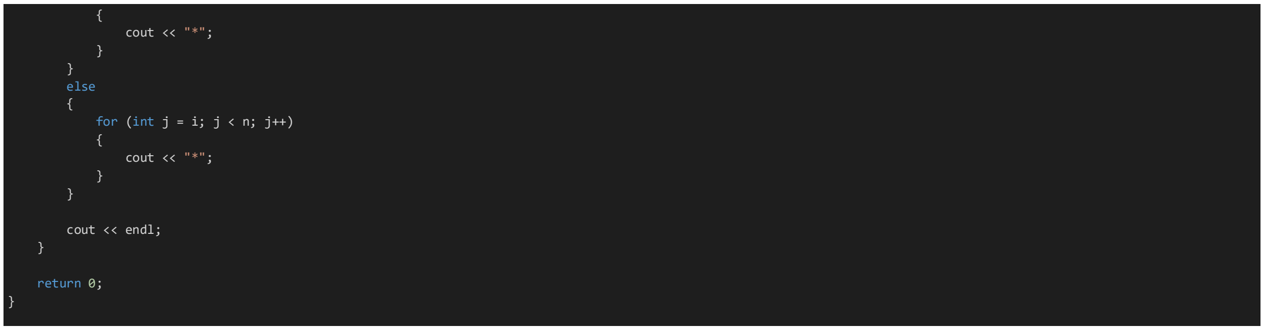
Output

```
    *
   ***
  *****
 *******
*********
```

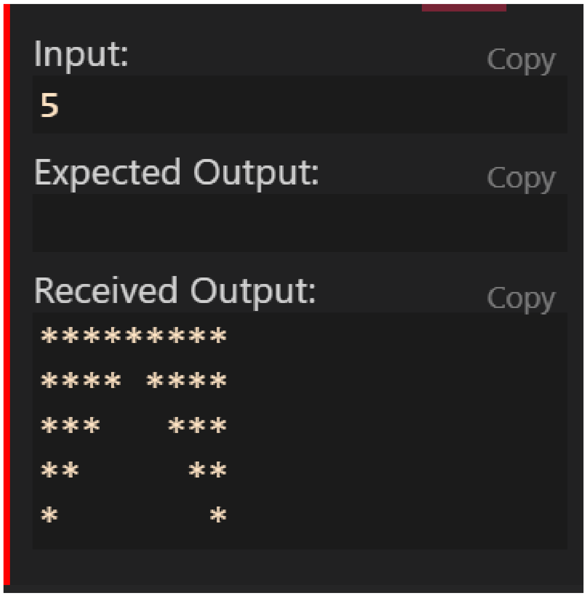
4.
```
*********

**** ****

***   ***

**     **

*       *
```



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

int main()
{
    int n;
    cin >> n;

    for (int i = 0; i < n; i++)
    {
        if (i == 0 || i == 1)
        {
            for (int j = n - 1; j > 0; j--)
            {
                cout << "*";
            }
        }
        else
        {
            for (int j = i; j < n; j++)
            {
                cout << "*";
            }
        }

        if (i == 0)
        {
            cout << "$";
        }
        else
        {
            for (int k = 0; k < 2 * i - 1; k++)
            {
                cout << "#";
            }
        }

        if (i == 0 || i == 1)
        {
            for (int j = n - 1; j > 0; j--)
```



```
            {
                cout << "*";
            }
        }
        else
        {
            for (int j = i; j < n; j++)
            {
                cout << "*";
            }
        }

        cout << endl;
    }

    return 0;
}
```



Input:

```
5
```

Expected Output:

Received Output:

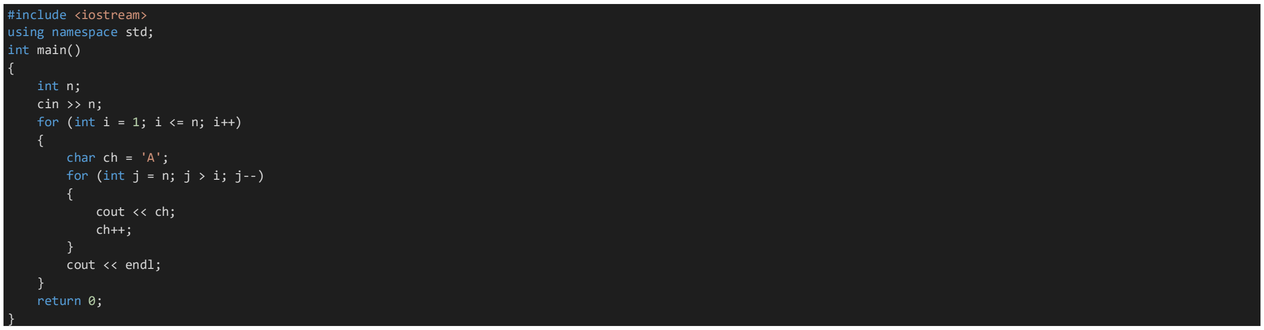
```
*********
**** ****
***   ***
**     **
*       *
```

5. ABCD

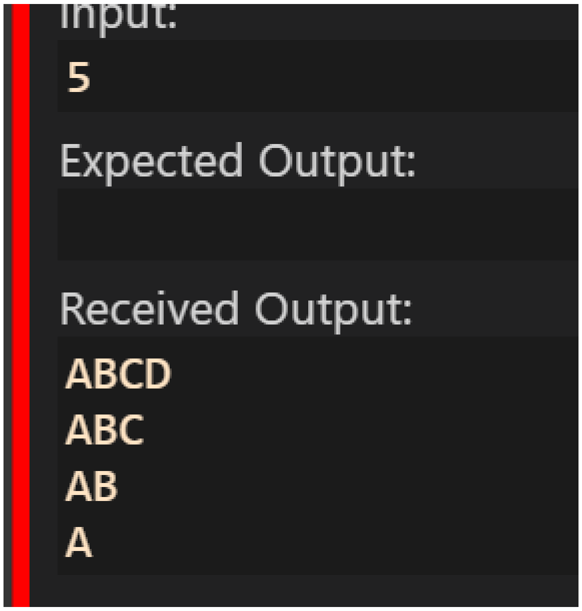
   ABC

   AB

   A



```
#include <iostream>
using namespace std;
int main()
{
    int n;
    cin >> n;
    for (int i = 1; i <= n; i++)
    {
        char ch = 'A';
        for (int j = n; j > i; j--)
        {
            cout << ch;
            ch++;
        }
        cout << endl;
    }
    return 0;
}
```



Input:

```
5
```

Expected Output:

Received Output:

```
ABCD
ABC
AB
A
```

6. 1

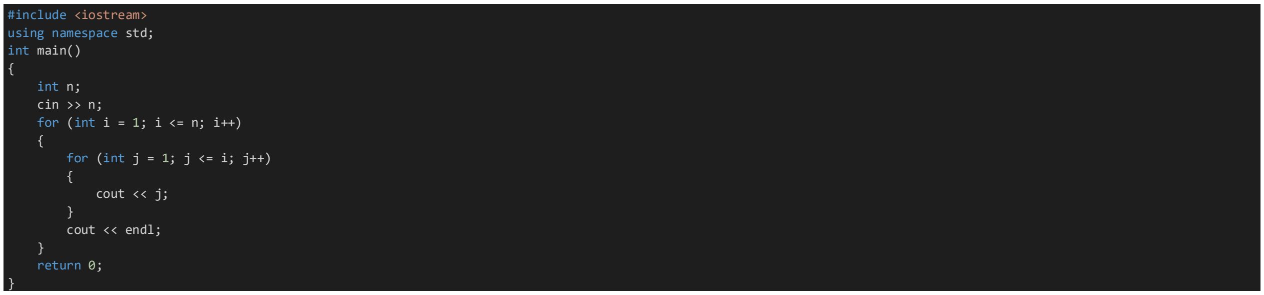
   12

   123

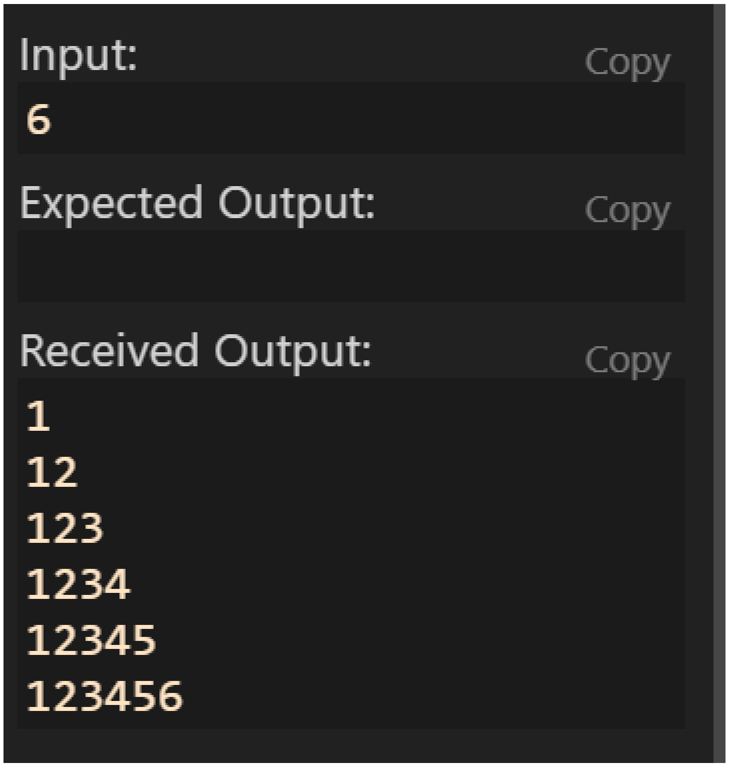
   1234

   12345

   123456



```
#include <iostream>
using namespace std;
int main()
{
    int n;
    cin >> n;
    for (int i = 1; i <= n; i++)
    {
        for (int j = 1; j <= i; j++)
        {
            cout << j;
        }
        cout << endl;
    }
    return 0;
}
```

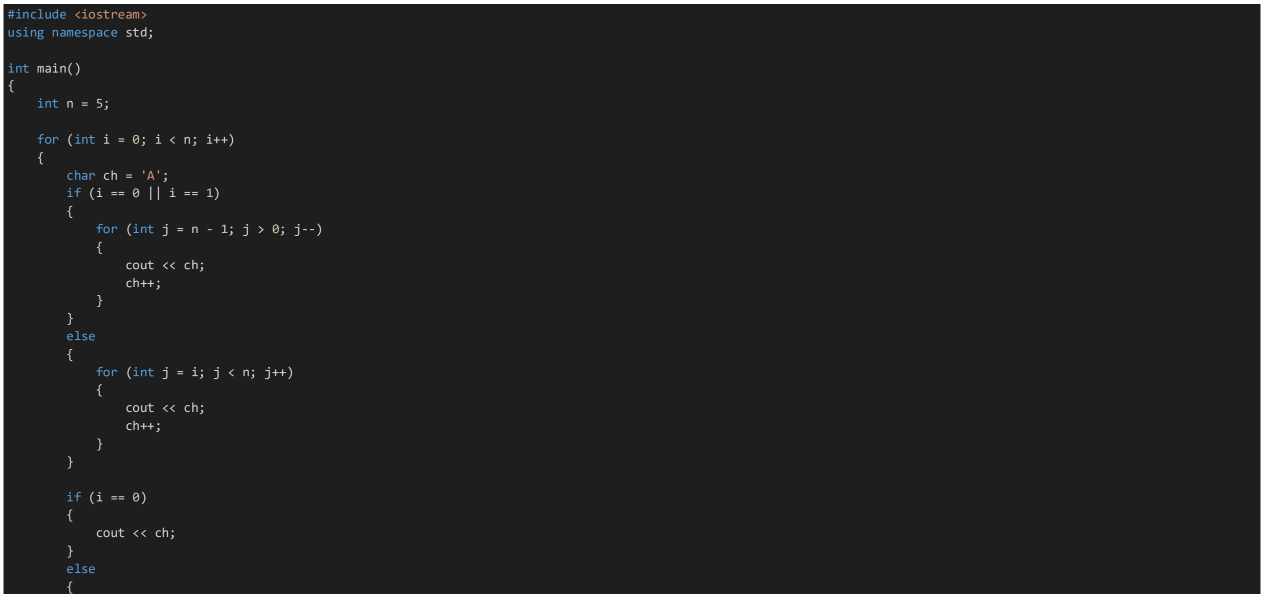


```
6
```

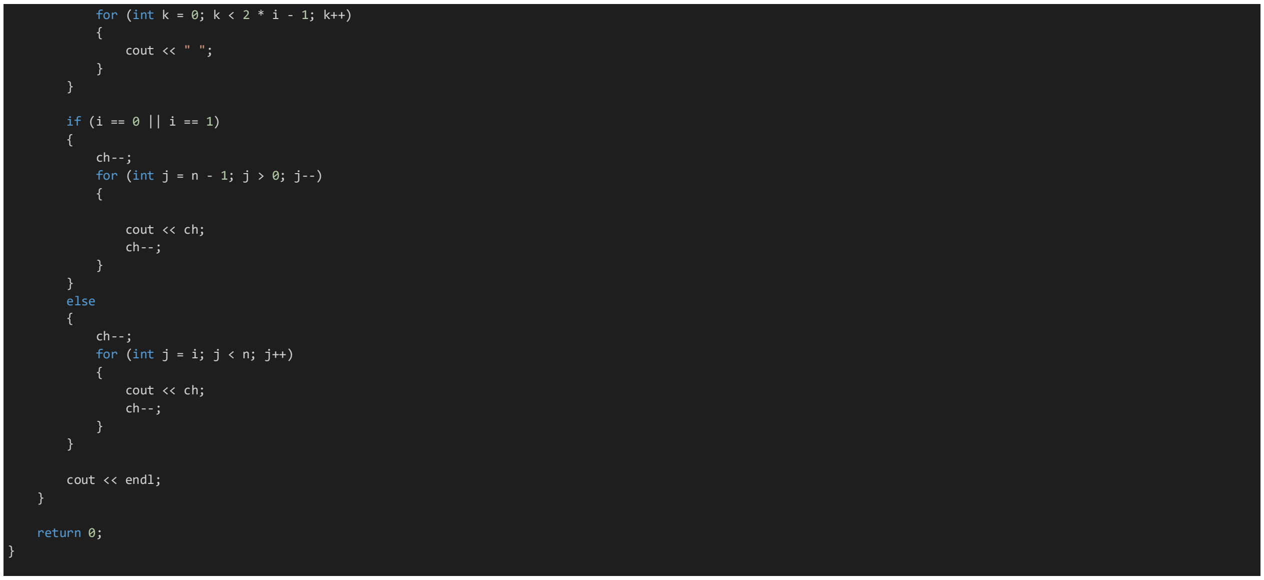
```
1
12
123
1234
12345
123456
```

7. ABCDEDCBA

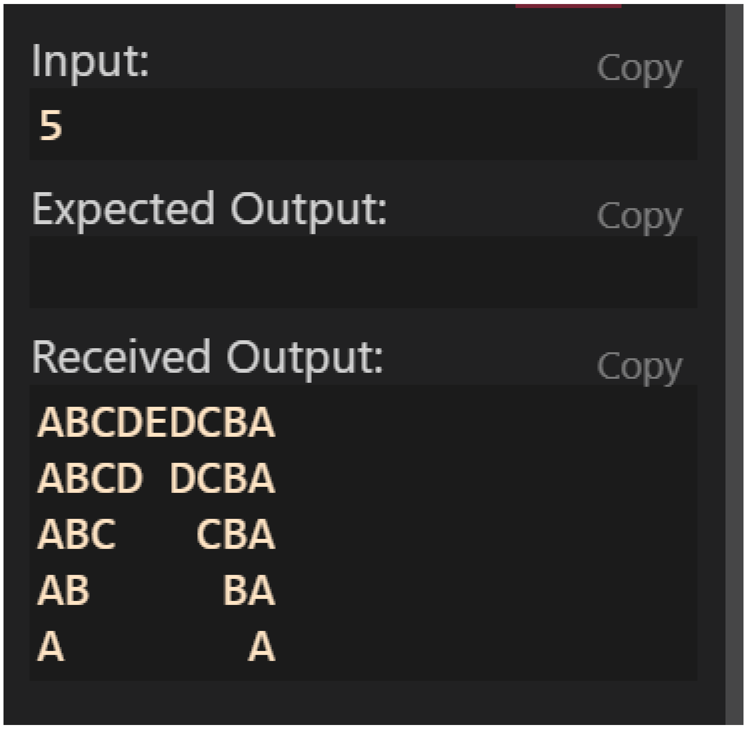
```
ABCD DCBA

ABC     CBA

AB          BA

A              A
```



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

int main()
{
    int n = 5;

    for (int i = 0; i < n; i++)
    {
        char ch = 'A';
        if (i == 0 || i == 1)
        {
            for (int j = n - 1; j > 0; j--)
            {
                cout << ch;
                ch++;
            }
        }
        else
        {
            for (int j = i; j < n; j++)
            {
                cout << ch;
                ch++;
            }
        }

        if (i == 0)
        {
            cout << ch;
        }
        else
        {
```



```
            for (int k = 0; k < 2 * i - 1; k++)
            {
                cout << " ";
            }
        }

        if (i == 0 || i == 1)
        {
            ch--;
            for (int j = n - 1; j > 0; j--)
            {

                cout << ch;
                ch--;
            }
        }
        else
        {
            ch--;
            for (int j = i; j < n; j++)
            {
                cout << ch;
                ch--;
            }
        }

        cout << endl;
    }

    return 0;
}
```



Input:

```
5
```

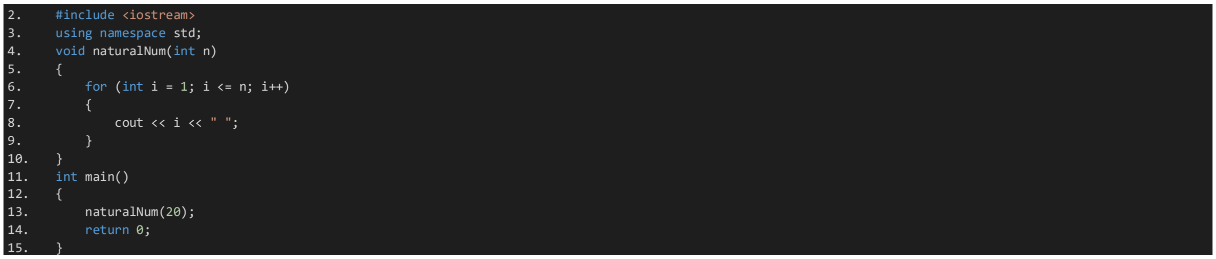
Expected Output:

Received Output:

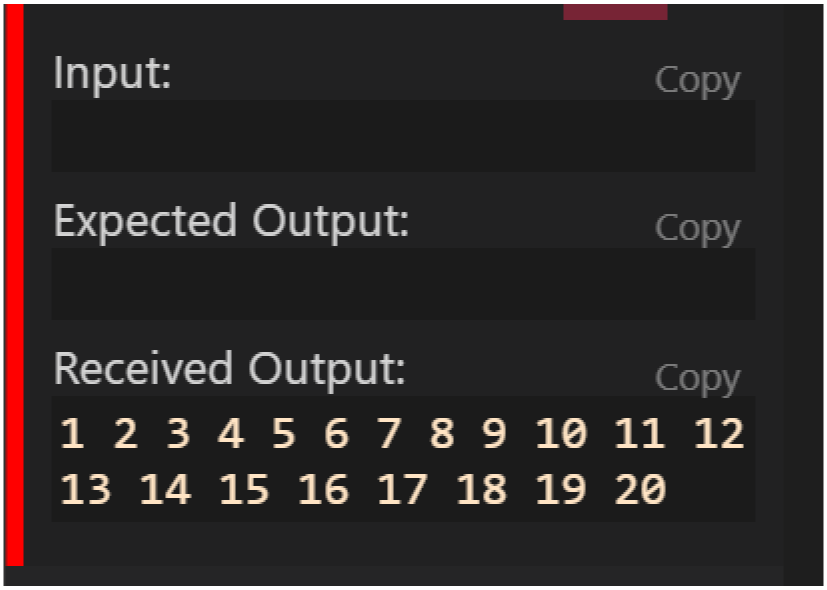
```
ABCDEDCBA
ABCD DCBA
ABC   CBA
AB     BA
A       A
```

C++ ASSIGNMENT 2

1. WAP for printing all natural numbers till 20.



```cpp
#include <iostream>
using namespace std;
void naturalNum(int n)
{
    for (int i = 1; i <= n; i++)
    {
        cout << i << " ";
    }
}
int main()
{
    naturalNum(20);
    return 0;
}
```



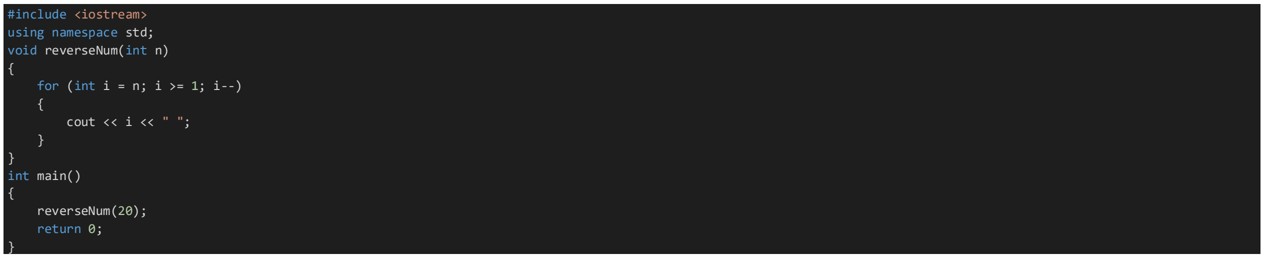
Input:

Expected Output:

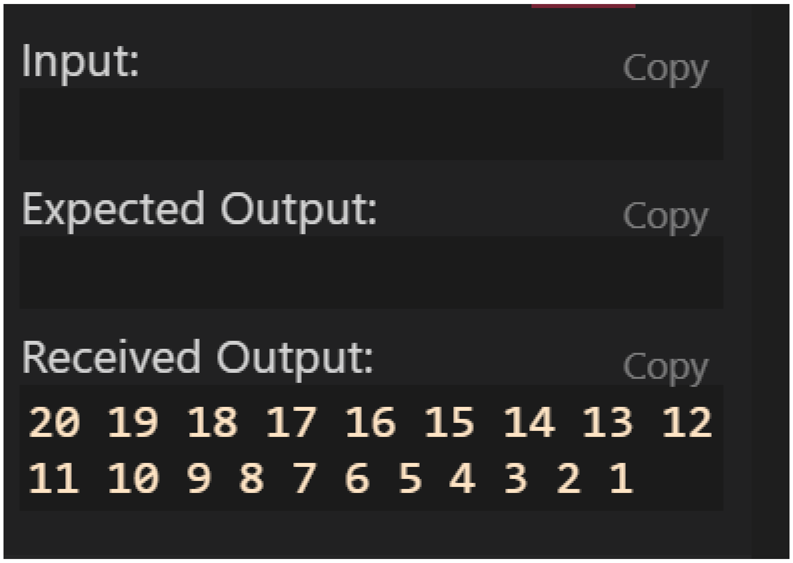
Received Output:

```
1 2 3 4 5 6 7 8 9 10 11 12
13 14 15 16 17 18 19 20
```

2. WAP for printing all natural numbers in reverse order starting from 20.

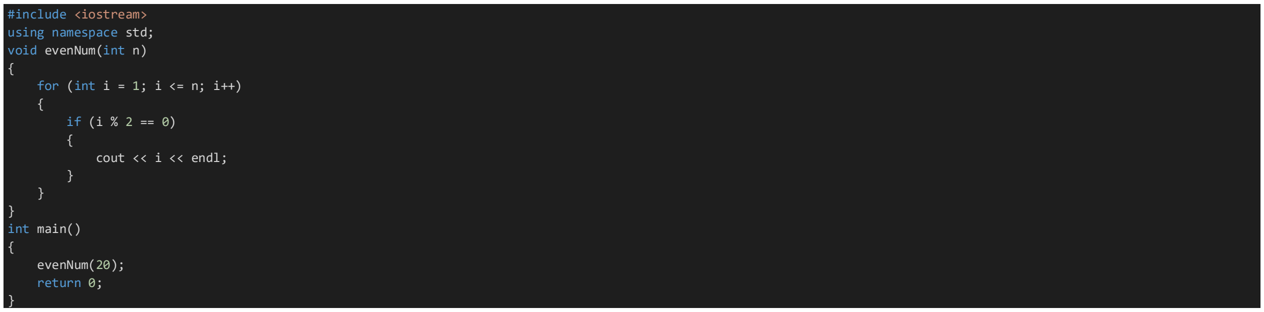


```cpp
#include <iostream>
using namespace std;
void reverseNum(int n)
{
    for (int i = n; i >= 1; i--)
    {
        cout << i << " ";
    }
}
int main()
{
    reverseNum(20);
    return 0;
}
```

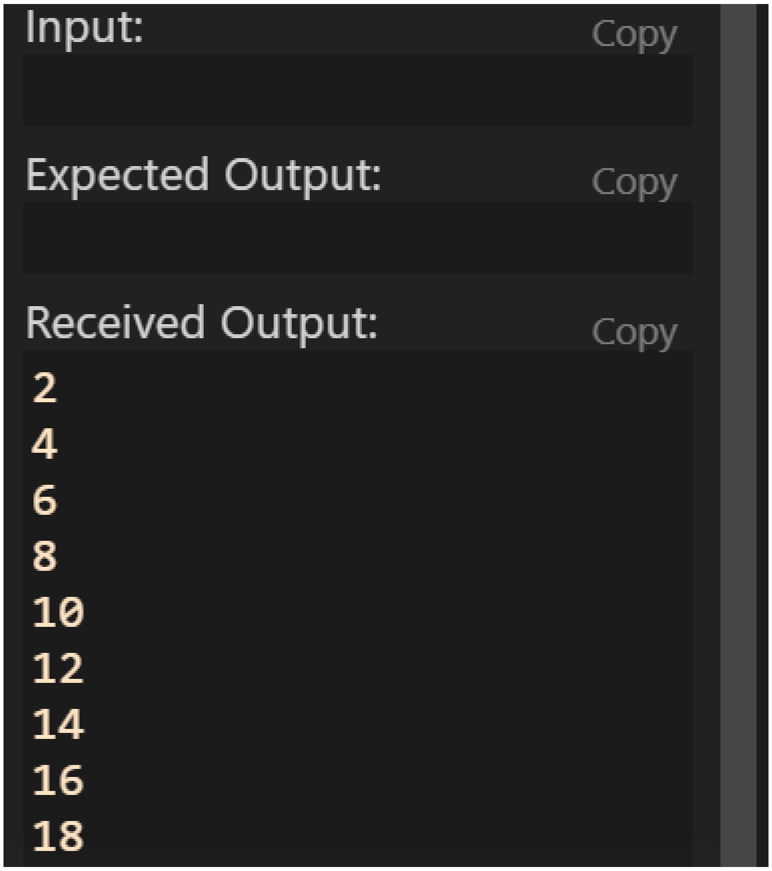


```
20 19 18 17 16 15 14 13 12
11 10 9 8 7 6 5 4 3 2 1
```

3. WAP for printing all even numbers from 1 to 20.

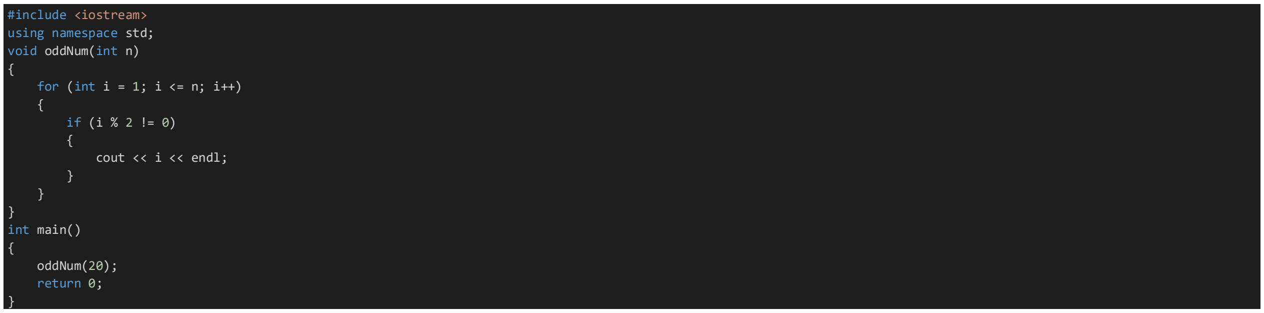


```cpp
#include <iostream>
using namespace std;
void evenNum(int n)
{
    for (int i = 1; i <= n; i++)
    {
        if (i % 2 == 0)
        {
            cout << i << endl;
        }
    }
}
int main()
{
    evenNum(20);
    return 0;
}
```

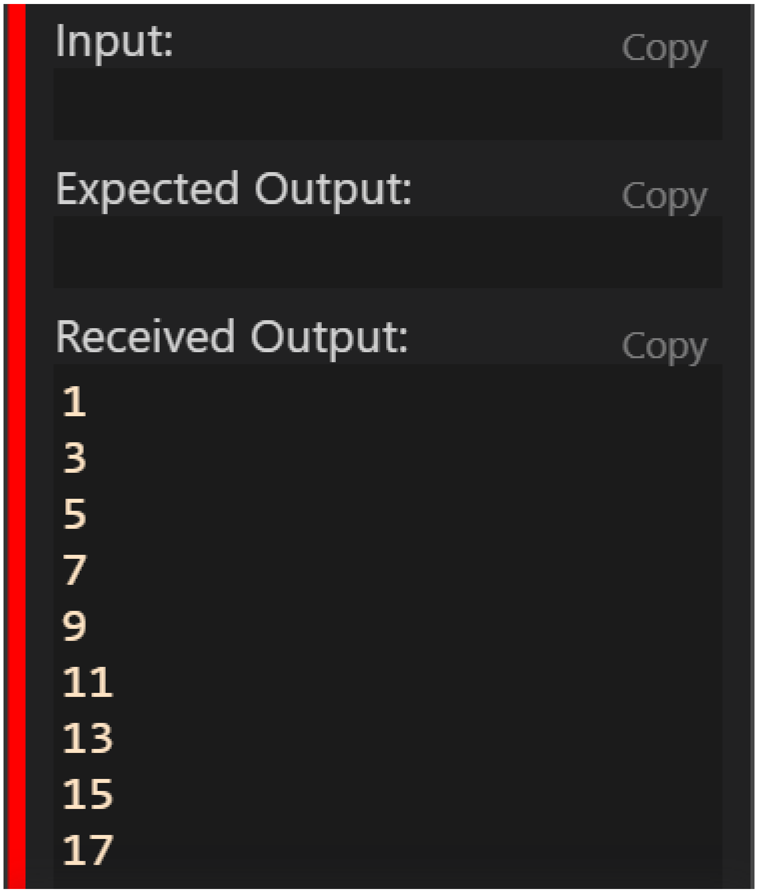


```
2
4
6
8
10
12
14
16
18
```

4. WAP for printing all odd numbers from 1 to 20.

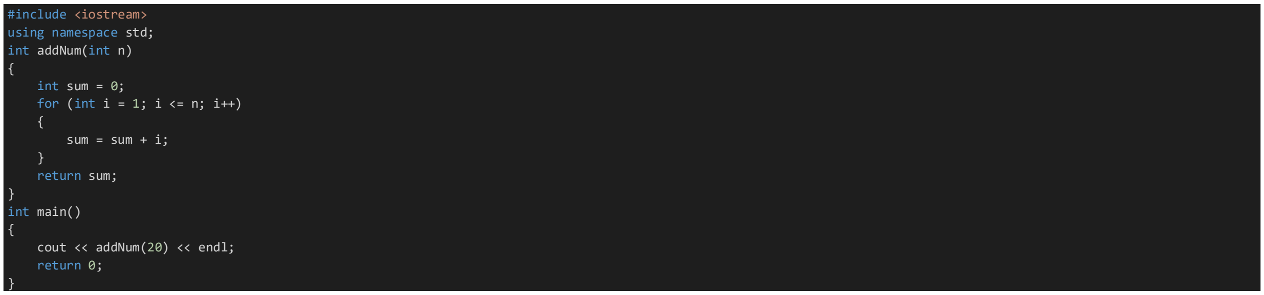


```
#include <iostream>
using namespace std;
void oddNum(int n)
{
    for (int i = 1; i <= n; i++)
    {
        if (i % 2 != 0)
        {
            cout << i << endl;
        }
    }
}
int main()
{
    oddNum(20);
    return 0;
}
```

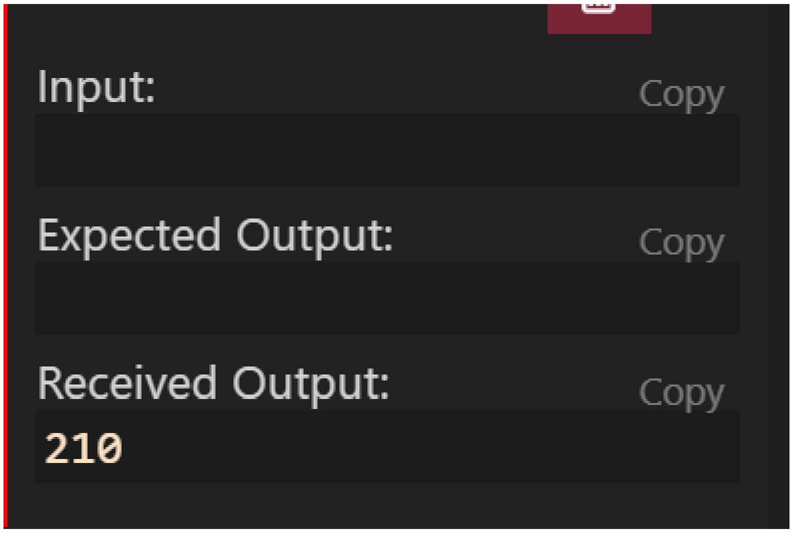


```
1
3
5
7
9
11
13
15
17
```

5. WAP for adding all numbers from 1 to 20.



```cpp
#include <iostream>
using namespace std;
int addNum(int n)
{
    int sum = 0;
    for (int i = 1; i <= n; i++)
    {
        sum = sum + i;
    }
    return sum;
}
int main()
{
    cout << addNum(20) << endl;
    return 0;
}
```

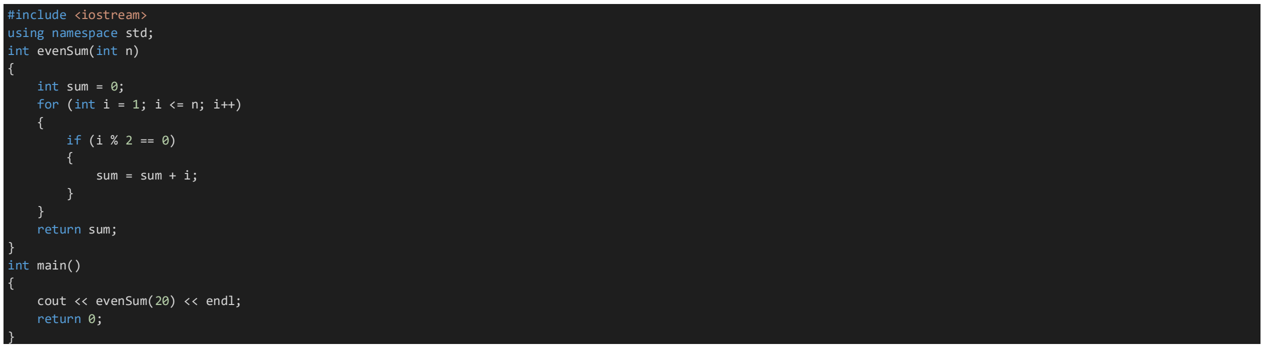

Input:

Expected Output:

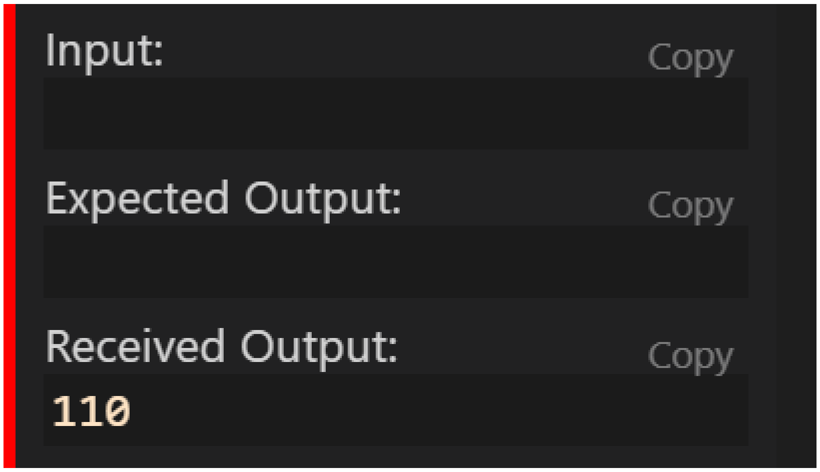
Received Output:

```
210
```

6. WAP for finding sum of all even numbers till 20.



```cpp
#include <iostream>
using namespace std;
int evenSum(int n)
{
    int sum = 0;
    for (int i = 1; i <= n; i++)
    {
        if (i % 2 == 0)
        {
            sum = sum + i;
        }
    }
    return sum;
}
int main()
{
    cout << evenSum(20) << endl;
    return 0;
}
```

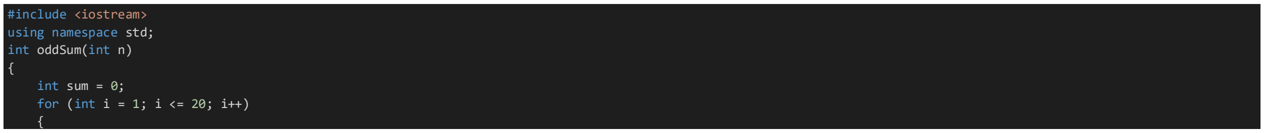

Input:

Expected Output:

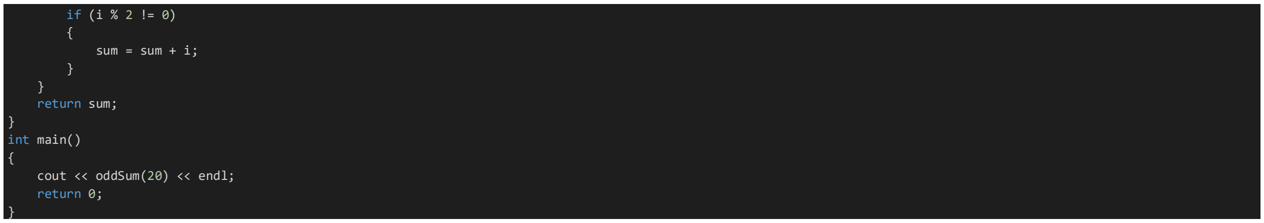
Received Output:

```
110
```

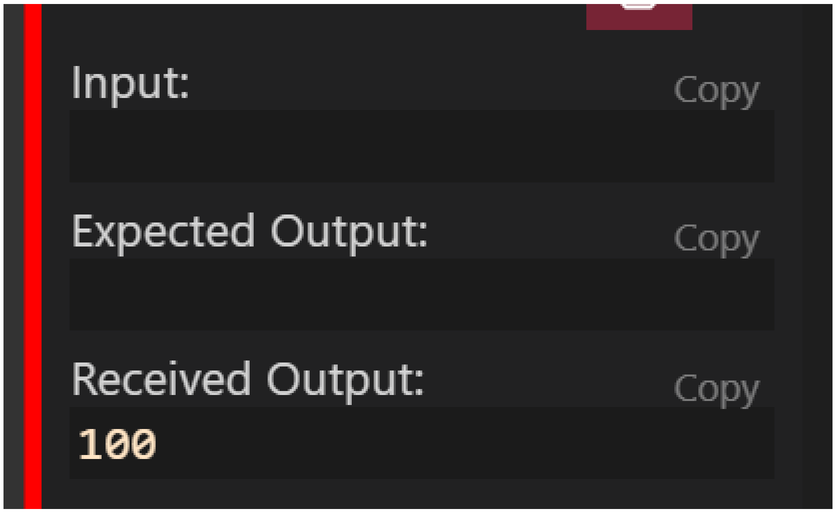
7. WAP for finding sum of all odd numbers till 20.



```cpp
#include <iostream>
using namespace std;
int oddSum(int n)
{
    int sum = 0;
    for (int i = 1; i <= 20; i++)
    {
```



```
        if (i % 2 != 0)
        {
            sum = sum + i;
        }
    }
    return sum;
}
int main()
{
    cout << oddSum(20) << endl;
    return 0;
}
```



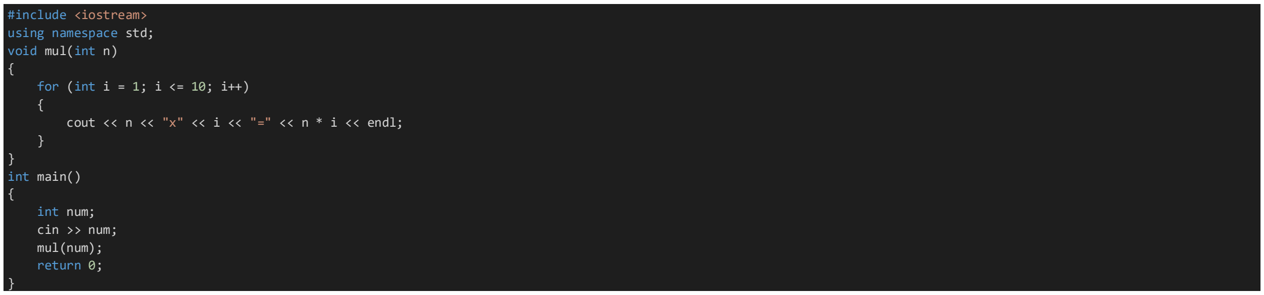
Input:

Expected Output:

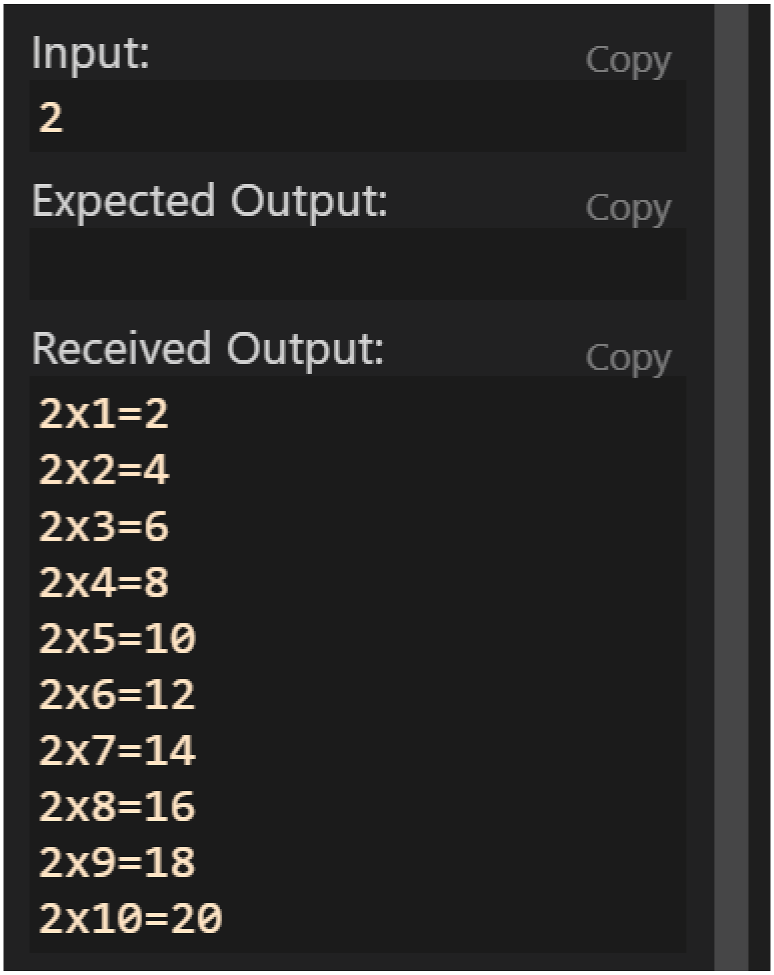
Received Output:

```
100
```

8. WAP for printing multiplication table of a number. For eg. Display should be “ 2 X 1 = 2”



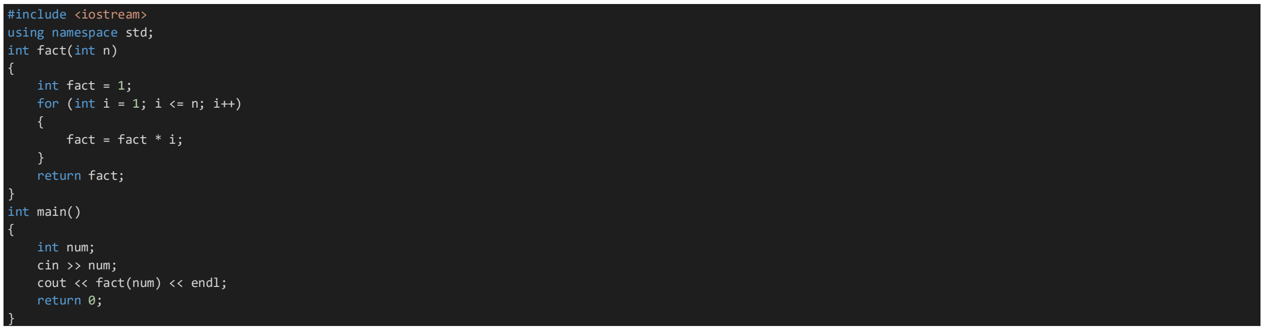
```
#include <iostream>
using namespace std;
void mul(int n)
{
    for (int i = 1; i <= 10; i++)
    {
        cout << n << "x" << i << "=" << n * i << endl;
    }
}
int main()
{
    int num;
    cin >> num;
    mul(num);
    return 0;
}
```



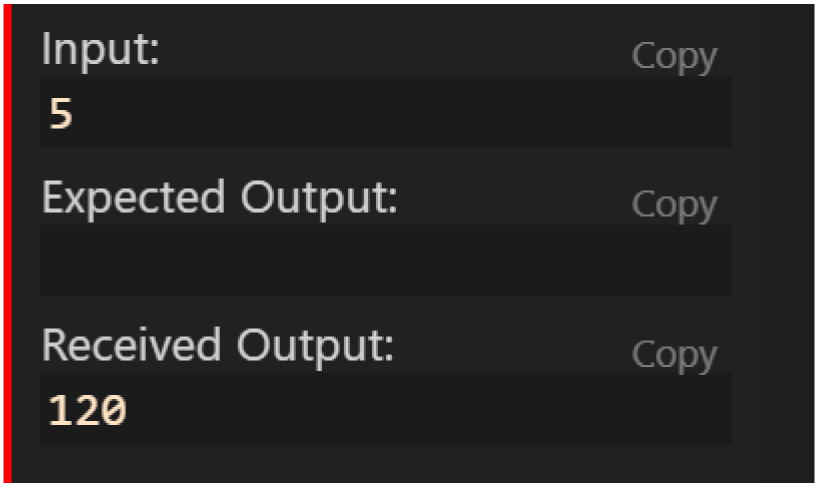
```
2
```

```
2x1=2
2x2=4
2x3=6
2x4=8
2x5=10
2x6=12
2x7=14
2x8=16
2x9=18
2x10=20
```

9. WAP to calculate factorial of a number.



```cpp
#include <iostream>
using namespace std;
int fact(int n)
{
    int fact = 1;
    for (int i = 1; i <= n; i++)
    {
        fact = fact * i;
    }
    return fact;
}
int main()
{
    int num;
    cin >> num;
    cout << fact(num) << endl;
    return 0;
}
```



Input:

```
5
```

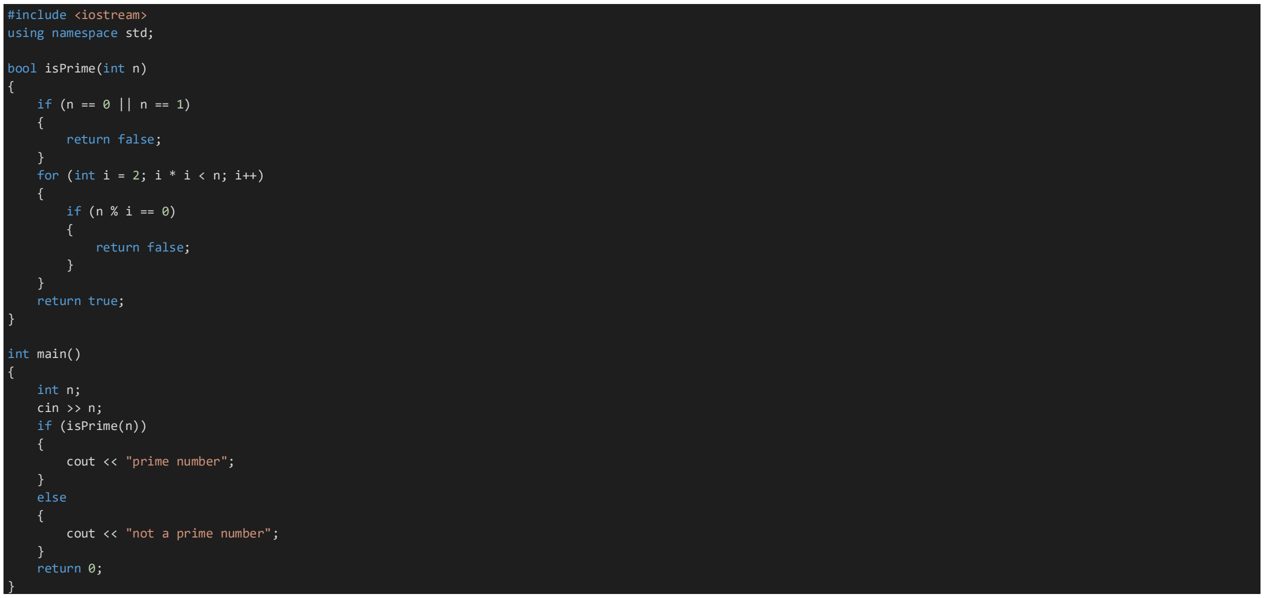
Expected Output:

```
```

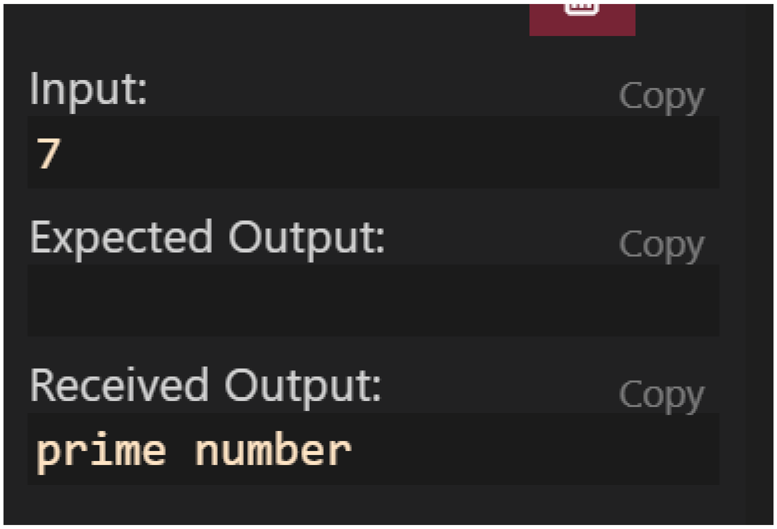
Received Output:

```
120
```

10. WAP to check whether a number is prime or not.



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

bool isPrime(int n)
{
    if (n == 0 || n == 1)
    {
        return false;
    }
    for (int i = 2; i * i < n; i++)
    {
        if (n % i == 0)
        {
            return false;
        }
    }
    return true;
}

int main()
{
    int n;
    cin >> n;
    if (isPrime(n))
    {
        cout << "prime number";
    }
    else
    {
        cout << "not a prime number";
    }
    return 0;
}
```



Input:

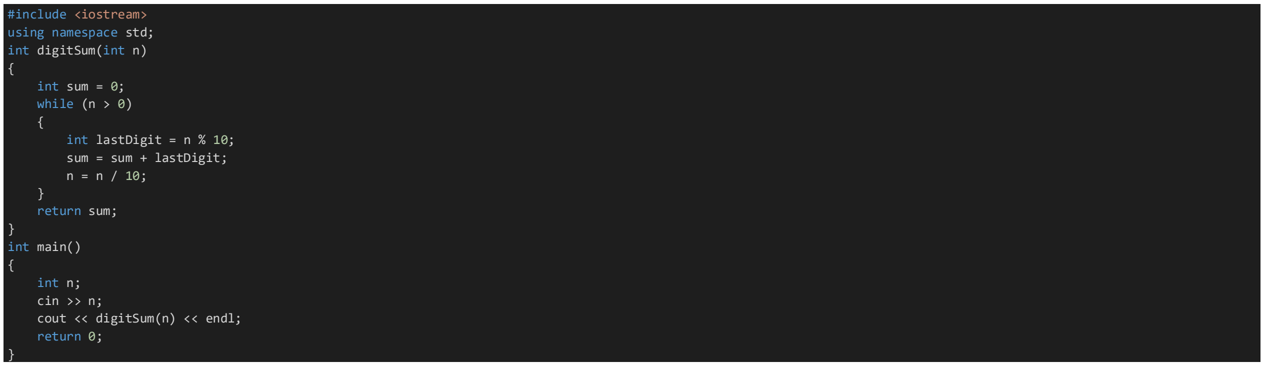
```
7
```

Expected Output:

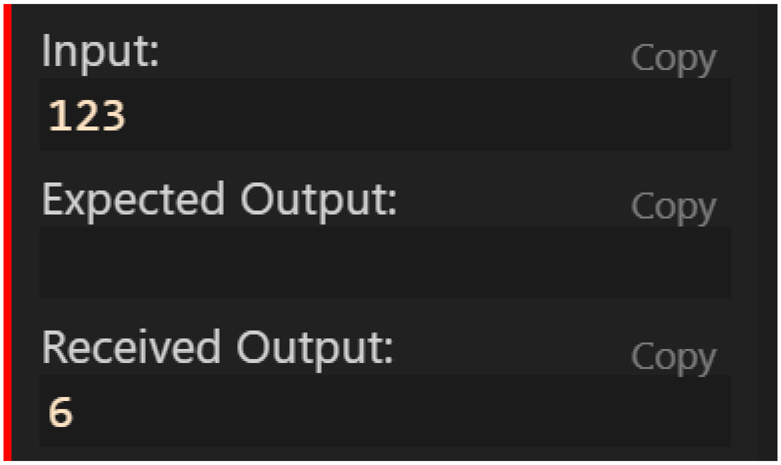
Received Output:

```
prime number
```

11. WAP to print all digits of a number and their sum.



```cpp
#include <iostream>
using namespace std;
int digitSum(int n)
{
    int sum = 0;
    while (n > 0)
    {
        int lastDigit = n % 10;
        sum = sum + lastDigit;
        n = n / 10;
    }
    return sum;
}
int main()
{
    int n;
    cin >> n;
    cout << digitSum(n) << endl;
    return 0;
}
```



Input:

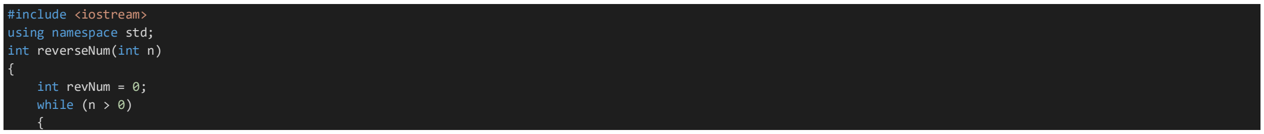
```
123
```

Expected Output:

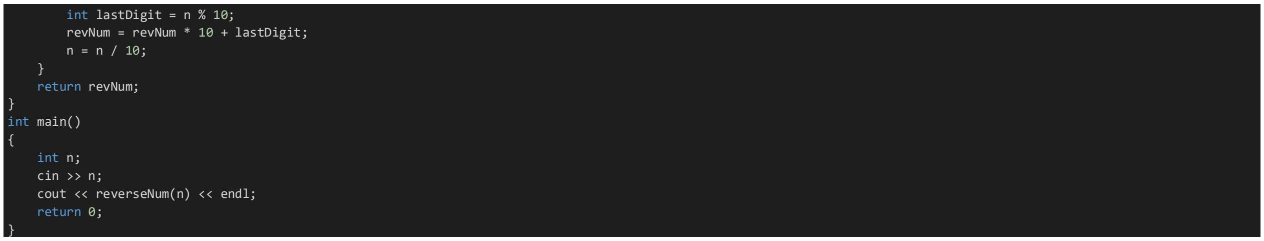
Received Output:

```
6
```

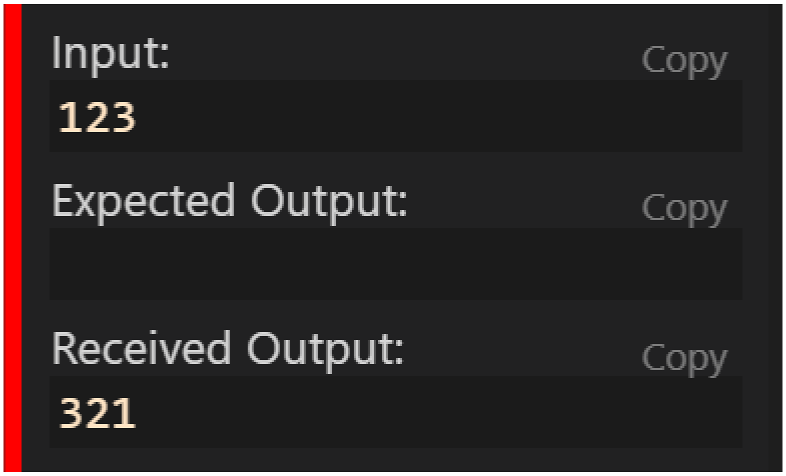
12. WAP to print reverse of a number.



```cpp
#include <iostream>
using namespace std;
int reverseNum(int n)
{
    int revNum = 0;
    while (n > 0)
    {
```



```
        int lastDigit = n % 10;
        revNum = revNum * 10 + lastDigit;
        n = n / 10;
    }
    return revNum;
}
int main()
{
    int n;
    cin >> n;
    cout << reverseNum(n) << endl;
    return 0;
}
```



Input:

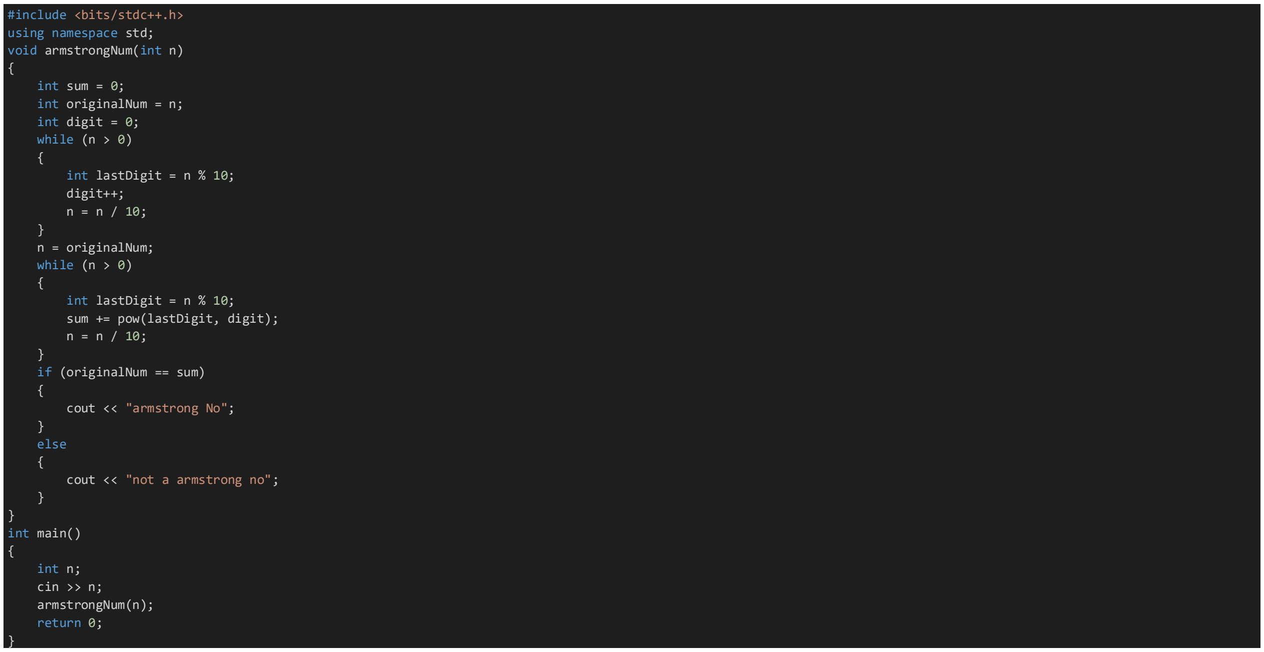
```
123
```

Expected Output:

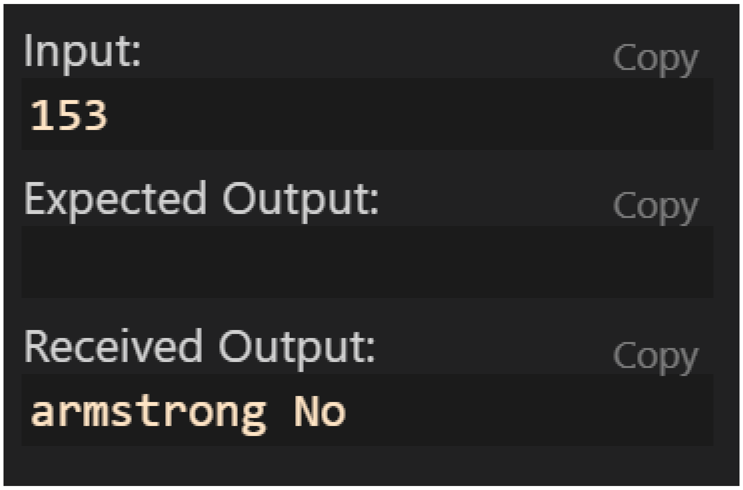
Received Output:

```
321
```

13. WAP to check whether the number is Armstrong or not.



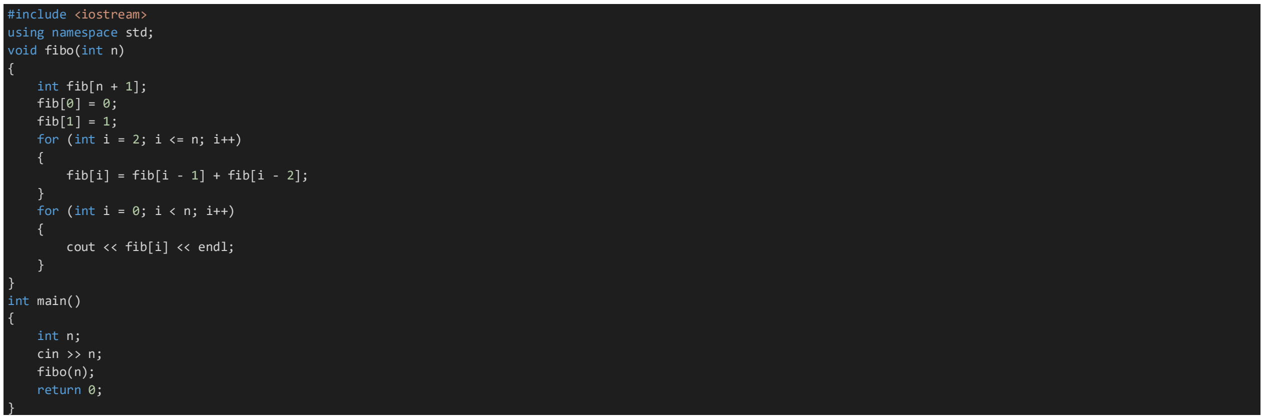
```
#include <bits/stdc++.h>
using namespace std;
void armstrongNum(int n)
{
    int sum = 0;
    int originalNum = n;
    int digit = 0;
    while (n > 0)
    {
        int lastDigit = n % 10;
        digit++;
        n = n / 10;
    }
    n = originalNum;
    while (n > 0)
    {
        int lastDigit = n % 10;
        sum += pow(lastDigit, digit);
        n = n / 10;
    }
    if (originalNum == sum)
    {
        cout << "armstrong No";
    }
    else
    {
        cout << "not a armstrong no";
    }
}
int main()
{
    int n;
    cin >> n;
    armstrongNum(n);
    return 0;
}
```



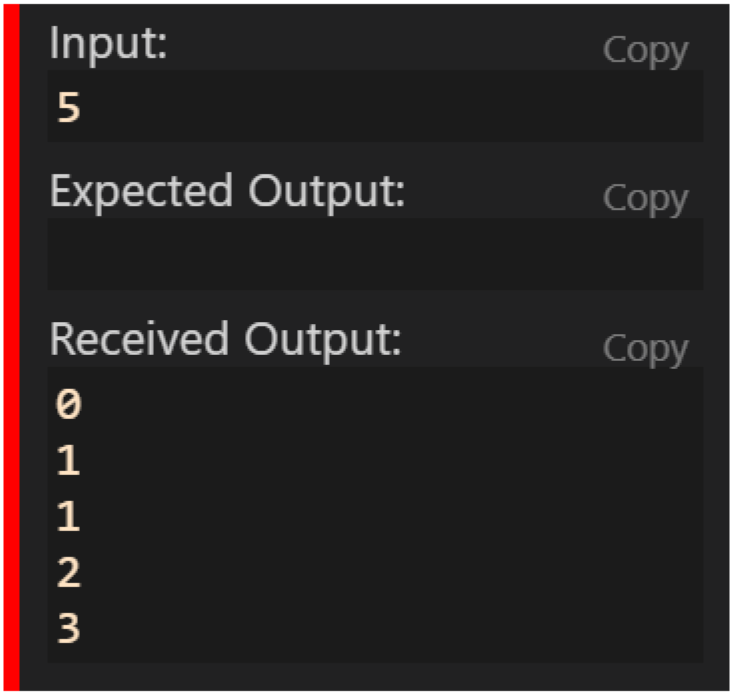
```
153
```

```
armstrong No
```

14. WAP to print the Fibonacci series in a given range.



```
#include <iostream>
using namespace std;
void fibo(int n)
{
    int fib[n + 1];
    fib[0] = 0;
    fib[1] = 1;
    for (int i = 2; i <= n; i++)
    {
        fib[i] = fib[i - 1] + fib[i - 2];
    }
    for (int i = 0; i < n; i++)
    {
        cout << fib[i] << endl;
    }
}
int main()
{
    int n;
    cin >> n;
    fibo(n);
    return 0;
}
```



Input: Copy

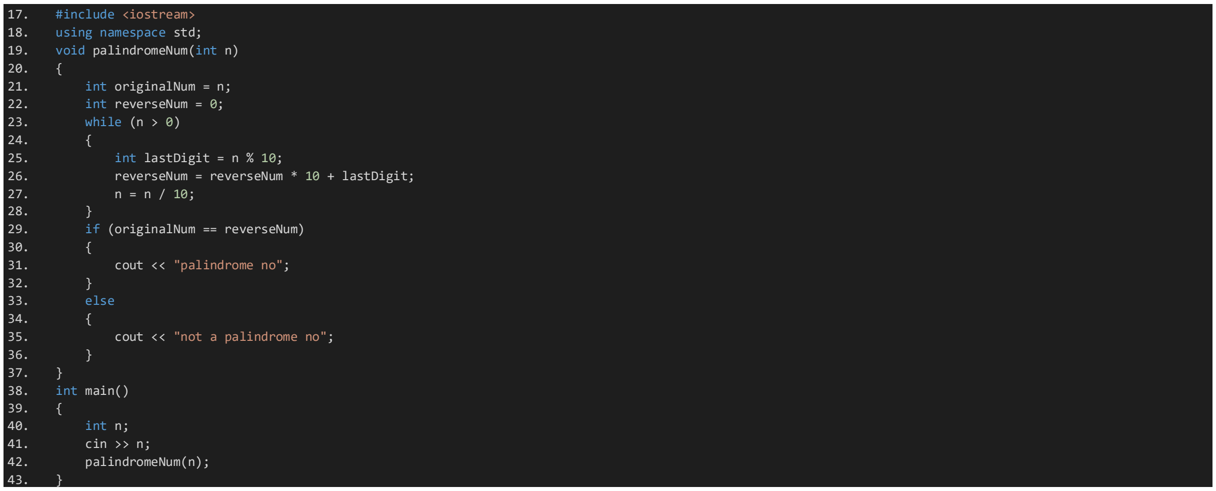
```
5
```

Expected Output: Copy

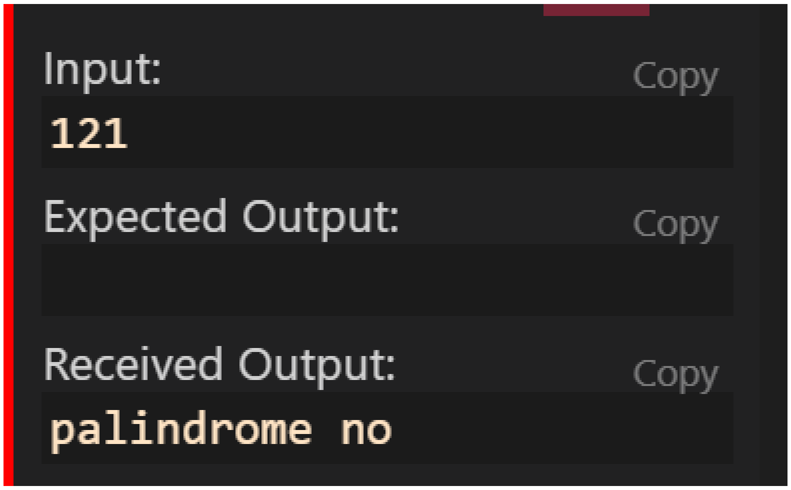
Received Output: Copy

```
0
1
1
2
3
```

16. WAP to check whether the number entered is palindrome or not.



```
#include <iostream>
using namespace std;
void palindromeNum(int n)
{
    int originalNum = n;
    int reverseNum = 0;
    while (n > 0)
    {
        int lastDigit = n % 10;
        reverseNum = reverseNum * 10 + lastDigit;
        n = n / 10;
    }
    if (originalNum == reverseNum)
    {
        cout << "palindrome no";
    }
    else
    {
        cout << "not a palindrome no";
    }
}
int main()
{
    int n;
    cin >> n;
    palindromeNum(n);
}
```



Input:

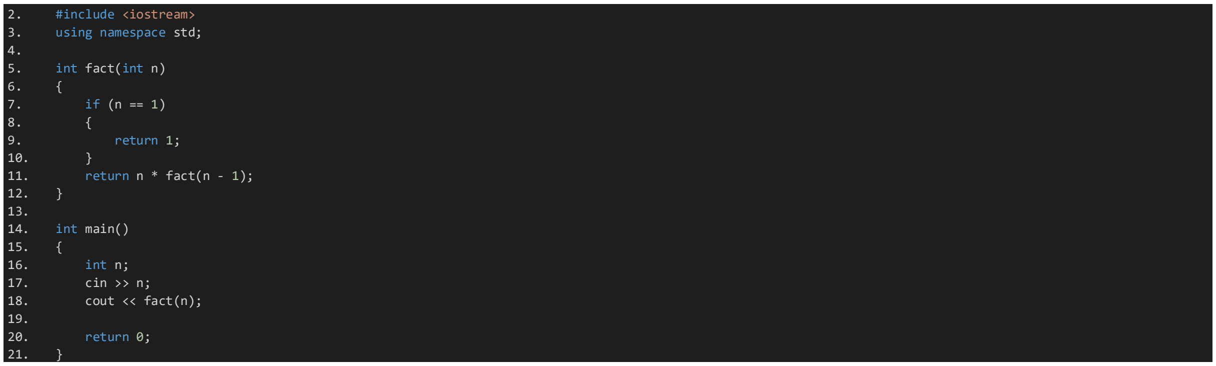
```
121
```

Expected Output:

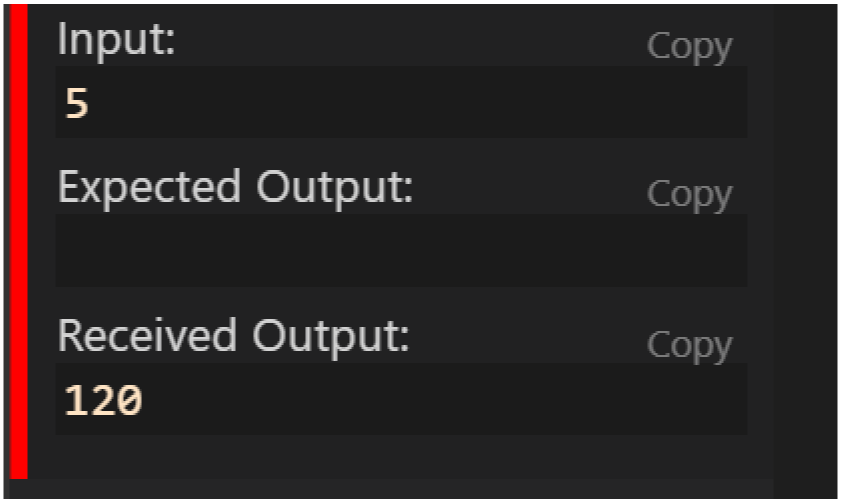
Received Output:

```
palindrome no
```

1. WAP to calculate factorial of a number.

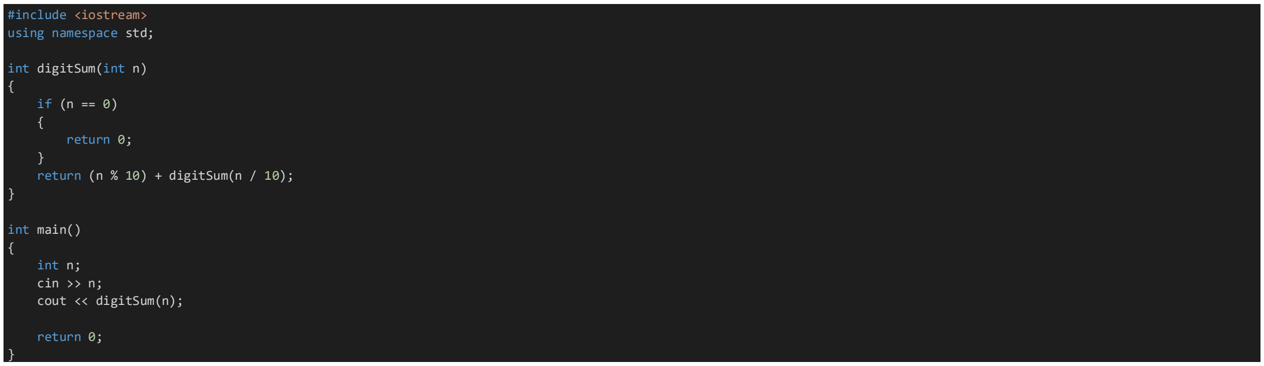


```
#include <iostream>
using namespace std;

int fact(int n)
{
    if (n == 1)
    {
        return 1;
    }
    return n * fact(n - 1);
}

int main()
{
    int n;
    cin >> n;
    cout << fact(n);

    return 0;
}
```

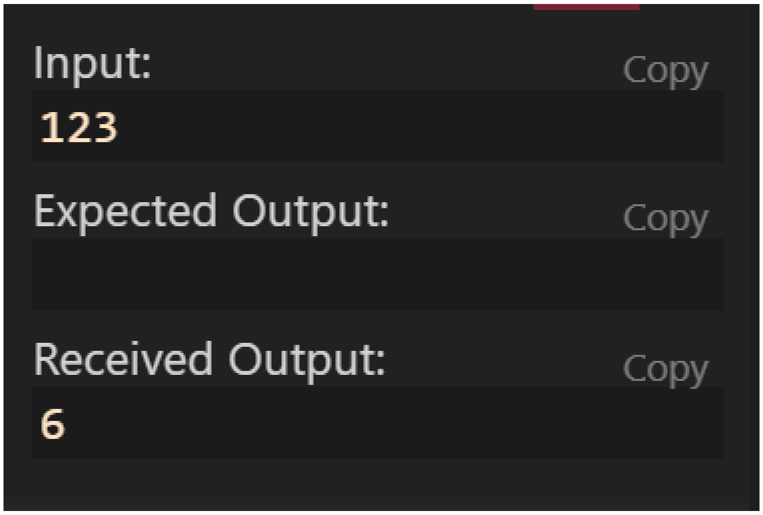


```
Input:
5
Expected Output:

Received Output:
120
```

2. WAP to print all digits of a number and their sum.



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

int digitSum(int n)
{
    if (n == 0)
    {
        return 0;
    }
    return (n % 10) + digitSum(n / 10);
}

int main()
{
    int n;
    cin >> n;
    cout << digitSum(n);

    return 0;
}
```



Input:

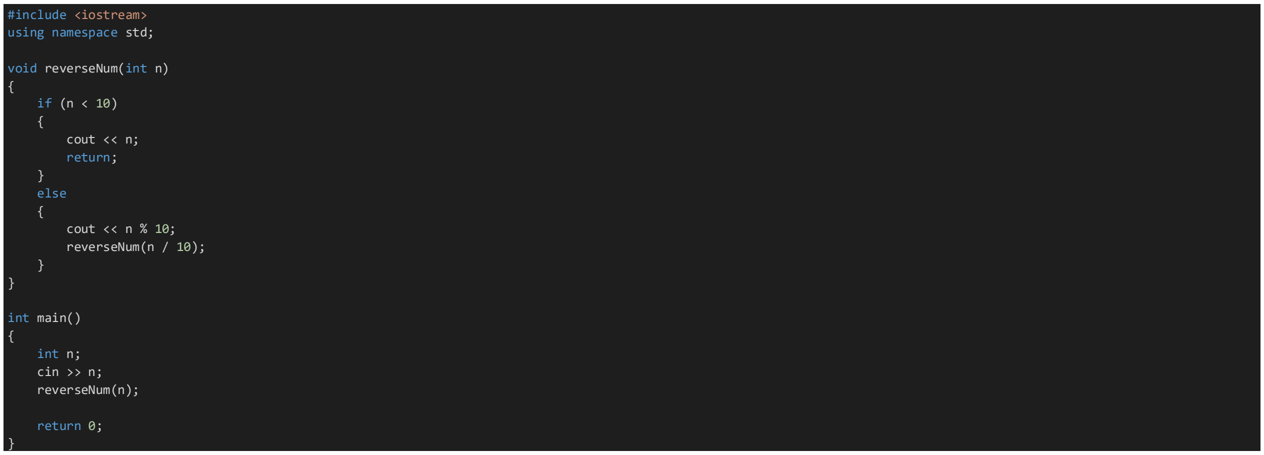
```
123
```

Expected Output:

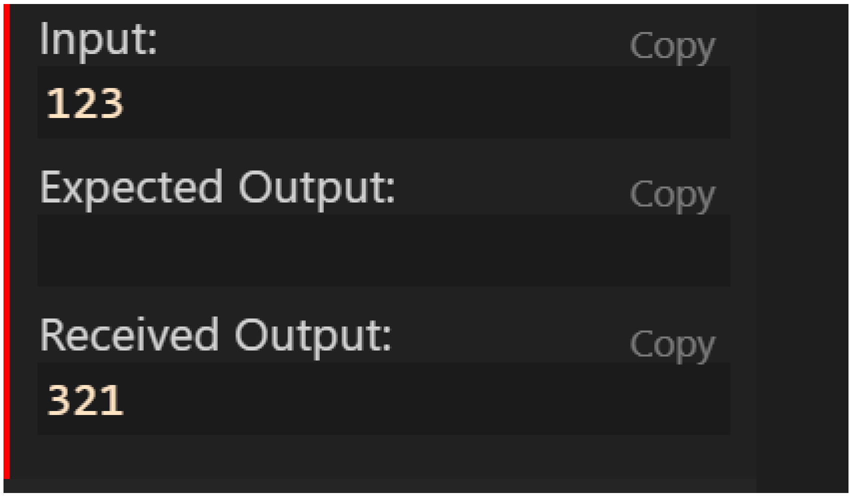
Received Output:

```
6
```

3. WAP to print reverse of a number.



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

void reverseNum(int n)
{
    if (n < 10)
    {
        cout << n;
        return;
    }
    else
    {
        cout << n % 10;
        reverseNum(n / 10);
    }
}

int main()
{
    int n;
    cin >> n;
    reverseNum(n);

    return 0;
}
```



Input:

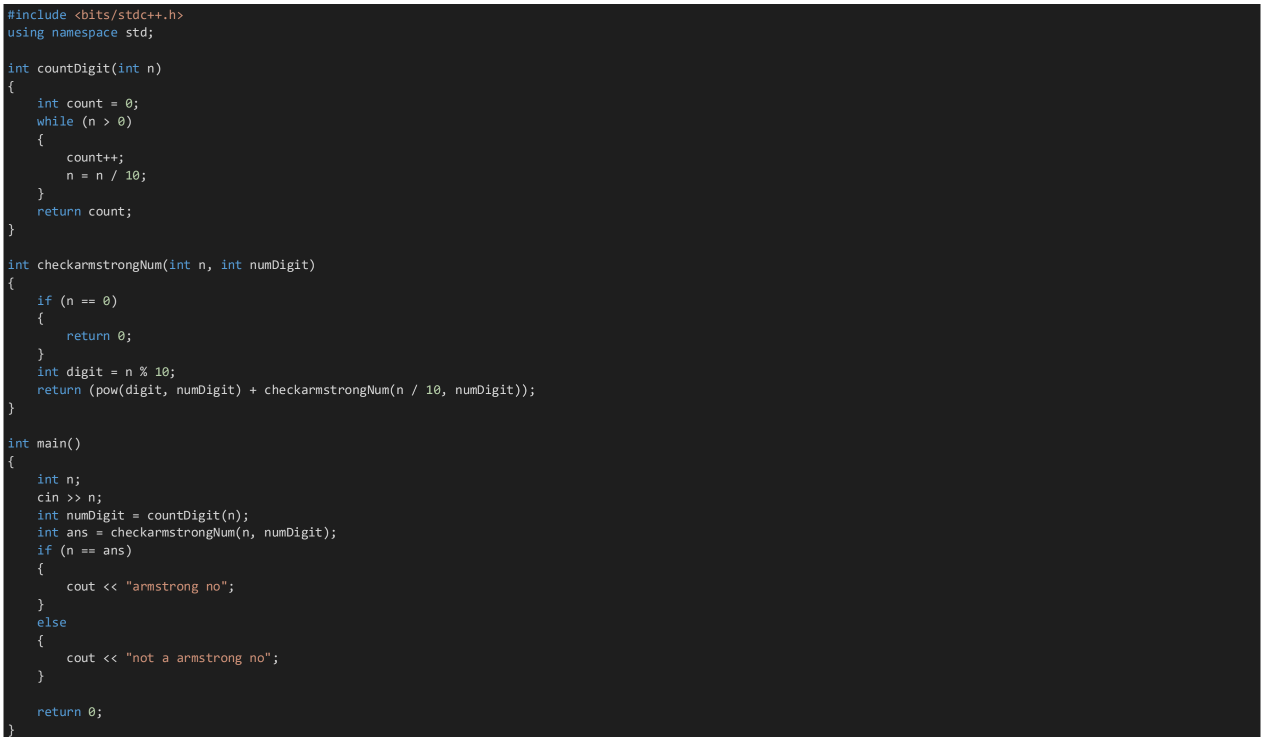
```
123
```

Expected Output:

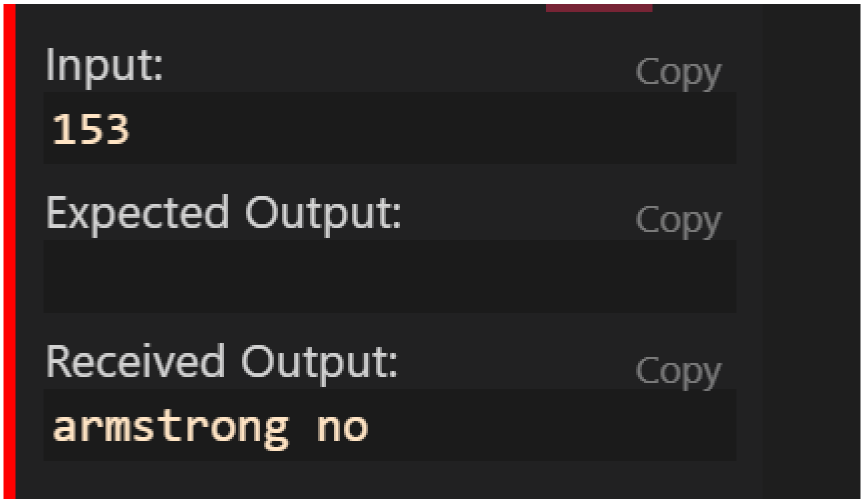
Received Output:

```
321
```

4. WAP to check whether the number is Armstrong or not.



```cpp
#include <bits/stdc++.h>
using namespace std;

int countDigit(int n)
{
    int count = 0;
    while (n > 0)
    {
        count++;
        n = n / 10;
    }
    return count;
}

int checkarmstrongNum(int n, int numDigit)
{
    if (n == 0)
    {
        return 0;
    }
    int digit = n % 10;
    return (pow(digit, numDigit) + checkarmstrongNum(n / 10, numDigit));
}

int main()
{
    int n;
    cin >> n;
    int numDigit = countDigit(n);
    int ans = checkarmstrongNum(n, numDigit);
    if (n == ans)
    {
        cout << "armstrong no";
    }
    else
    {
        cout << "not a armstrong no";
    }

    return 0;
}
```



Input:

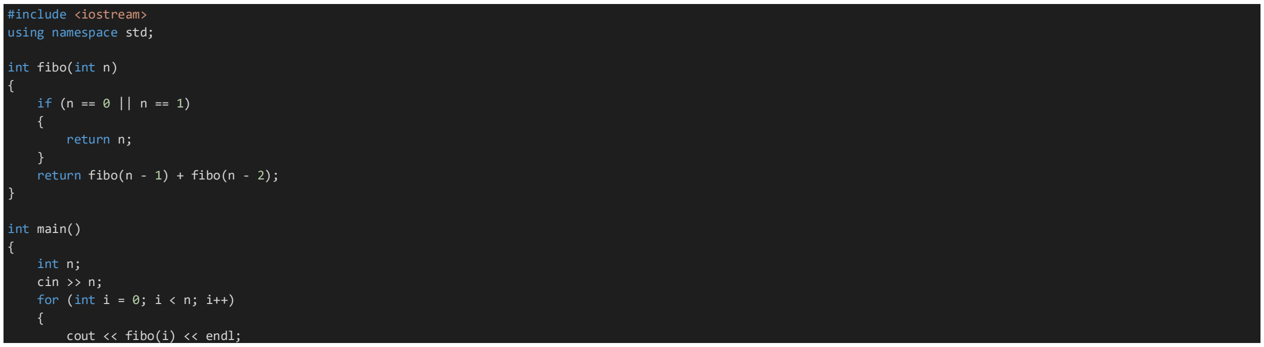
```
153
```

Expected Output:

Received Output:

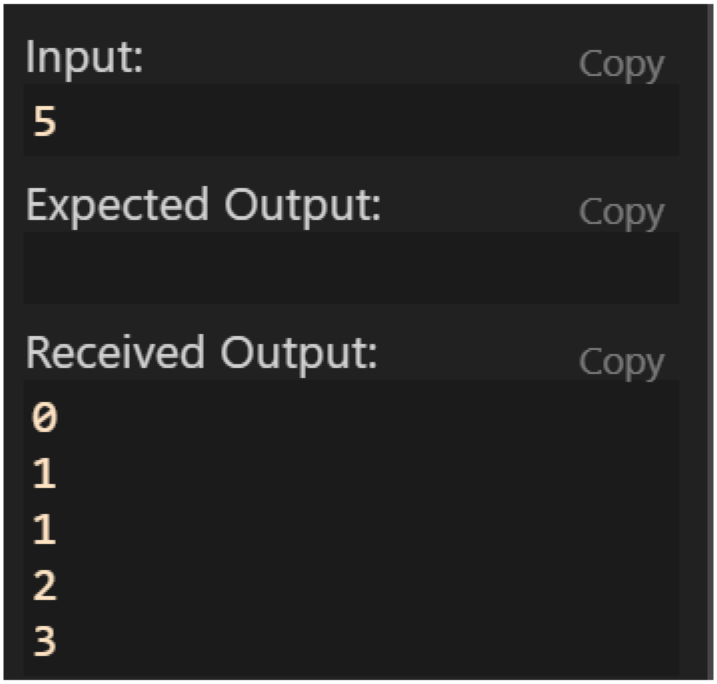
```
armstrong no
```

5. WAP to print the Fibonacci series in a given range.



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

int fibo(int n)
{
    if (n == 0 || n == 1)
    {
        return n;
    }
    return fibo(n - 1) + fibo(n - 2);
}

int main()
{
    int n;
    cin >> n;
    for (int i = 0; i < n; i++)
    {
        cout << fibo(i) << endl;
```

```
    }

    return 0;
}
```



Input:

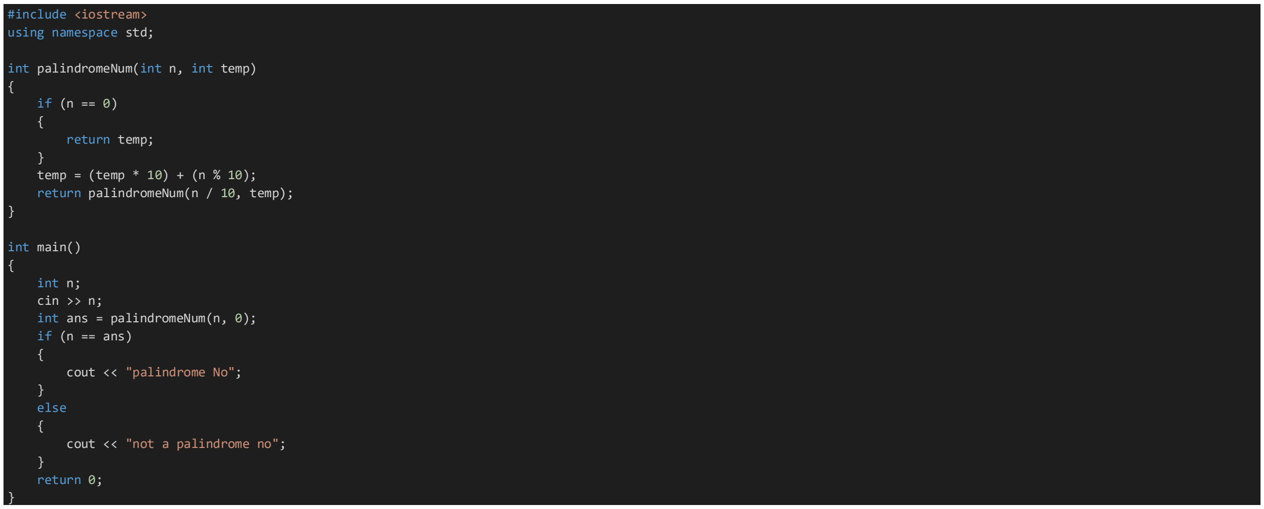
```
5
```

Expected Output:

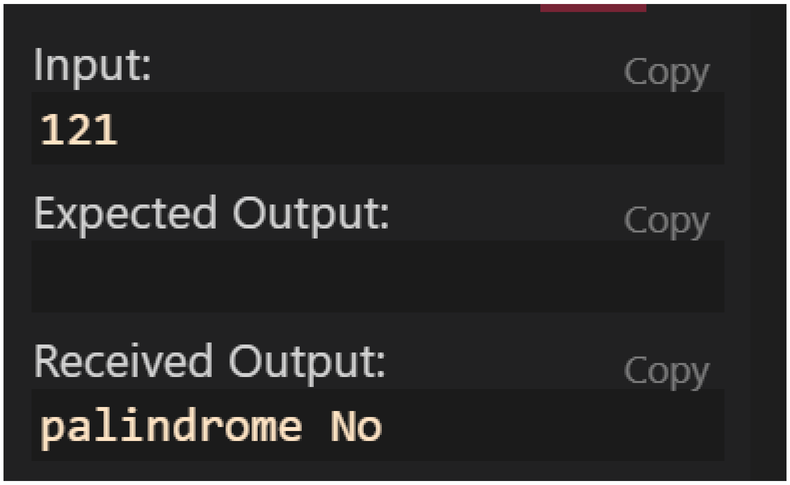
Received Output:

```
0
1
1
2
3
```

6. WAP to check whether the number entered is palindrome or not.



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

int palindromeNum(int n, int temp)
{
    if (n == 0)
    {
        return temp;
    }
    temp = (temp * 10) + (n % 10);
    return palindromeNum(n / 10, temp);
}

int main()
{
    int n;
    cin >> n;
    int ans = palindromeNum(n, 0);
    if (n == ans)
    {
        cout << "palindrome No";
    }
    else
    {
        cout << "not a palindrome no";
    }
    return 0;
}
```



Input:

```
121
```

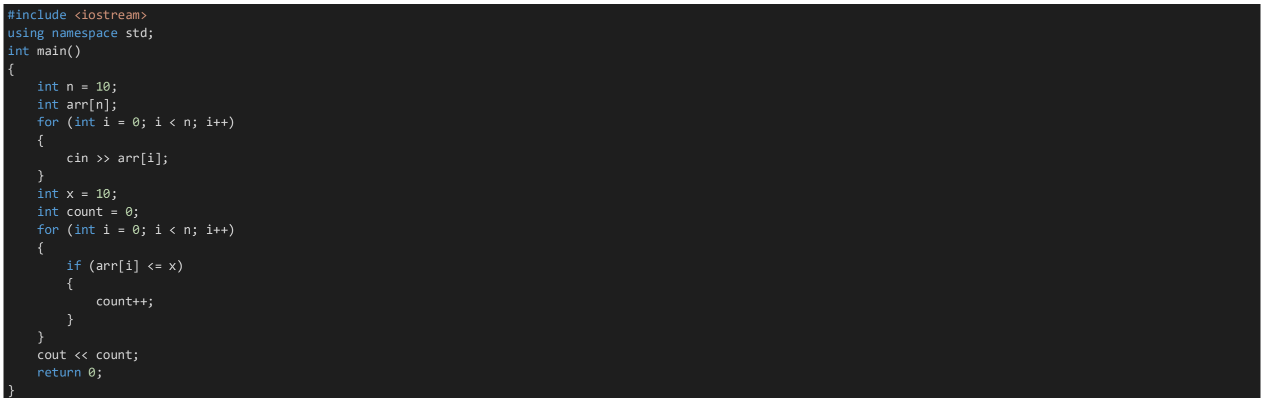
Expected Output:

```

```

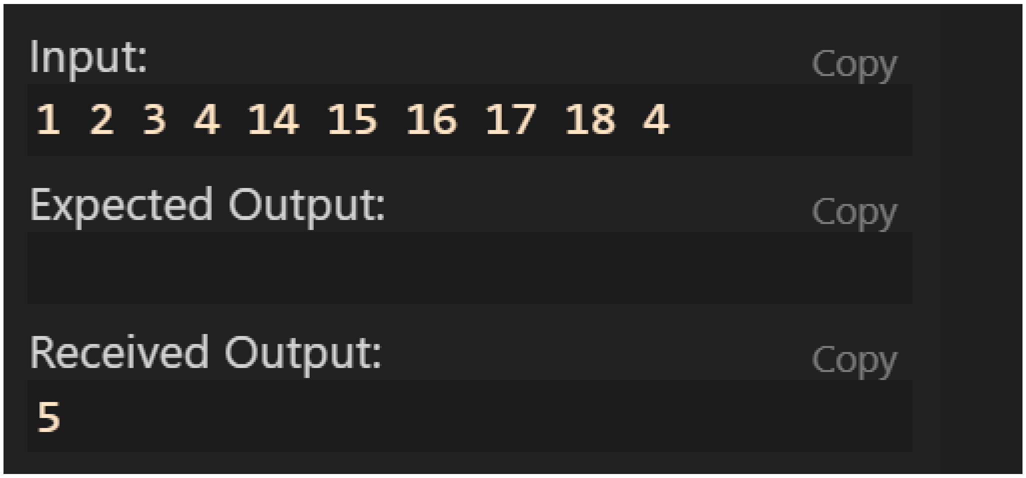
Received Output:

```
palindrome No
```

1. Write a program that asks the user to take array of 10 integers. The program must compute and write how many integers are greater than or equal to 10.

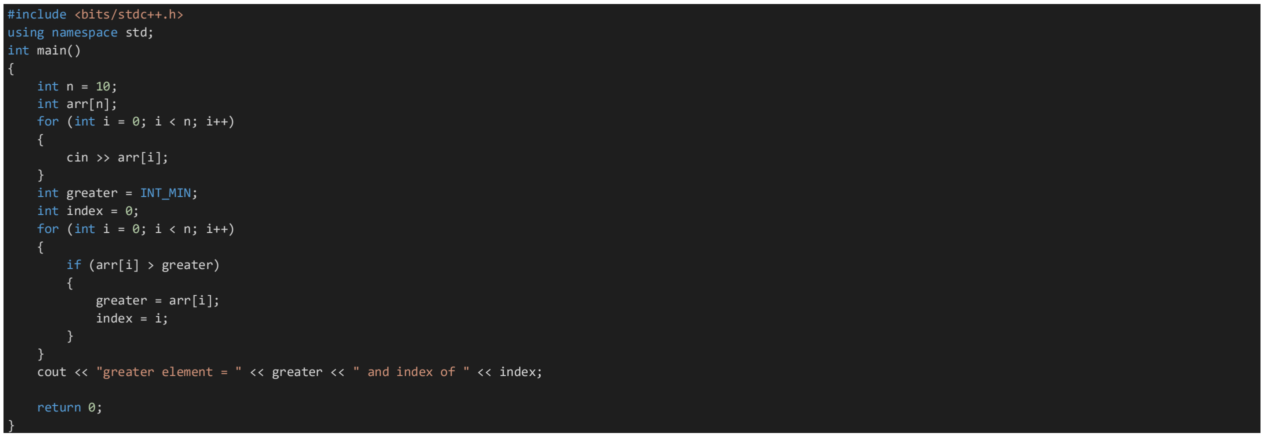


```
#include <iostream>
using namespace std;
int main()
{
    int n = 10;
    int arr[n];
    for (int i = 0; i < n; i++)
    {
        cin >> arr[i];
    }
    int x = 10;
    int count = 0;
    for (int i = 0; i < n; i++)
    {
        if (arr[i] <= x)
        {
            count++;
        }
    }
    cout << count;
    return 0;
}
```

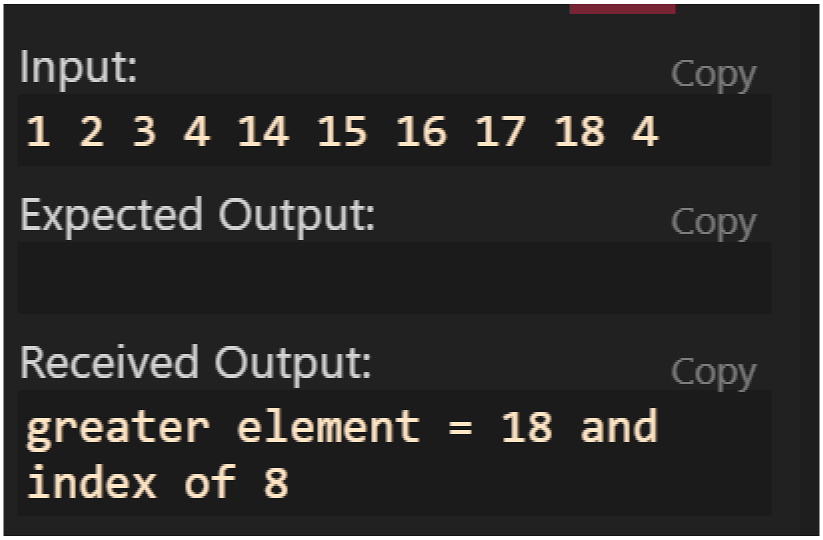


```
Input:
1 2 3 4 14 15 16 17 18 4
Expected Output:

Received Output:
5
```

2. Write a program that asks the user to take array of 10 integers. The program must output the largest element in the array, and the index at which that element was found.



```
#include <bits/stdc++.h>
using namespace std;
int main()
{
    int n = 10;
    int arr[n];
    for (int i = 0; i < n; i++)
    {
        cin >> arr[i];
    }
    int greater = INT_MIN;
    int index = 0;
    for (int i = 0; i < n; i++)
    {
        if (arr[i] > greater)
        {
            greater = arr[i];
            index = i;
        }
    }
    cout << "greater element = " << greater << " and index of " << index;

    return 0;
}
```



Input:

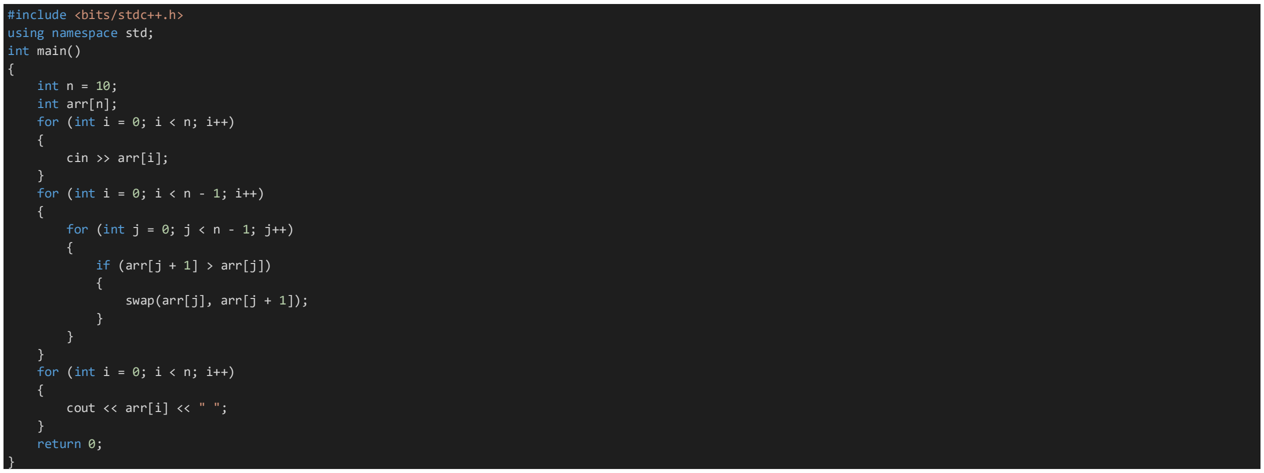
```
1 2 3 4 14 15 16 17 18 4
```

Expected Output:

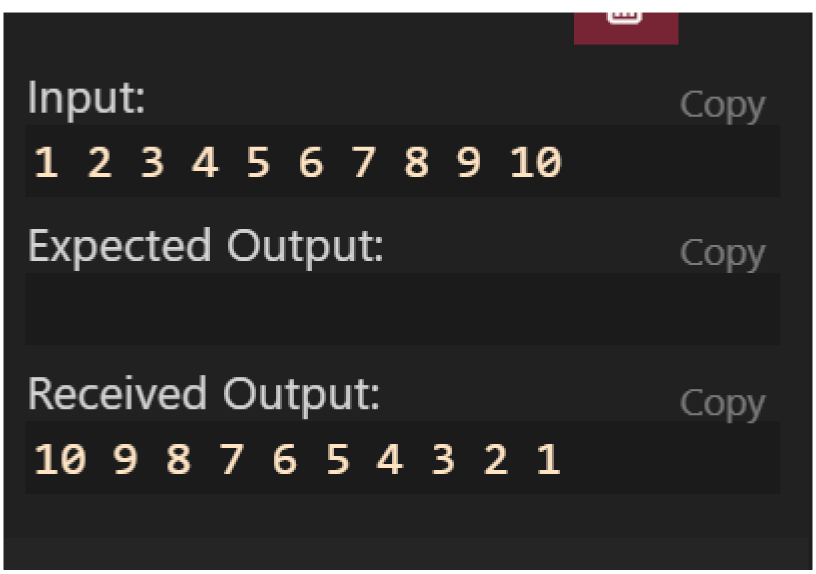
Received Output:

```
greater element = 18 and
index of 8
```

3. Write a program that asks the user to take array of 10 integers. The program will then sort the array in descending order and display it.



```cpp
#include <bits/stdc++.h>
using namespace std;
int main()
{
    int n = 10;
    int arr[n];
    for (int i = 0; i < n; i++)
    {
        cin >> arr[i];
    }
    for (int i = 0; i < n - 1; i++)
    {
        for (int j = 0; j < n - 1; j++)
        {
            if (arr[j + 1] > arr[j])
            {
                swap(arr[j], arr[j + 1]);
            }
        }
    }
    for (int i = 0; i < n; i++)
    {
        cout << arr[i] << " ";
    }
    return 0;
}
```



Input:

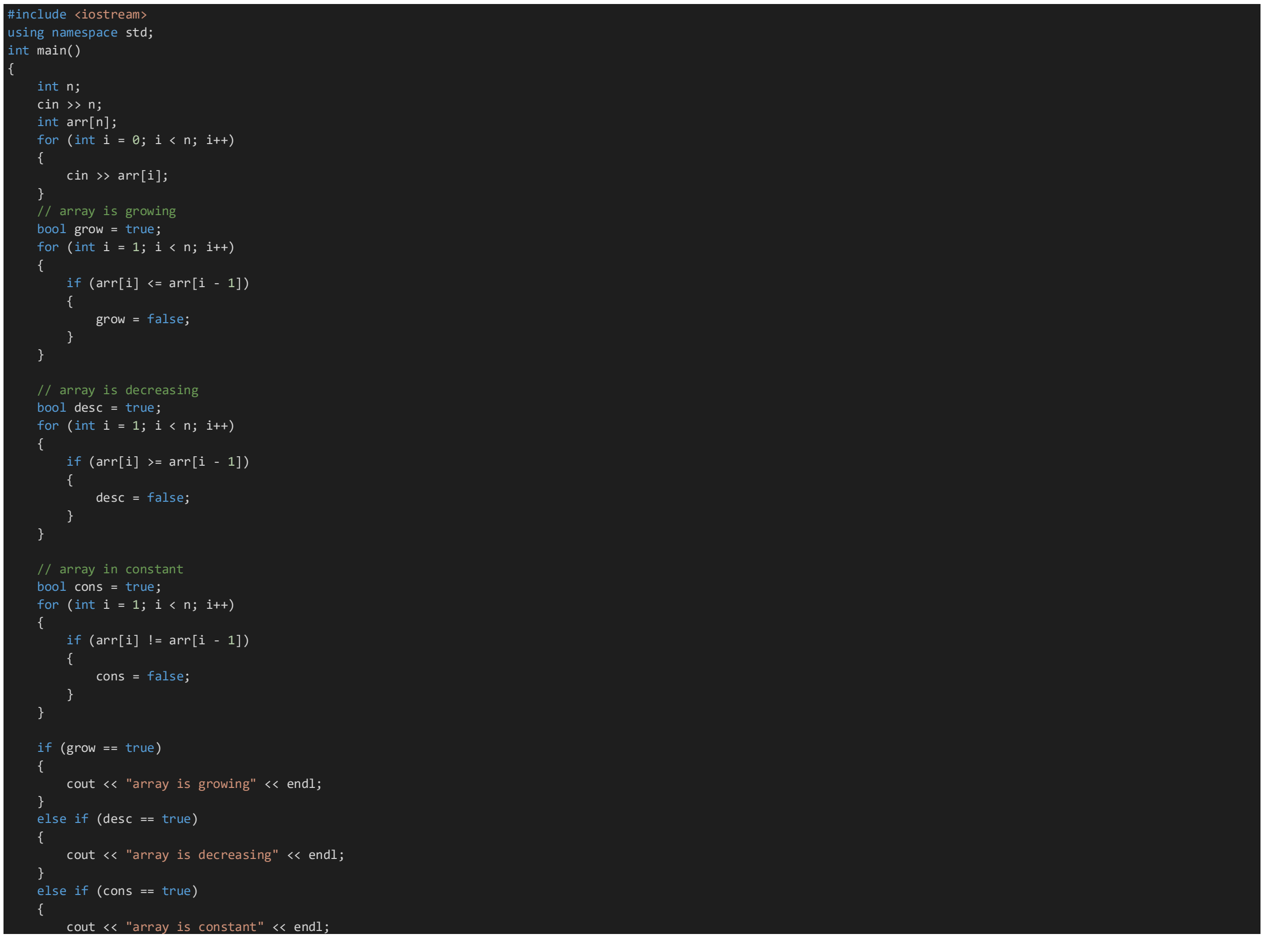
```
1 2 3 4 5 6 7 8 9 10
```

Expected Output:

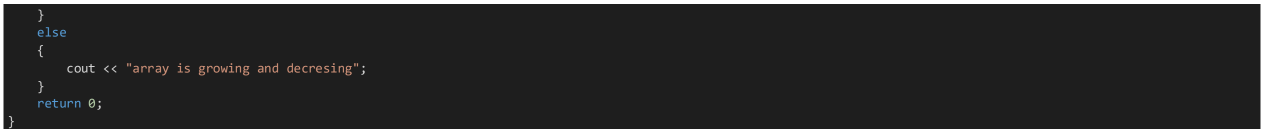
Received Output:

```
10 9 8 7 6 5 4 3 2 1
```

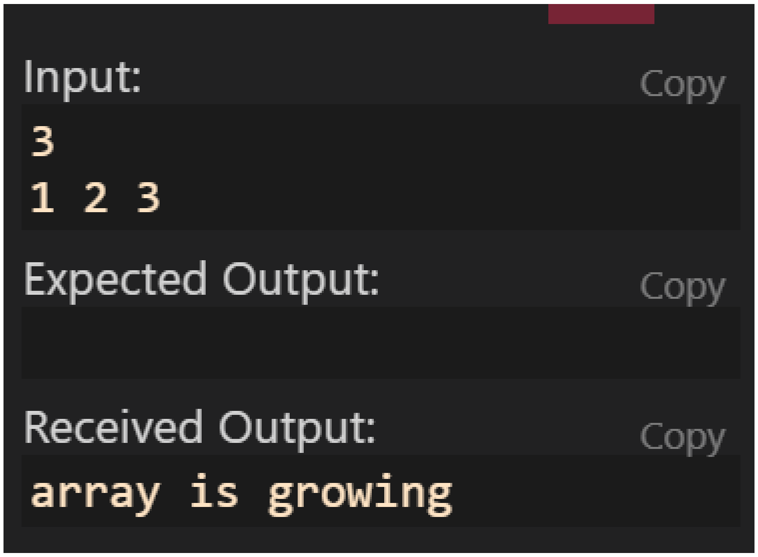
4. Write a program that asks the user to take array of 10 integers. The program will then display either “the array is growing”, “the array is decreasing”, “the array is constant”, or “the array is growing and decreasing.”;



```
#include <iostream>
using namespace std;
int main()
{
    int n;
    cin >> n;
    int arr[n];
    for (int i = 0; i < n; i++)
    {
        cin >> arr[i];
    }
    // array is growing
    bool grow = true;
    for (int i = 1; i < n; i++)
    {
        if (arr[i] <= arr[i - 1])
        {
            grow = false;
        }
    }

    // array is decreasing
    bool desc = true;
    for (int i = 1; i < n; i++)
    {
        if (arr[i] >= arr[i - 1])
        {
            desc = false;
        }
    }

    // array in constant
    bool cons = true;
    for (int i = 1; i < n; i++)
    {
        if (arr[i] != arr[i - 1])
        {
            cons = false;
        }
    }

    if (grow == true)
    {
        cout << "array is growing" << endl;
    }
    else if (desc == true)
    {
        cout << "array is decreasing" << endl;
    }
    else if (cons == true)
    {
        cout << "array is constant" << endl;
```



```
    }
    else
    {
        cout << "array is growing and decresing";
    }
    return 0;
}
```



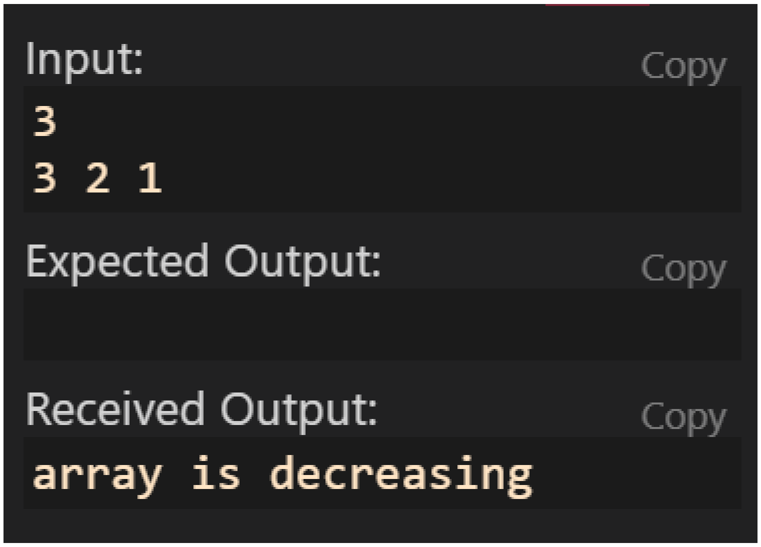
Input:

```
3
1 2 3
```

Expected Output:

Received Output:

```
array is growing
```



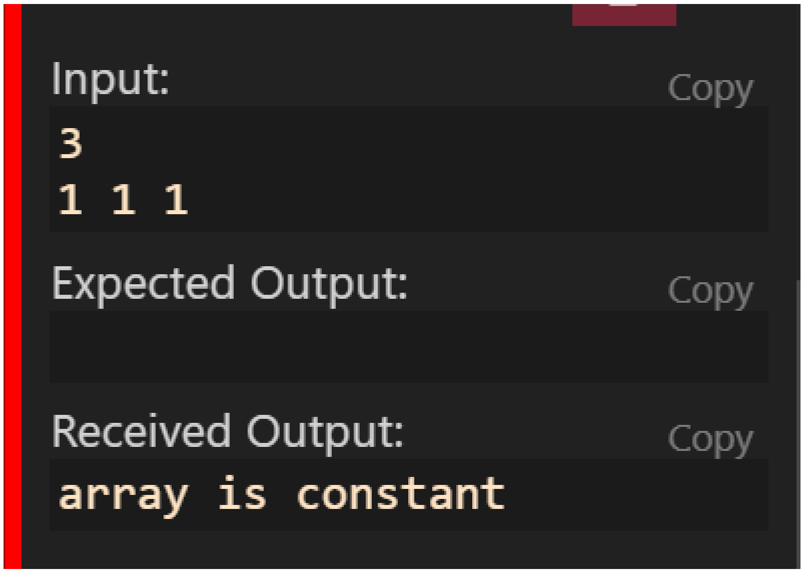
Input:

```
3
3 2 1
```

Expected Output:

Received Output:

```
array is decreasing
```

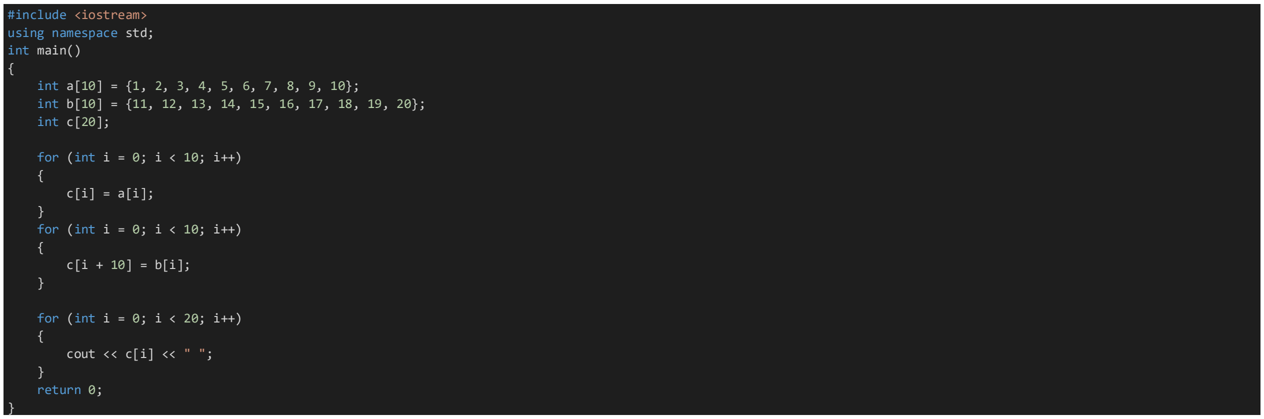


```
3
1 1 1
```

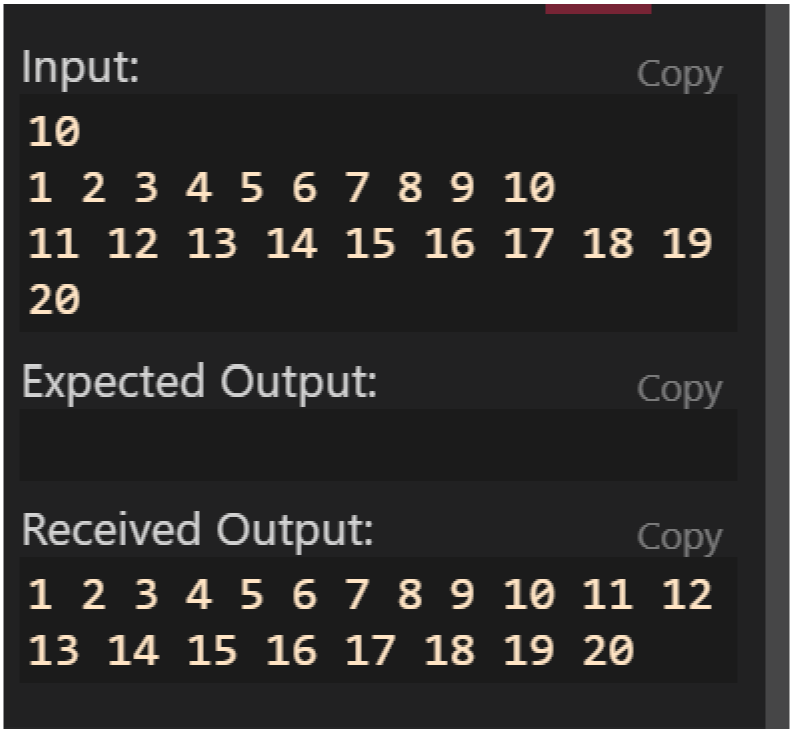
```
array is constant
```

5. Write a program which takes 2 arrays of 10 integers each, a and b. c is an array with 20 integers.

The program should put into c the appending of b to a, the first 10 integers of c from array a, the latter 10 from b. Then the program should display c.



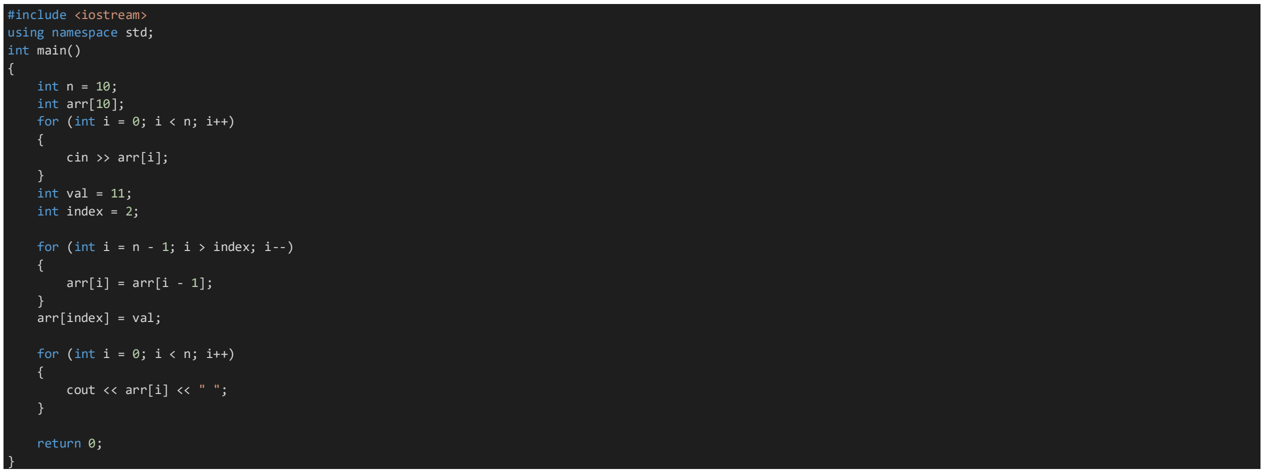
```cpp
#include <iostream>
using namespace std;
int main()
{
    int a[10] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};
    int b[10] = {11, 12, 13, 14, 15, 16, 17, 18, 19, 20};
    int c[20];

    for (int i = 0; i < 10; i++)
    {
        c[i] = a[i];
    }
    for (int i = 0; i < 10; i++)
    {
        c[i + 10] = b[i];
    }

    for (int i = 0; i < 20; i++)
    {
        cout << c[i] << " ";
    }
    return 0;
}
```



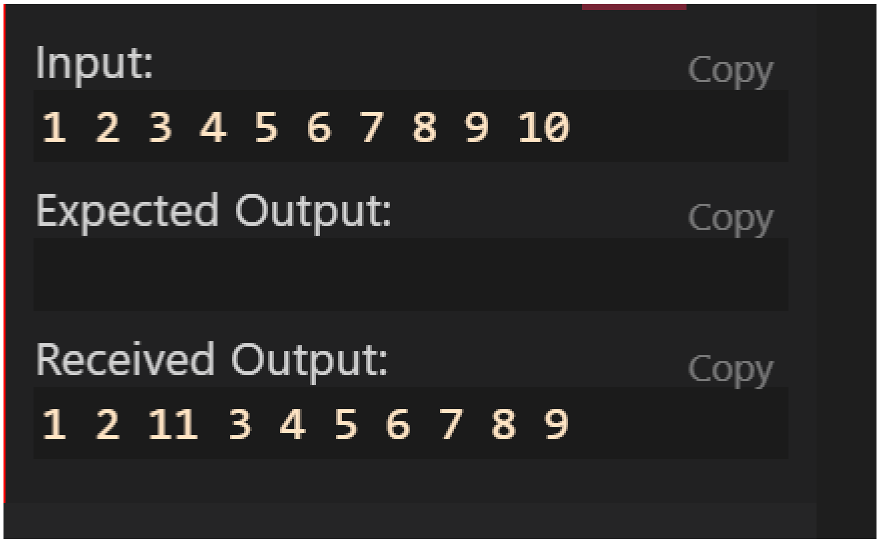
```
10
1 2 3 4 5 6 7 8 9 10
11 12 13 14 15 16 17 18 19
20
```

```
1 2 3 4 5 6 7 8 9 10 11 12
13 14 15 16 17 18 19 20
```

6. Write a program that asks the user to take an array of 10 integer and an integer value V and an index value i between 0 and 9. The program must put the value V at the place i in the array, shifting each element right and dropping off the last element. The program must then write the final array.



```cpp
#include <iostream>
using namespace std;
int main()
{
    int n = 10;
    int arr[10];
    for (int i = 0; i < n; i++)
    {
        cin >> arr[i];
    }
    int val = 11;
    int index = 2;

    for (int i = n - 1; i > index; i--)
    {
        arr[i] = arr[i - 1];
    }
    arr[index] = val;

    for (int i = 0; i < n; i++)
    {
        cout << arr[i] << " ";
    }

    return 0;
}
```



Input:

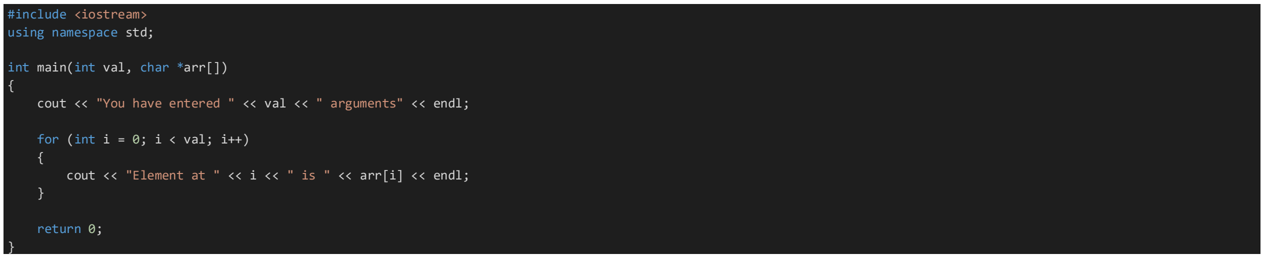
```
1 2 3 4 5 6 7 8 9 10
```

Expected Output:

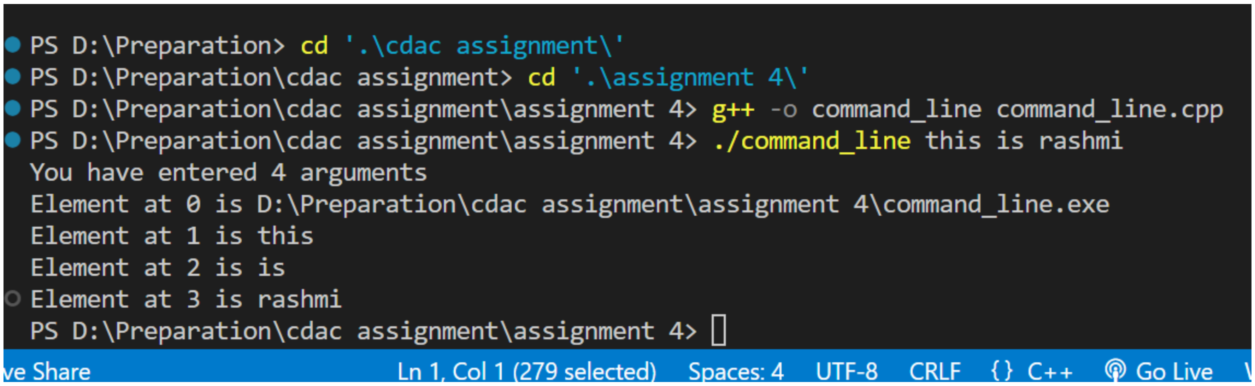
Received Output:

```
1 2 11 3 4 5 6 7 8 9
```

7. Write a program to handle the command line arguments entered by the user.

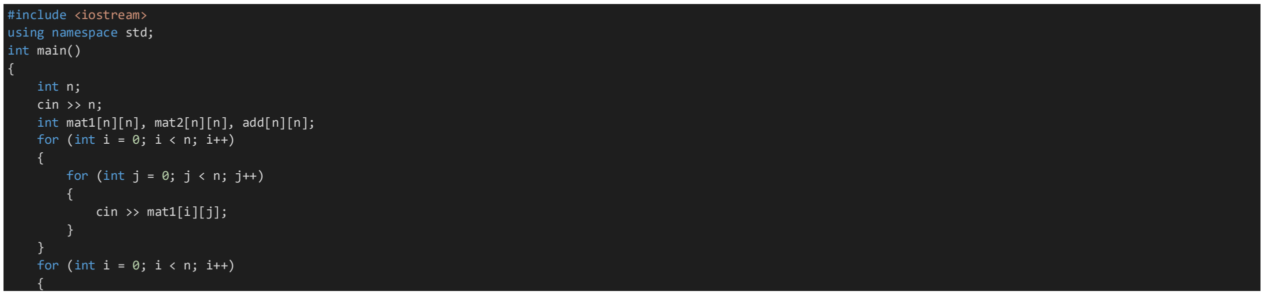


```cpp
#include <iostream>
using namespace std;

int main(int val, char *arr[])
{
    cout << "You have entered " << val << " arguments" << endl;

    for (int i = 0; i < val; i++)
    {
        cout << "Element at " << i << " is " << arr[i] << endl;
    }

    return 0;
}
```

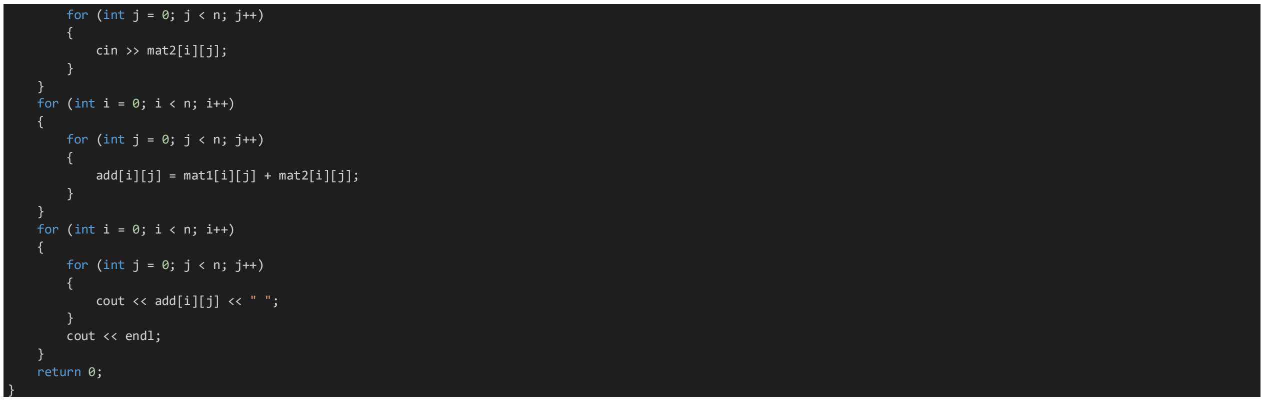


```
PS D:\Preparation> cd '.\cdac assignment\'
PS D:\Preparation\cdac assignment> cd '.\assignment 4\'
PS D:\Preparation\cdac assignment\assignment 4> g++ -o command_line command_line.cpp
PS D:\Preparation\cdac assignment\assignment 4> ./command_line this is rashmi
You have entered 4 arguments
Element at 0 is D:\Preparation\cdac assignment\assignment 4\command_line.exe
Element at 1 is this
Element at 2 is is
Element at 3 is rashmi
PS D:\Preparation\cdac assignment\assignment 4> 
```

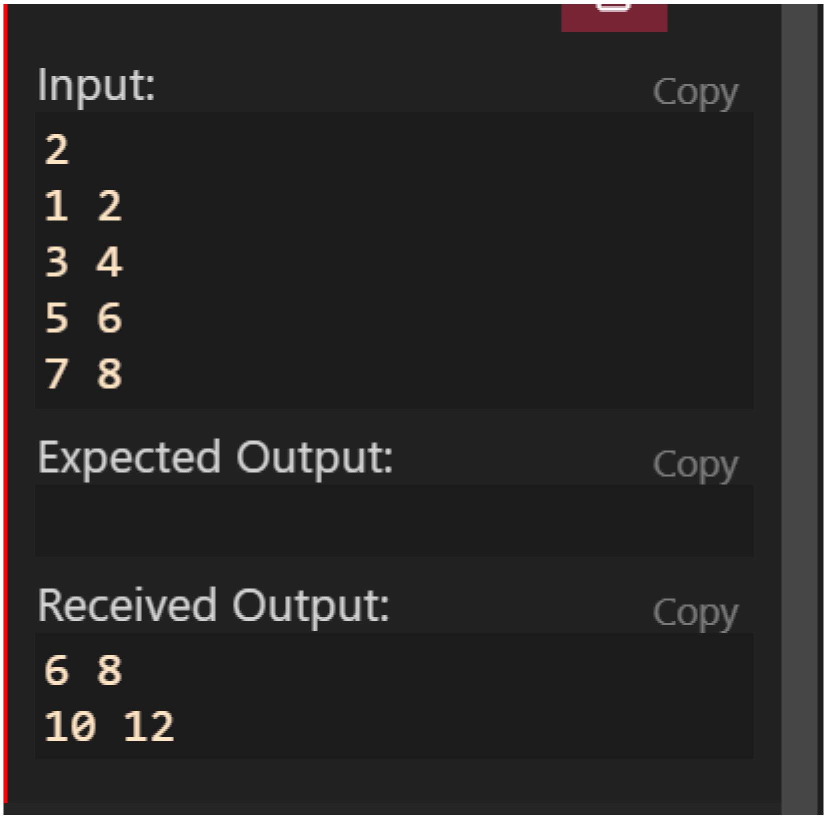
8. Write a program to add 2 matrices.



```cpp
#include <iostream>
using namespace std;
int main()
{
    int n;
    cin >> n;
    int mat1[n][n], mat2[n][n], add[n][n];
    for (int i = 0; i < n; i++)
    {
        for (int j = 0; j < n; j++)
        {
            cin >> mat1[i][j];
        }
    }
    for (int i = 0; i < n; i++)
    {
```



```
        for (int j = 0; j < n; j++)
        {
            cin >> mat2[i][j];
        }
    }
    for (int i = 0; i < n; i++)
    {
        for (int j = 0; j < n; j++)
        {
            add[i][j] = mat1[i][j] + mat2[i][j];
        }
    }
    for (int i = 0; i < n; i++)
    {
        for (int j = 0; j < n; j++)
        {
            cout << add[i][j] << " ";
        }
        cout << endl;
    }
    return 0;
}
```



Input:

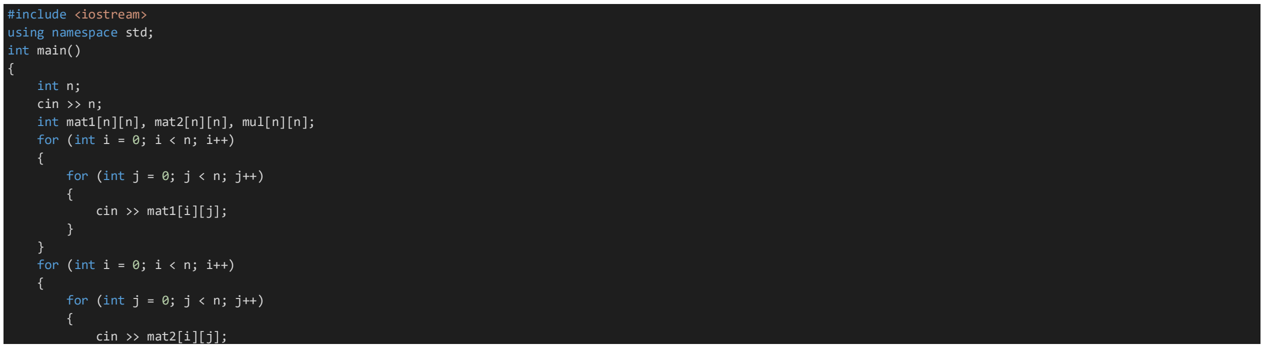
```
2
1 2
3 4
5 6
7 8
```

Expected Output:

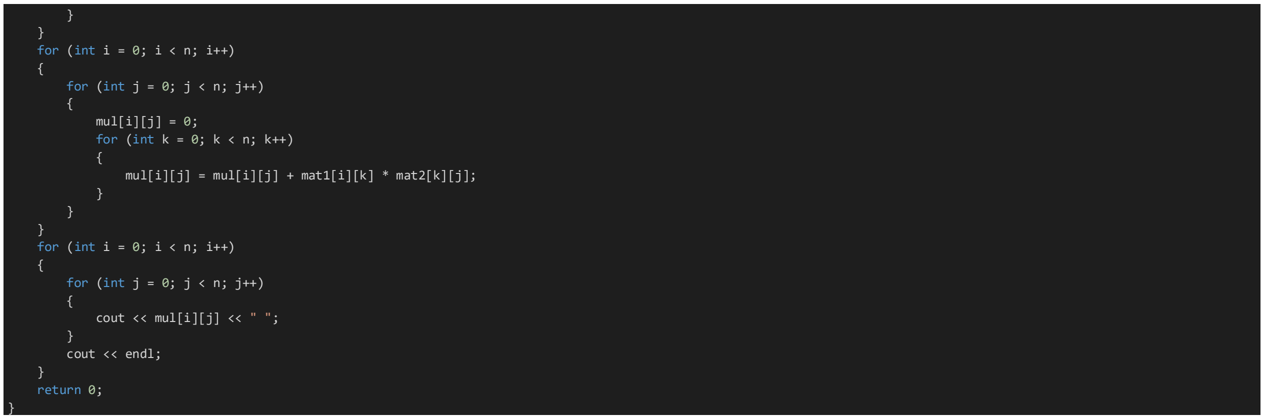
Received Output:

```
6 8
10 12
```

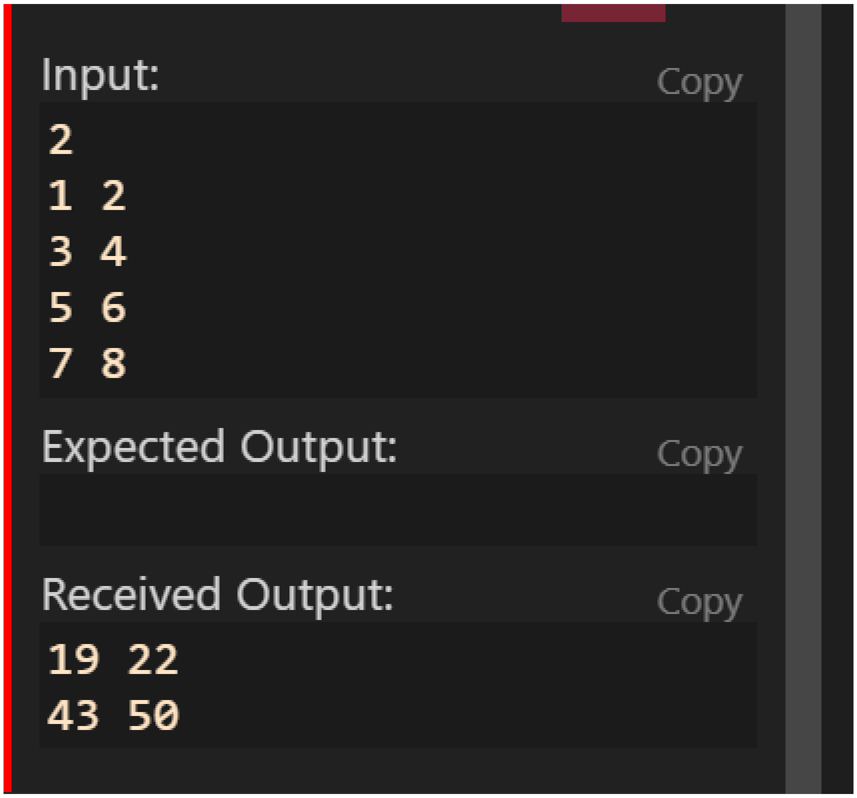
9. Write a program to multiply 2 matrices.



```
#include <iostream>
using namespace std;
int main()
{
    int n;
    cin >> n;
    int mat1[n][n], mat2[n][n], mul[n][n];
    for (int i = 0; i < n; i++)
    {
        for (int j = 0; j < n; j++)
        {
            cin >> mat1[i][j];
        }
    }
    for (int i = 0; i < n; i++)
    {
        for (int j = 0; j < n; j++)
        {
            cin >> mat2[i][j];
```



```
        }
    }
    for (int i = 0; i < n; i++)
    {
        for (int j = 0; j < n; j++)
        {
            mul[i][j] = 0;
            for (int k = 0; k < n; k++)
            {
                mul[i][j] = mul[i][j] + mat1[i][k] * mat2[k][j];
            }
        }
    }
    for (int i = 0; i < n; i++)
    {
        for (int j = 0; j < n; j++)
        {
            cout << mul[i][j] << " ";
        }
        cout << endl;
    }
    return 0;
}
```



Input:

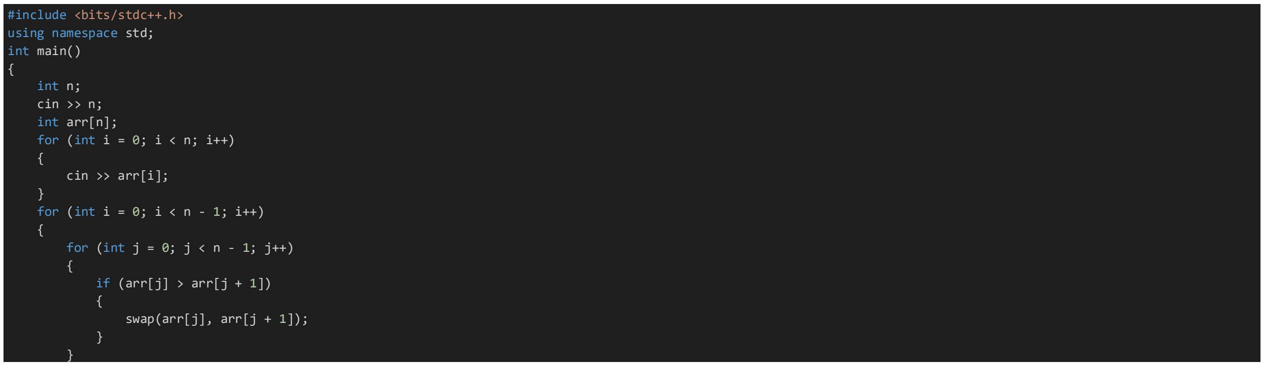
```
2
1 2
3 4
5 6
7 8
```

Expected Output:

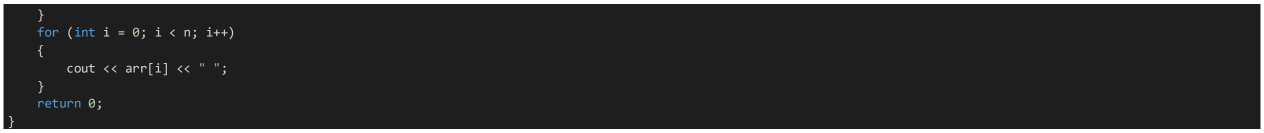
Received Output:

```
19 22
43 50
```

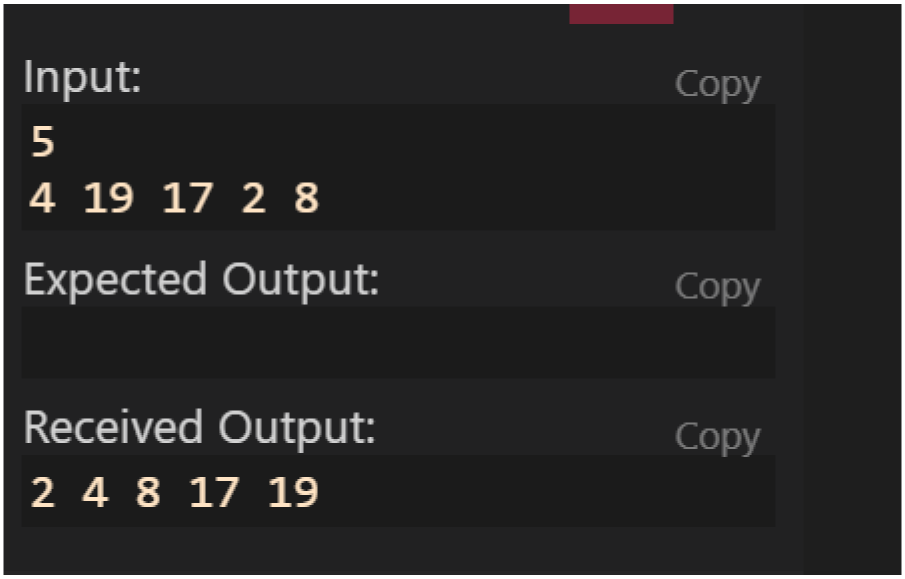
10. Write a program to implement sorting an array.



```
#include <bits/stdc++.h>
using namespace std;
int main()
{
    int n;
    cin >> n;
    int arr[n];
    for (int i = 0; i < n; i++)
    {
        cin >> arr[i];
    }
    for (int i = 0; i < n - 1; i++)
    {
        for (int j = 0; j < n - 1; j++)
        {
            if (arr[j] > arr[j + 1])
            {
                swap(arr[j], arr[j + 1]);
            }
        }
```



```
    }
    for (int i = 0; i < n; i++)
    {
        cout << arr[i] << " ";
    }
    return 0;
}
```



Input:

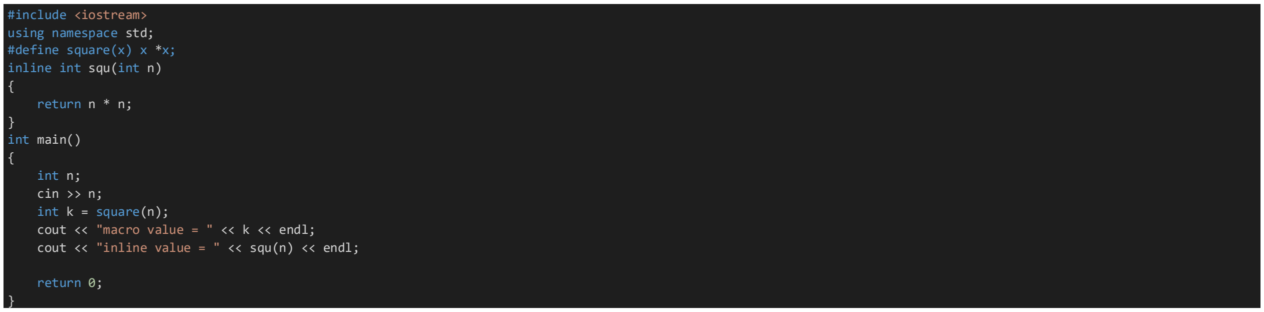
```
5
4 19 17 2 8
```

Expected Output:

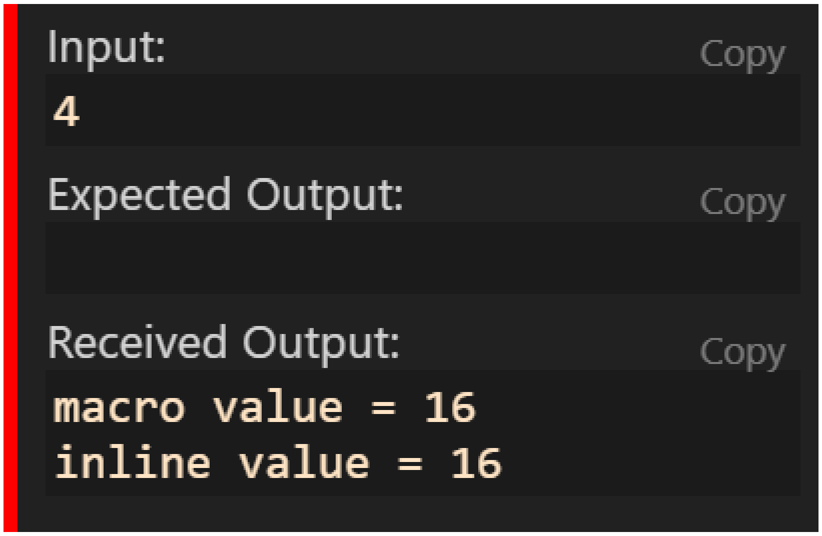
Received Output:

```
2 4 8 17 19
```

11. Write a program in C to calculate the square of the number using inline functions and macros both.



```
#include <iostream>
using namespace std;
#define square(x) x *x;
inline int squ(int n)
{
    return n * n;
}
int main()
{
    int n;
    cin >> n;
    int k = square(n);
    cout << "macro value = " << k << endl;
    cout << "inline value = " << squ(n) << endl;

    return 0;
}
```



Input:

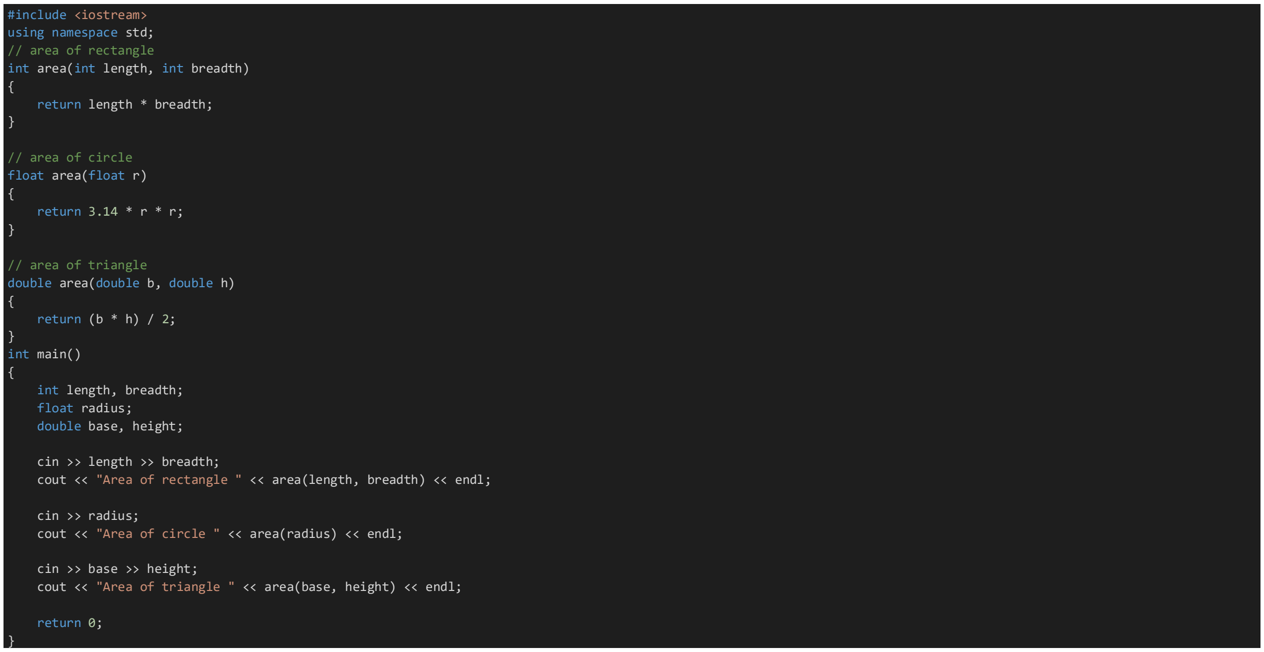
```
4
```

Expected Output:

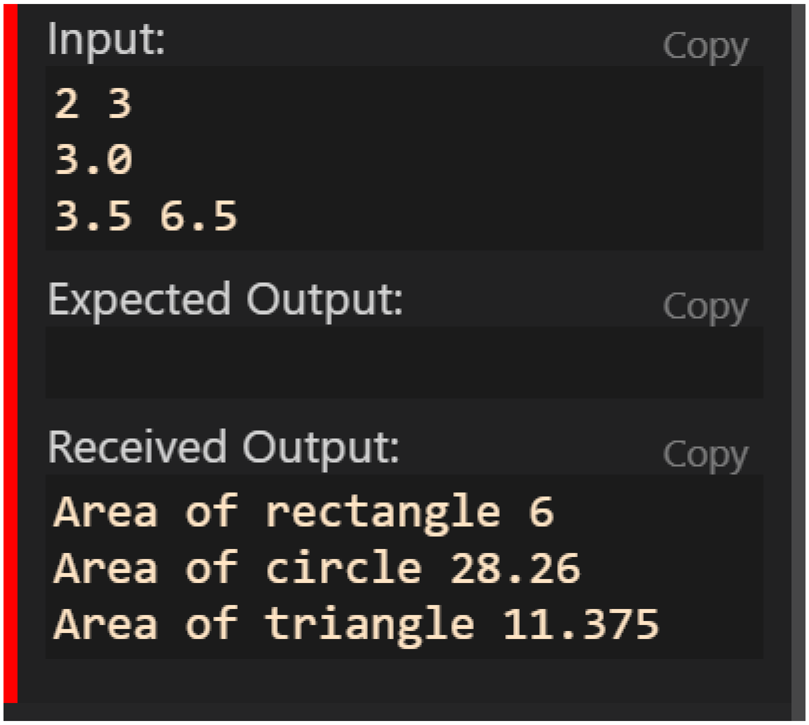
Received Output:

```
macro value = 16
inline value = 16
```

12. Write a program in C to calculate area of all figures using the concept of function overloading.



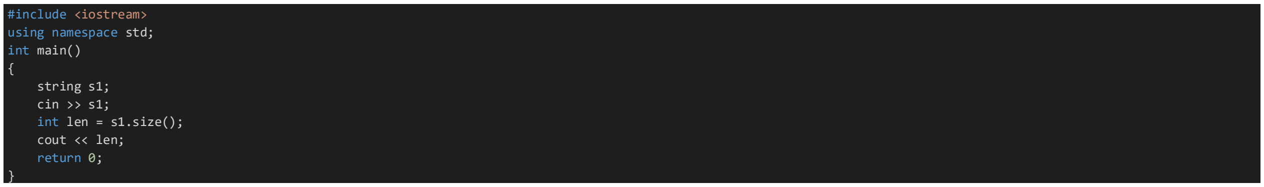
```
#include <iostream>
using namespace std;
// area of rectangle
int area(int length, int breadth)
{
    return length * breadth;
}

// area of circle
float area(float r)
{
    return 3.14 * r * r;
}

// area of triangle
double area(double b, double h)
{
    return (b * h) / 2;
}
int main()
{
    int length, breadth;
    float radius;
    double base, height;

    cin >> length >> breadth;
    cout << "Area of rectangle " << area(length, breadth) << endl;

    cin >> radius;
    cout << "Area of circle " << area(radius) << endl;

    cin >> base >> height;
    cout << "Area of triangle " << area(base, height) << endl;

    return 0;
}
```



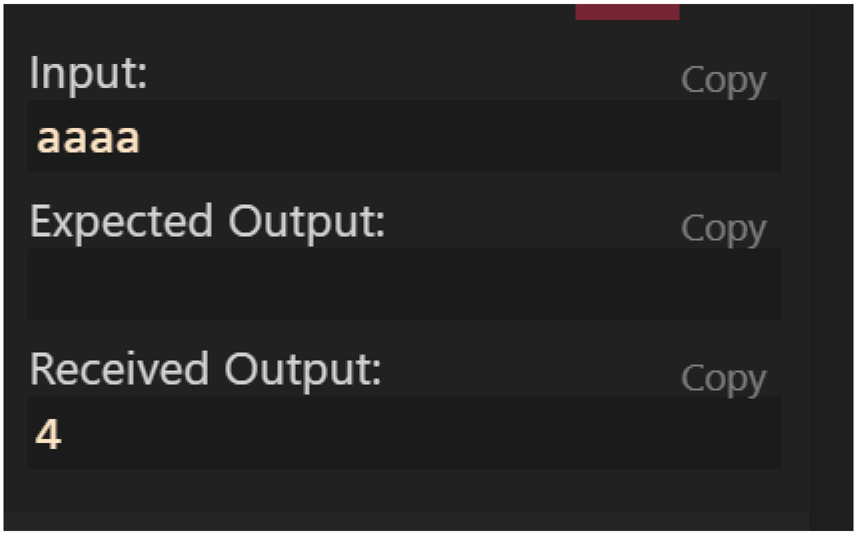
```
2 3
3.0
3.5 6.5
```

```
Area of rectangle 6
Area of circle 28.26
Area of triangle 11.375
```

1. Write a program to find the length of string.



```
#include <iostream>
using namespace std;
int main()
{
    string s1;
    cin >> s1;
    int len = s1.size();
    cout << len;
    return 0;
}
```



Input:

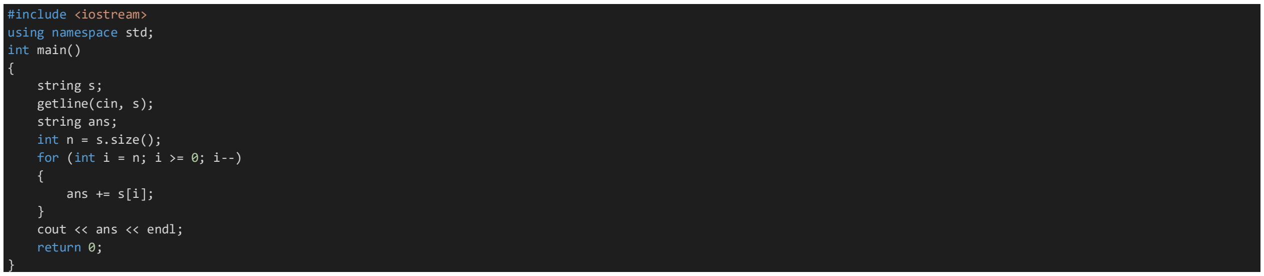
```
aaaa
```

Expected Output:

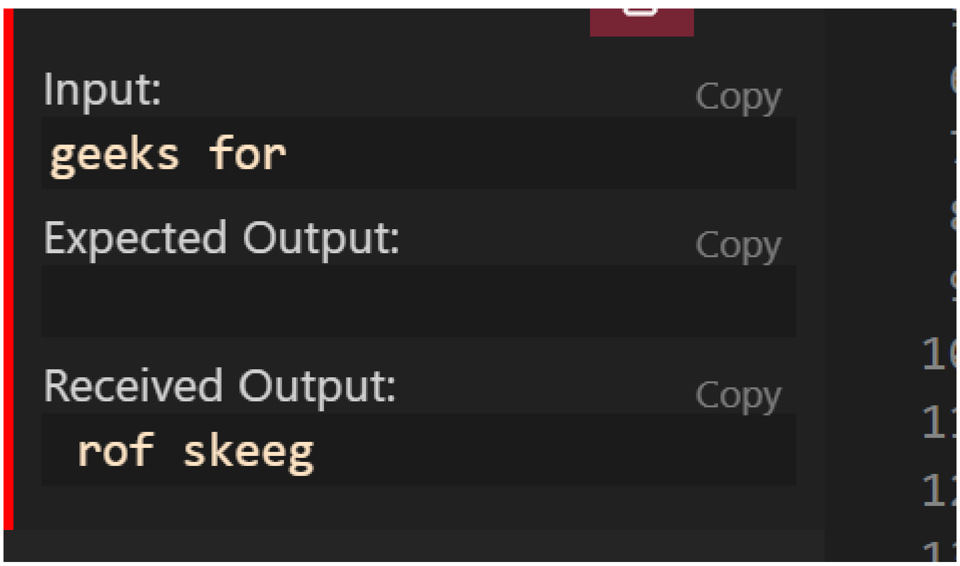
Received Output:

```
4
```

2. Write a program to display string from backward.



```
#include <iostream>
using namespace std;
int main()
{
    string s;
    getline(cin, s);
    string ans;
    int n = s.size();
    for (int i = n; i >= 0; i--)
    {
        ans += s[i];
    }
    cout << ans << endl;
    return 0;
}
```



Input:

```
geeks for
```

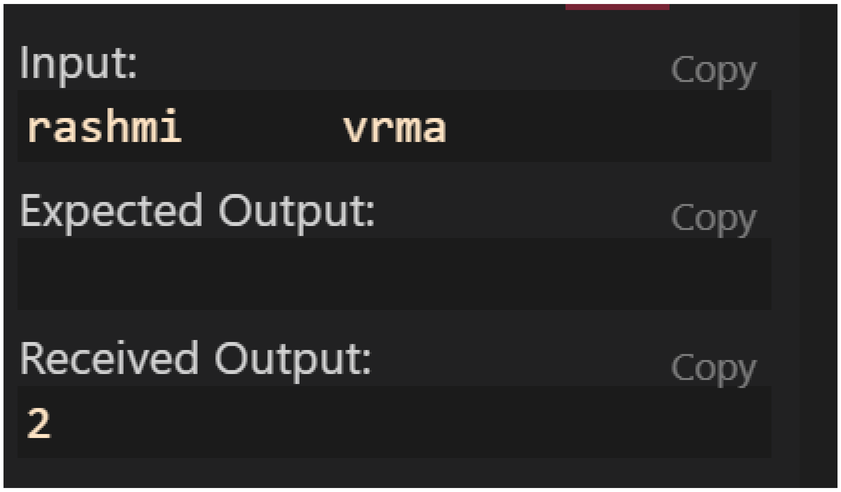
Expected Output:

Received Output:

```
 rof skeeg
```

3. Write a program to count number of words in string.

```cpp
#include <iostream>
using namespace std;
int main()
{
    string s;
    getline(cin, s);

    bool inWords = false;
    int word = 0;
    int n = s.size();
    for (int i = 0; i < n; i++)
    {
        if (s[i] != ' ' && inWords == false)
        {
            word++;
            inWords = true;
        }
        else if (s[i] == ' ')
        {
            inWords = false;
        }
    }
    cout << word;

    return 0;
}
```



Input:

```
rashmi        vrma
```

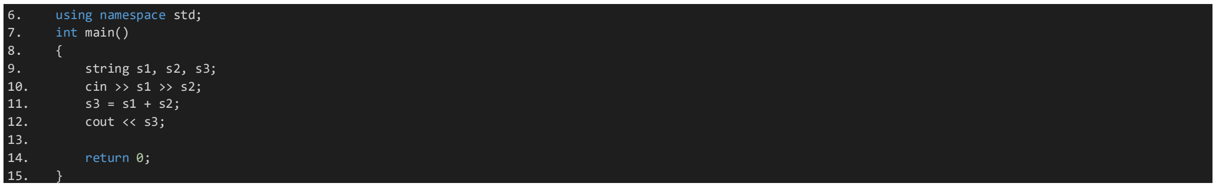
Expected Output:

Received Output:

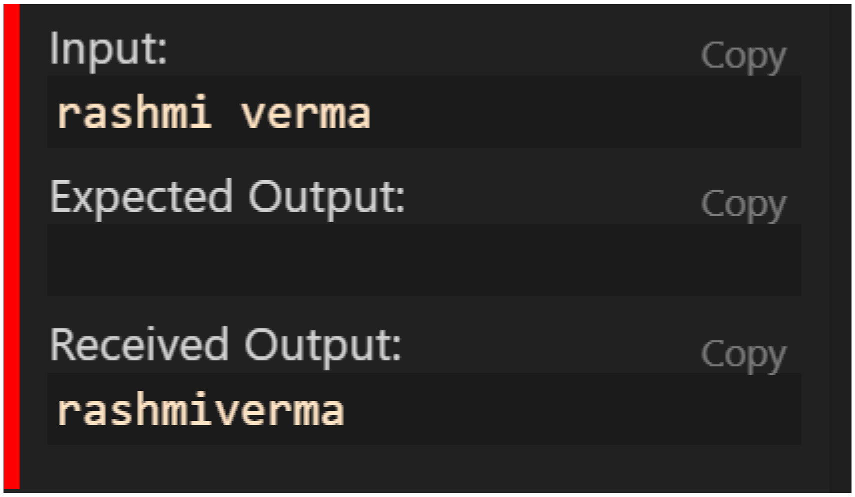
```
2
```

4. Write a program to concatenate one string contents to another.

```cpp
5.    #include <iostream>
```



```
using namespace std;
int main()
{
    string s1, s2, s3;
    cin >> s1 >> s2;
    s3 = s1 + s2;
    cout << s3;

    return 0;
}
```



Input:

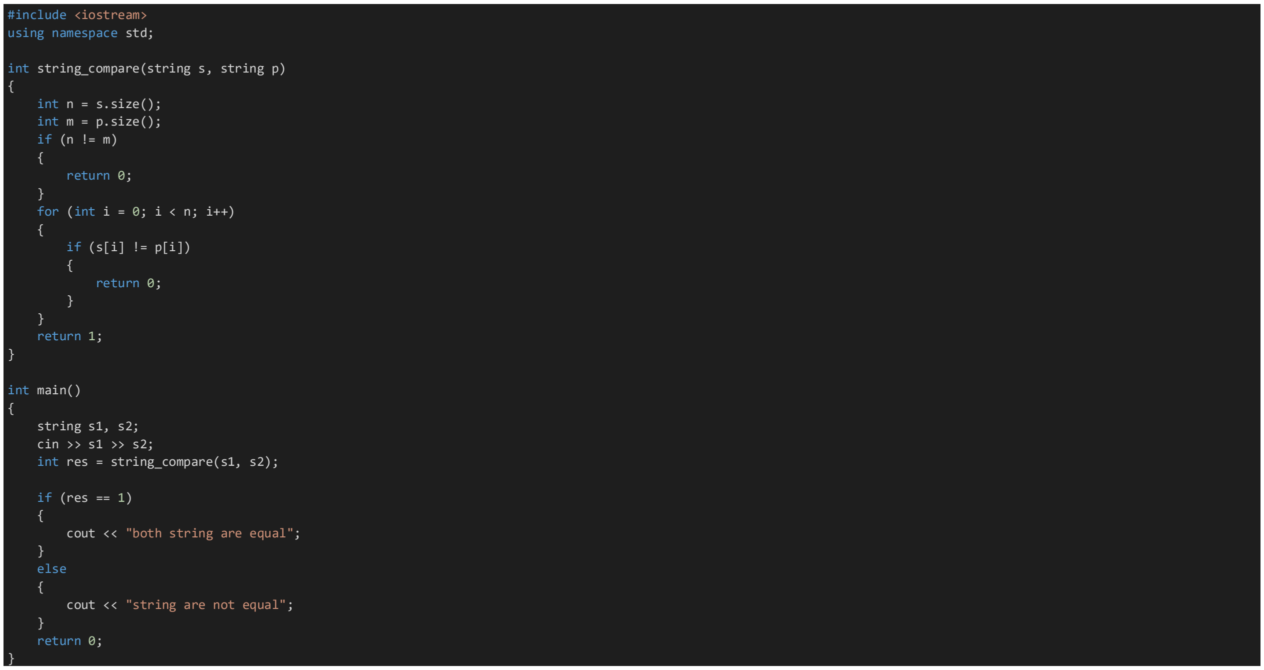
```
rashmi verma
```

Expected Output:

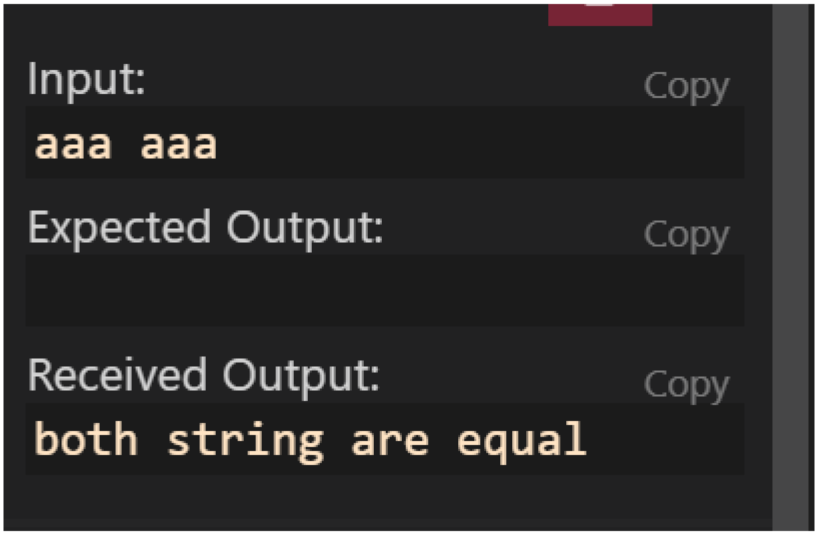
Received Output:

```
rashmiverma
```

5. Write a program to compare two strings they are exact equal or not.

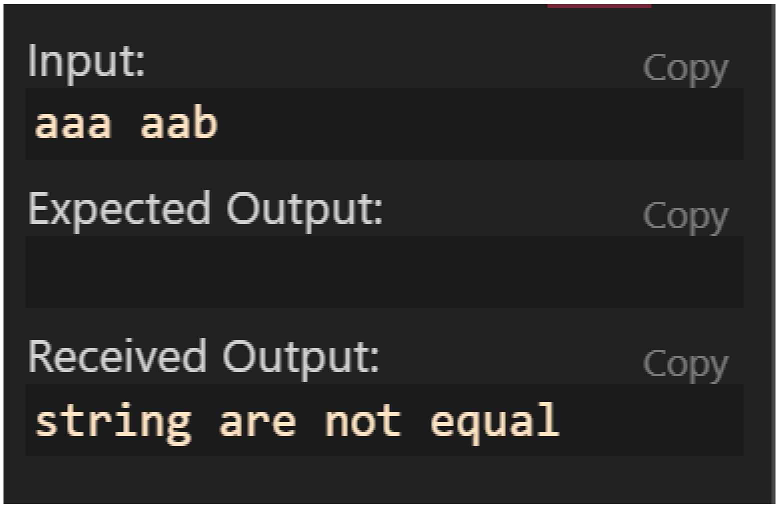


```
#include <iostream>
using namespace std;

int string_compare(string s, string p)
{
    int n = s.size();
    int m = p.size();
    if (n != m)
    {
        return 0;
    }
    for (int i = 0; i < n; i++)
    {
        if (s[i] != p[i])
        {
            return 0;
        }
    }
    return 1;
}

int main()
{
    string s1, s2;
    cin >> s1 >> s2;
    int res = string_compare(s1, s2);

    if (res == 1)
    {
        cout << "both string are equal";
    }
    else
    {
        cout << "string are not equal";
    }
    return 0;
}
```



```
aaa aaa
```

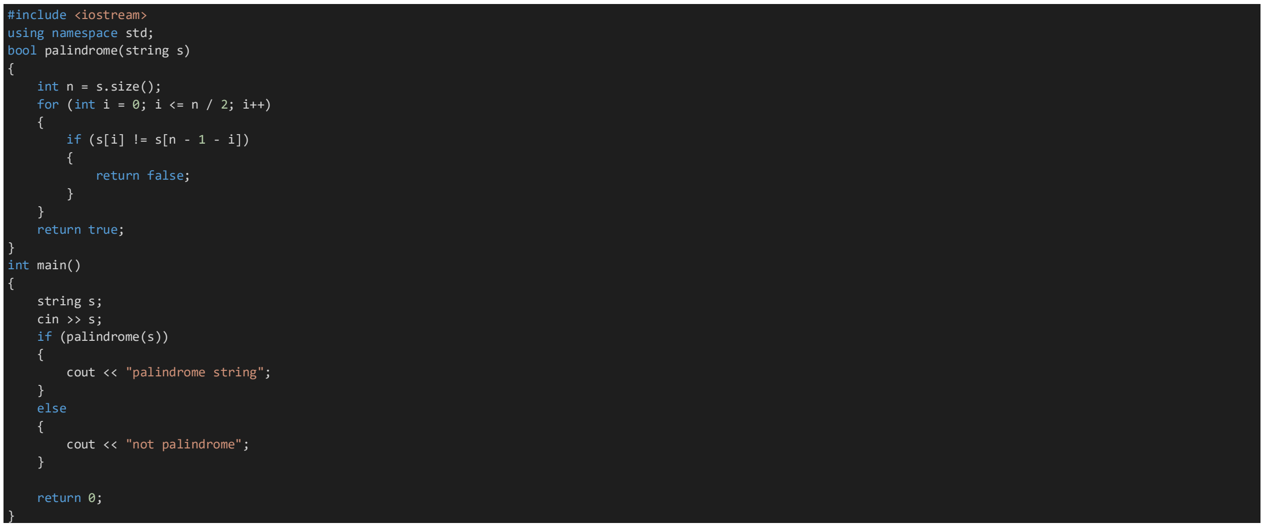
```
both string are equal
```



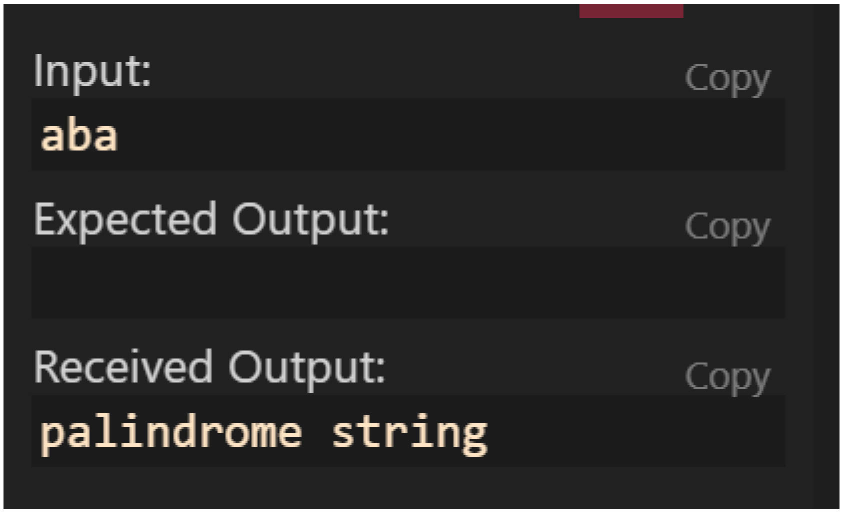
```
aaa aab
```

```
string are not equal
```

6. Write a program to check a string is palindrome or not.



```
#include <iostream>
using namespace std;
bool palindrome(string s)
{
    int n = s.size();
    for (int i = 0; i <= n / 2; i++)
    {
        if (s[i] != s[n - 1 - i])
        {
            return false;
        }
    }
    return true;
}
int main()
{
    string s;
    cin >> s;
    if (palindrome(s))
    {
        cout << "palindrome string";
    }
    else
    {
        cout << "not palindrome";
    }

    return 0;
}
```



Input:

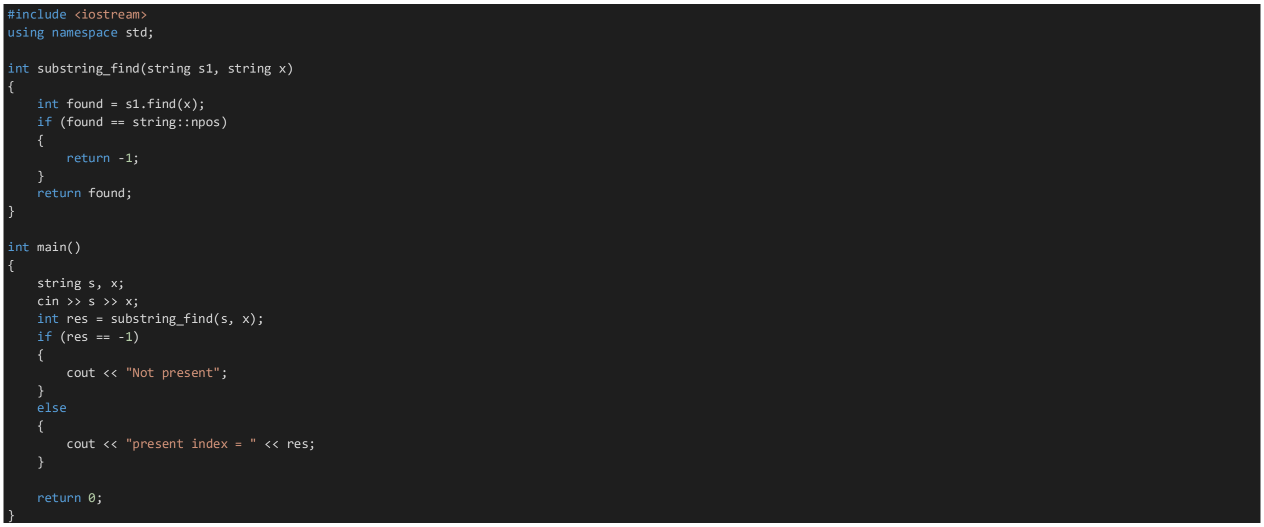
```
aba
```

Expected Output:

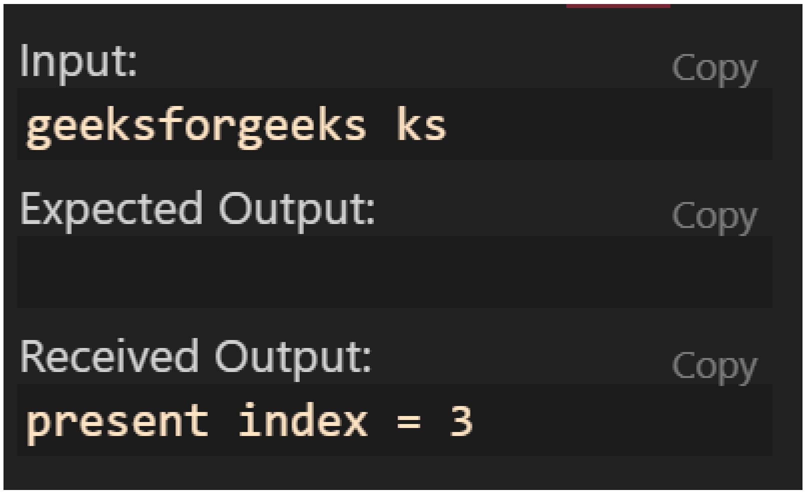
Received Output:

```
palindrome string
```

7. Write a program to find a substring within a string. If found display its starting position.



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

int substring_find(string s1, string x)
{
    int found = s1.find(x);
    if (found == string::npos)
    {
        return -1;
    }
    return found;
}

int main()
{
    string s, x;
    cin >> s >> x;
    int res = substring_find(s, x);
    if (res == -1)
    {
        cout << "Not present";
    }
    else
    {
        cout << "present index = " << res;
    }

    return 0;
}
```



Input:

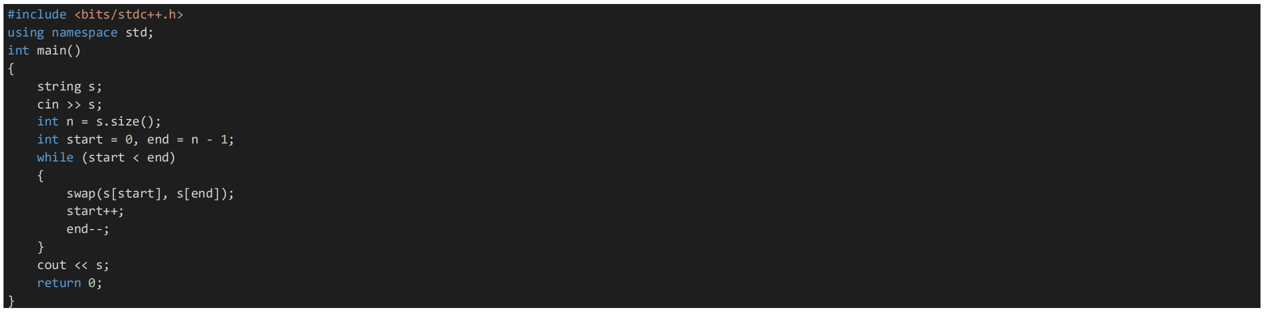
```
geeksforgeeks ks
```

Expected Output:

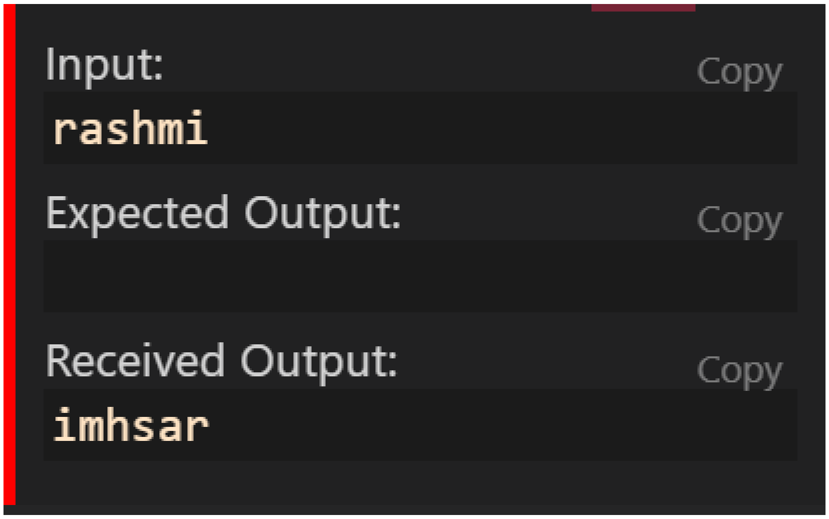
Received Output:

```
present index = 3
```

8. Write a program to reverse a string.



```
#include <bits/stdc++.h>
using namespace std;
int main()
{
    string s;
    cin >> s;
    int n = s.size();
    int start = 0, end = n - 1;
    while (start < end)
    {
        swap(s[start], s[end]);
        start++;
        end--;
    }
    cout << s;
    return 0;
}
```



Input:

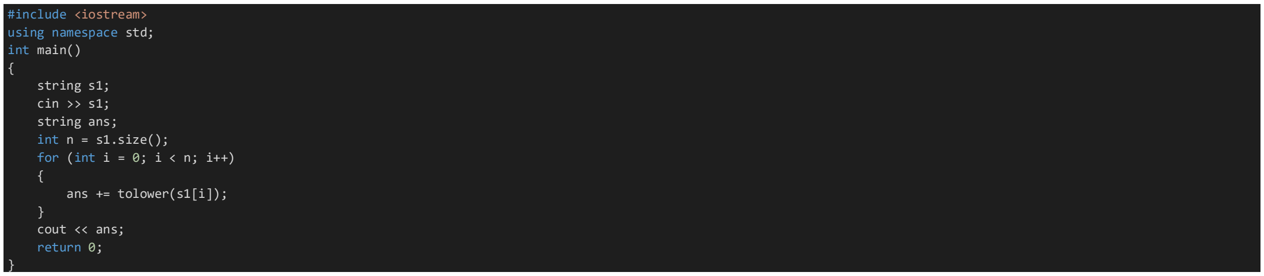
```
rashmi
```

Expected Output:

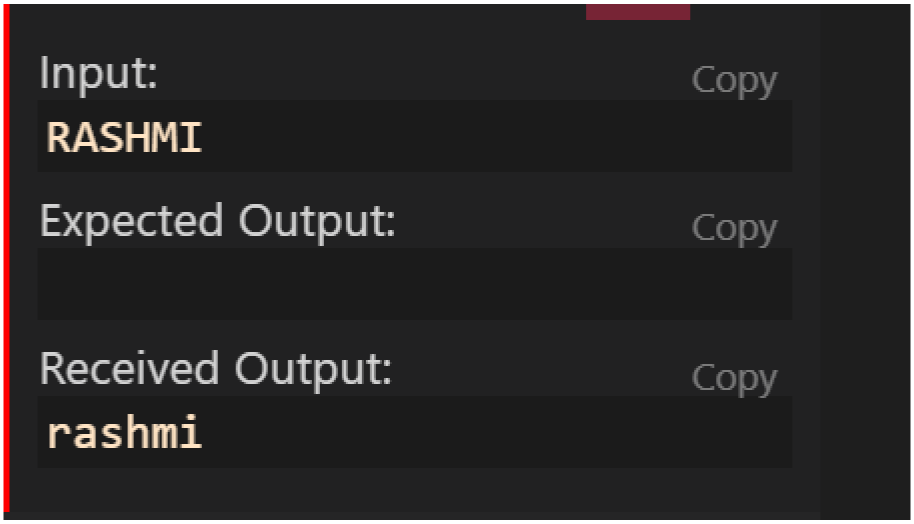
Received Output:

```
imhsar
```

9. Write a program to convert a string in lowercase.



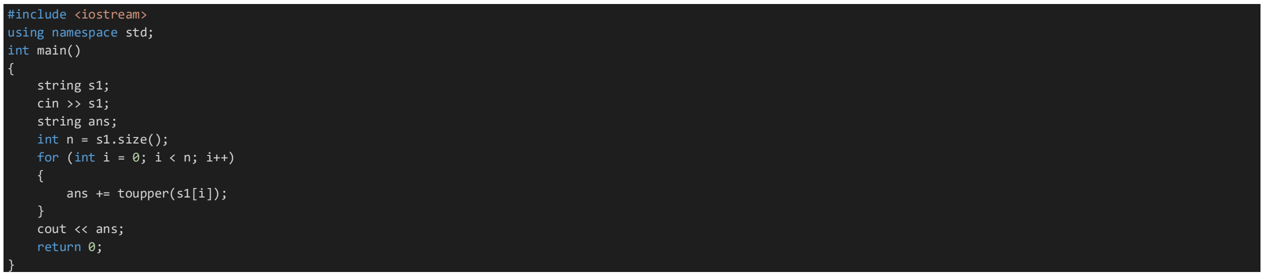
```
#include <iostream>
using namespace std;
int main()
{
    string s1;
    cin >> s1;
    string ans;
    int n = s1.size();
    for (int i = 0; i < n; i++)
    {
        ans += tolower(s1[i]);
    }
    cout << ans;
    return 0;
}
```



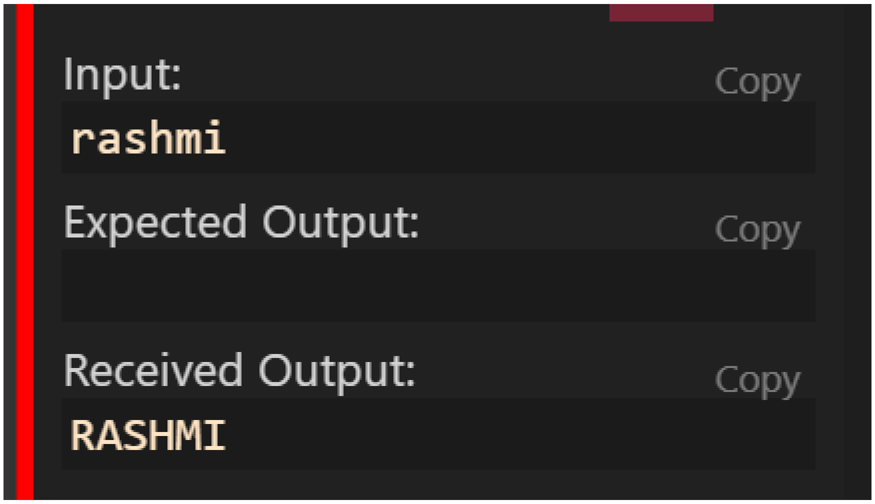
```
RASHMI
```

```
rashmi
```

10. Write a program to convert a string in uppercase.



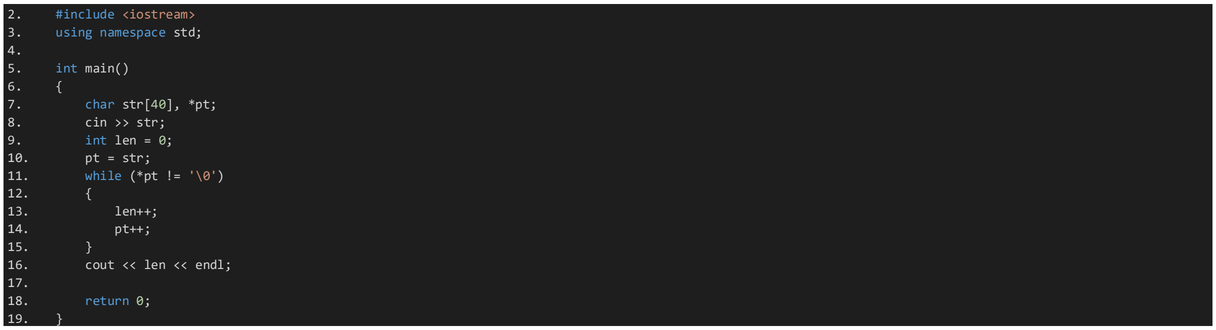
```
#include <iostream>
using namespace std;
int main()
{
    string s1;
    cin >> s1;
    string ans;
    int n = s1.size();
    for (int i = 0; i < n; i++)
    {
        ans += toupper(s1[i]);
    }
    cout << ans;
    return 0;
}
```



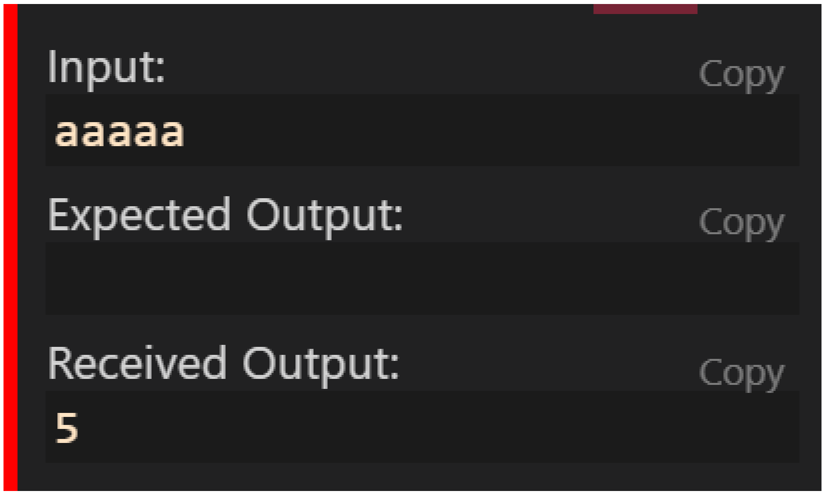
```
rashmi
```

```
RASHMI
```

1. Write a program to find the length of string.



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

int main()
{
    char str[40], *pt;
    cin >> str;
    int len = 0;
    pt = str;
    while (*pt != '\0')
    {
        len++;
        pt++;
    }
    cout << len << endl;

    return 0;
}
```



Input:

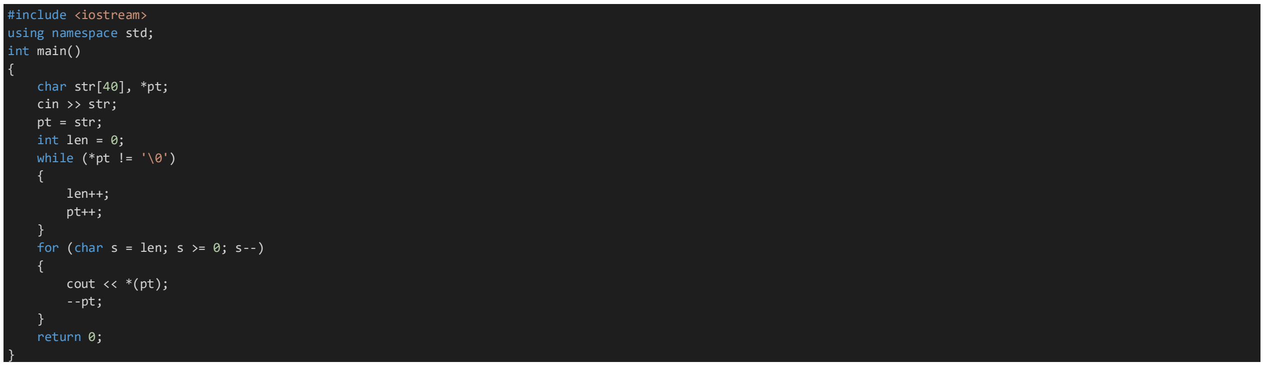
```
aaaaa
```

Expected Output:

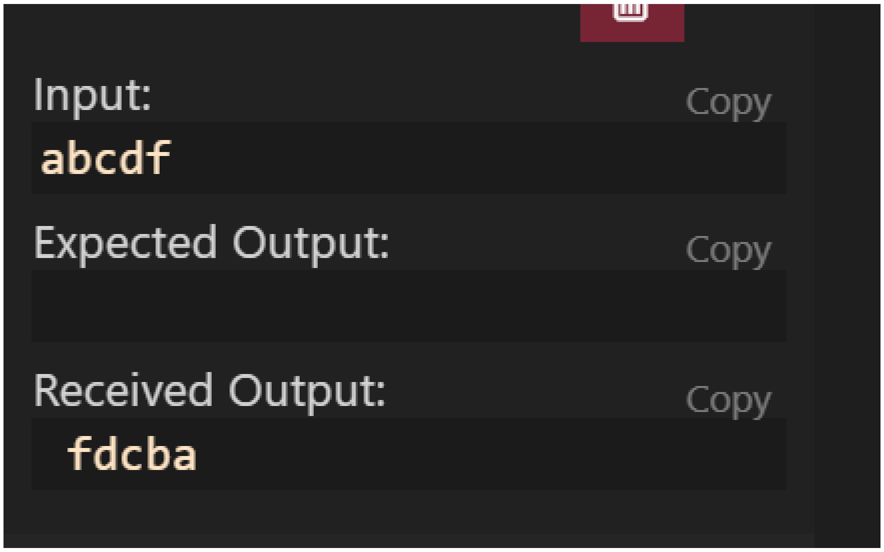
Received Output:

```
5
```

2. Write a program to display string from backward.



```
#include <iostream>
using namespace std;
int main()
{
    char str[40], *pt;
    cin >> str;
    pt = str;
    int len = 0;
    while (*pt != '\0')
    {
        len++;
        pt++;
    }
    for (char s = len; s >= 0; s--)
    {
        cout << *(pt);
        --pt;
    }
    return 0;
}
```



Input: Copy

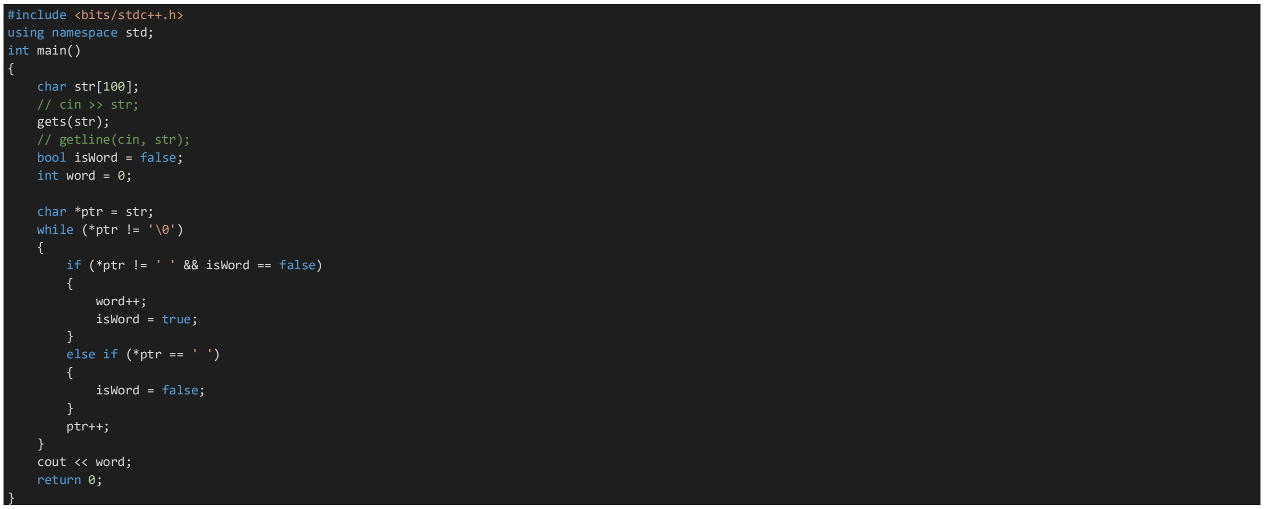
```
abcdf
```

Expected Output: Copy

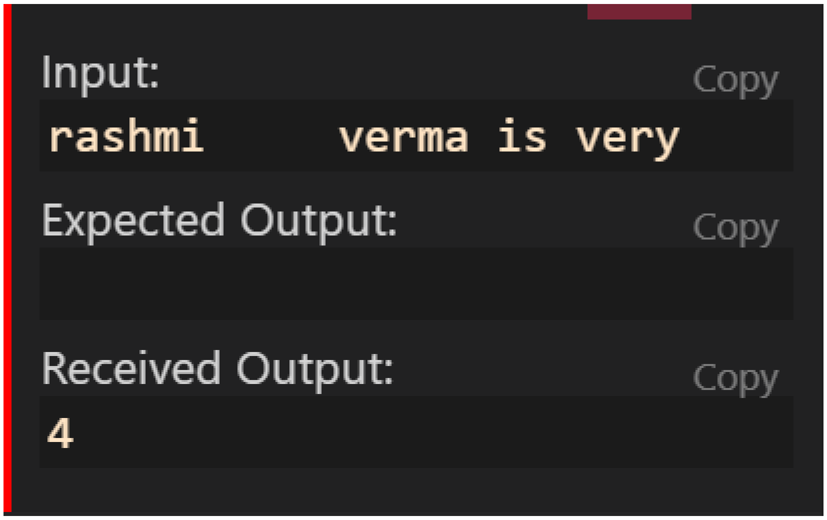
Received Output: Copy

```
 fdcba
```

3. Write a program to count number of words in string.



```
#include <bits/stdc++.h>
using namespace std;
int main()
{
    char str[100];
    // cin >> str;
    gets(str);
    // getline(cin, str);
    bool isWord = false;
    int word = 0;

    char *ptr = str;
    while (*ptr != '\0')
    {
        if (*ptr != ' ' && isWord == false)
        {
            word++;
            isWord = true;
        }
        else if (*ptr == ' ')
        {
            isWord = false;
        }
        ptr++;
    }
    cout << word;
    return 0;
}
```



Input: Copy

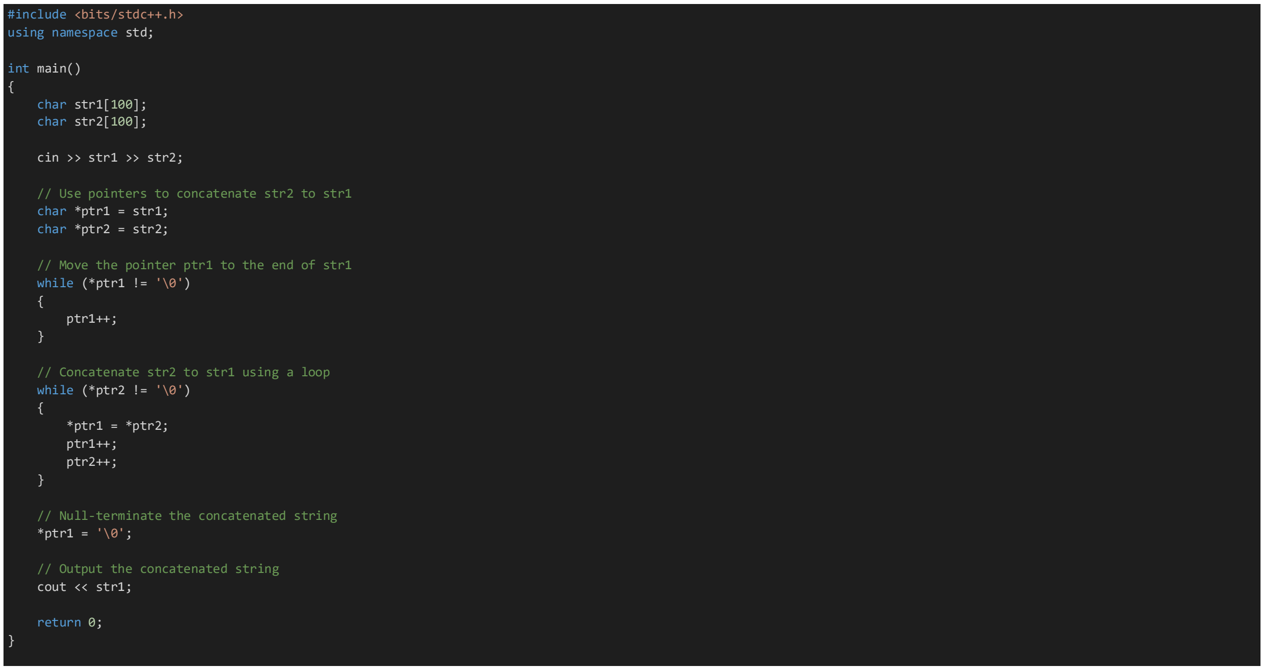
```
rashmi     verma is very
```

Expected Output: Copy

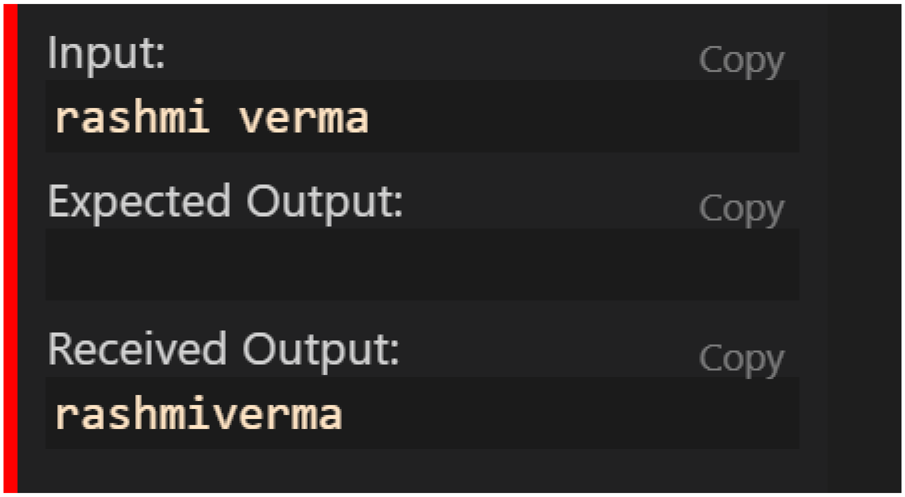
Received Output: Copy

```
4
```

4. Write a program to concatenate one string contents to another.



```
#include <bits/stdc++.h>
using namespace std;

int main()
{
    char str1[100];
    char str2[100];

    cin >> str1 >> str2;

    // Use pointers to concatenate str2 to str1
    char *ptr1 = str1;
    char *ptr2 = str2;

    // Move the pointer ptr1 to the end of str1
    while (*ptr1 != '\0')
    {
        ptr1++;
    }

    // Concatenate str2 to str1 using a loop
    while (*ptr2 != '\0')
    {
        *ptr1 = *ptr2;
        ptr1++;
        ptr2++;
    }

    // Null-terminate the concatenated string
    *ptr1 = '\0';

    // Output the concatenated string
    cout << str1;

    return 0;
}
```



Input:

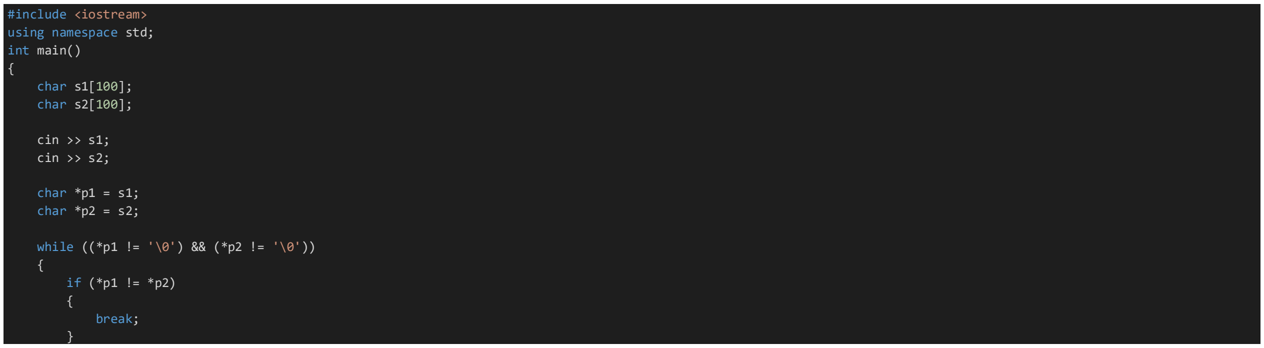
```
rashmi verma
```

Expected Output:

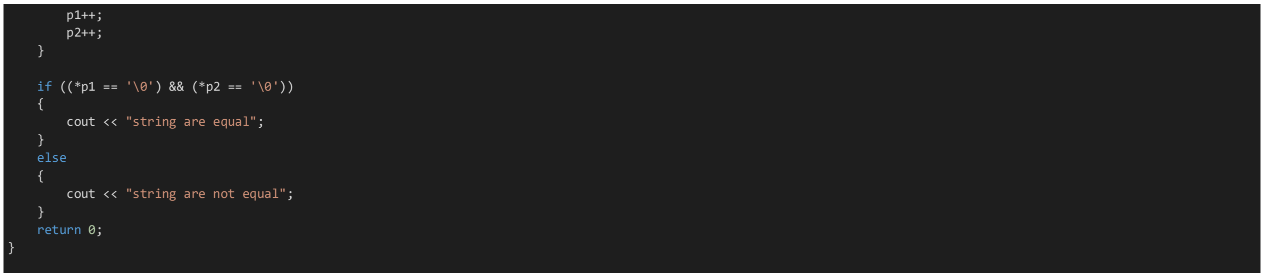
Received Output:

```
rashmiverma
```

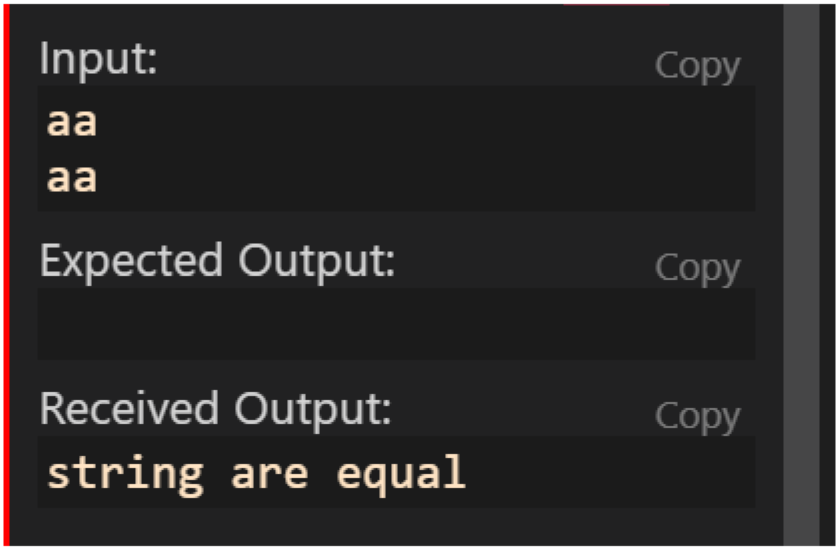
5. Write a program to compare two strings they are exact equal or not.



```
#include <iostream>
using namespace std;
int main()
{
    char s1[100];
    char s2[100];

    cin >> s1;
    cin >> s2;

    char *p1 = s1;
    char *p2 = s2;

    while ((*p1 != '\0') && (*p2 != '\0'))
    {
        if (*p1 != *p2)
        {
            break;
        }
```



```
        p1++;
        p2++;
    }

    if ((*p1 == '\0') && (*p2 == '\0'))
    {
        cout << "string are equal";
    }
    else
    {
        cout << "string are not equal";
    }
    return 0;
}
```



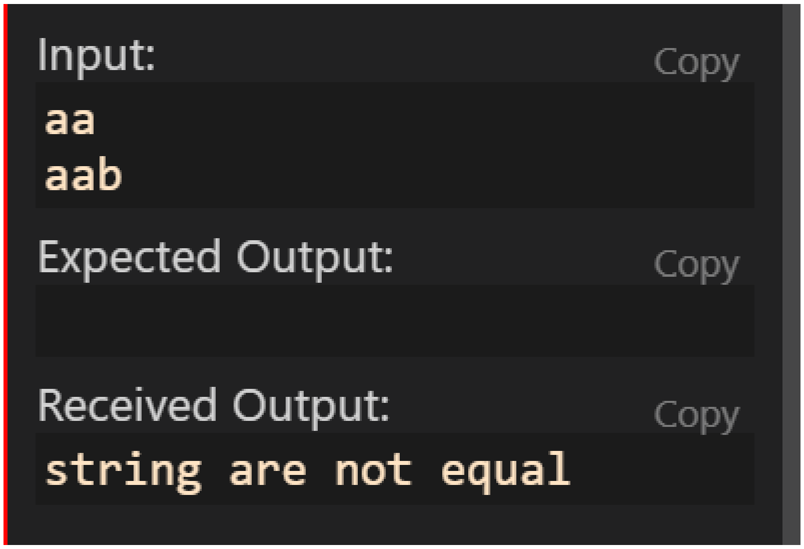
Input:

```
aa
aa
```

Expected Output:

Received Output:

```
string are equal
```



Input:

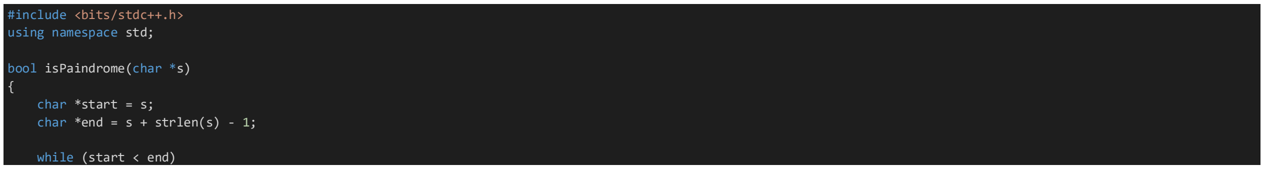
```
aa
aab
```

Expected Output:

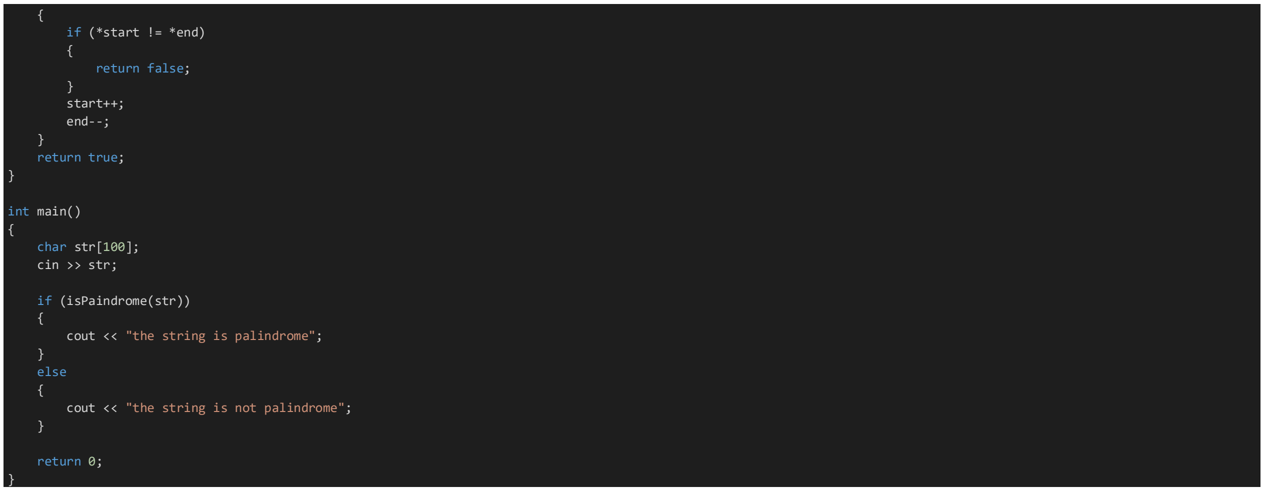
Received Output:

```
string are not equal
```

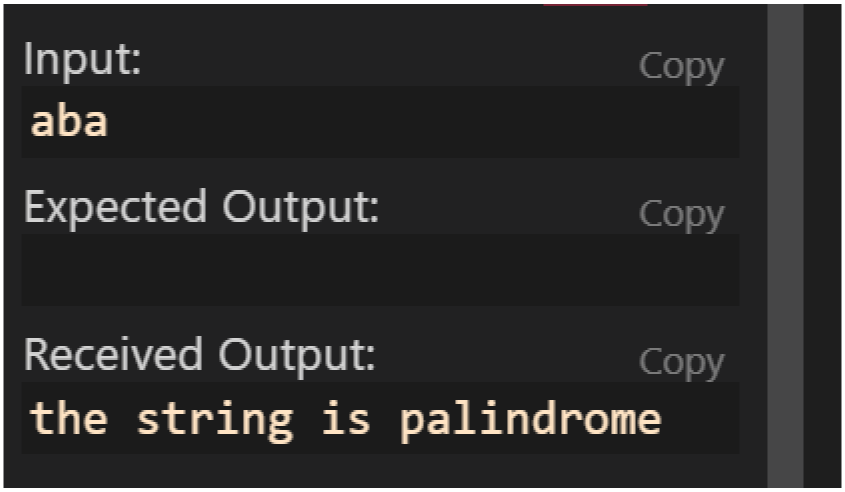
6. Write a program to check a string is palindrome or not.



```
#include <bits/stdc++.h>
using namespace std;

bool isPaindrome(char *s)
{
    char *start = s;
    char *end = s + strlen(s) - 1;

    while (start < end)
```



```
    {
        if (*start != *end)
        {
            return false;
        }
        start++;
        end--;
    }
    return true;
}

int main()
{
    char str[100];
    cin >> str;

    if (isPaindrome(str))
    {
        cout << "the string is palindrome";
    }
    else
    {
        cout << "the string is not palindrome";
    }

    return 0;
}
```



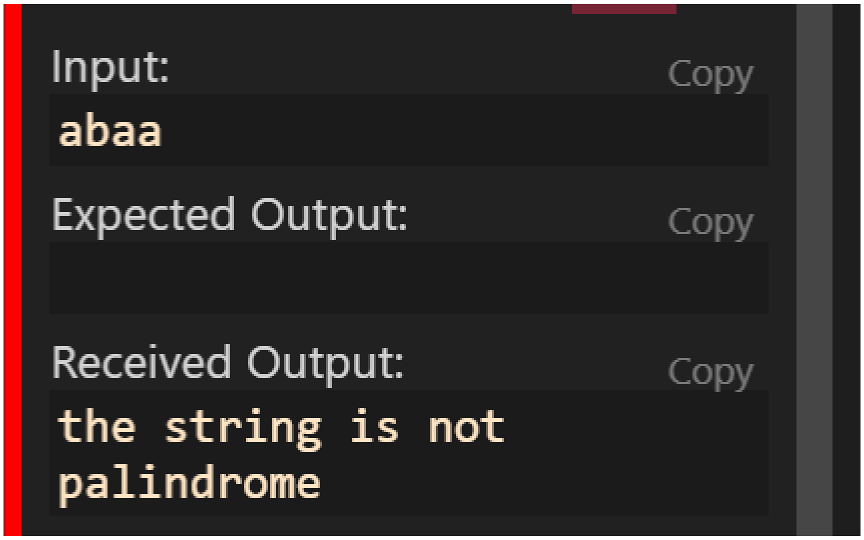
Input:

```
aba
```

Expected Output:

Received Output:

```
the string is palindrome
```



Input:

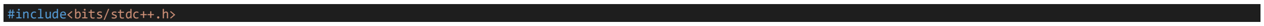
```
abaa
```

Expected Output:

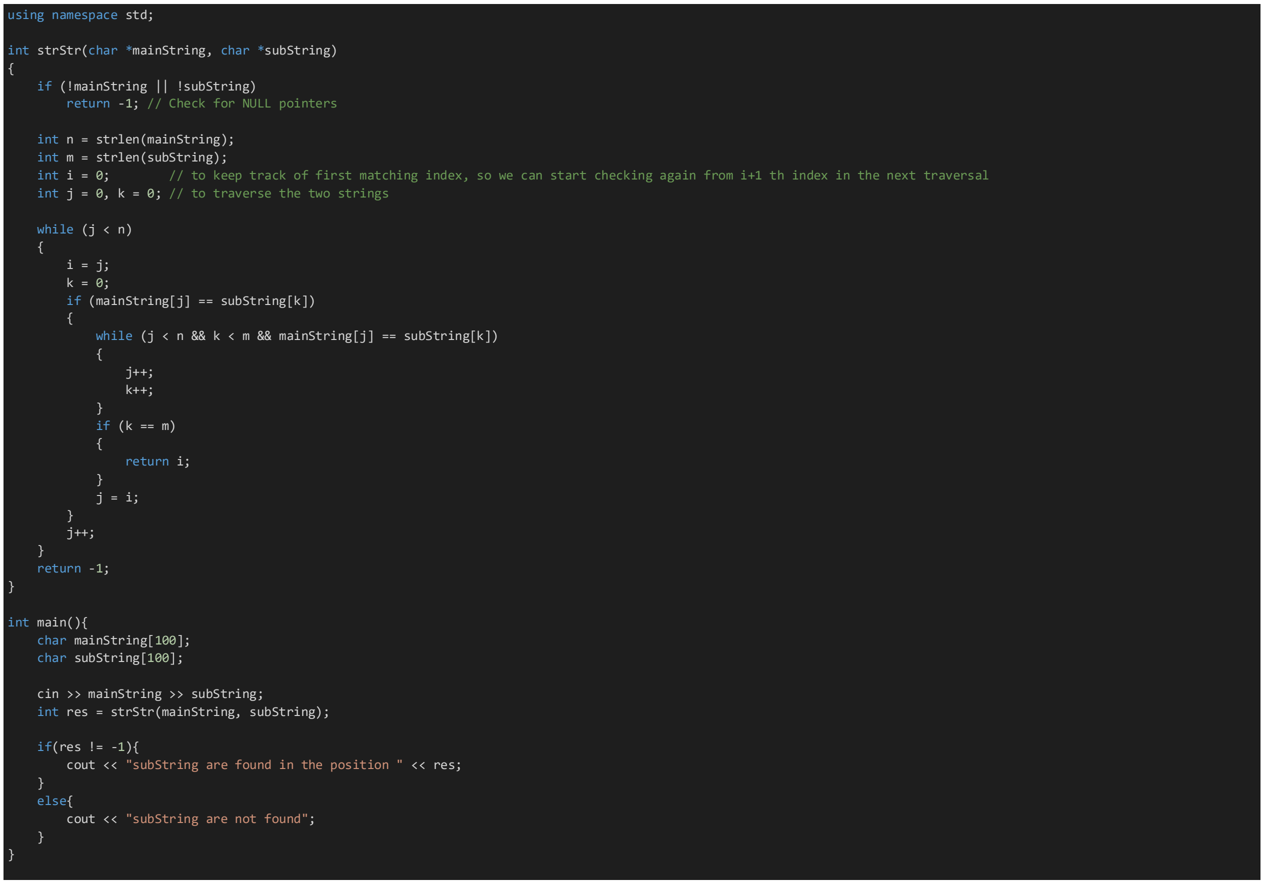
Received Output:

```
the string is not
palindrome
```

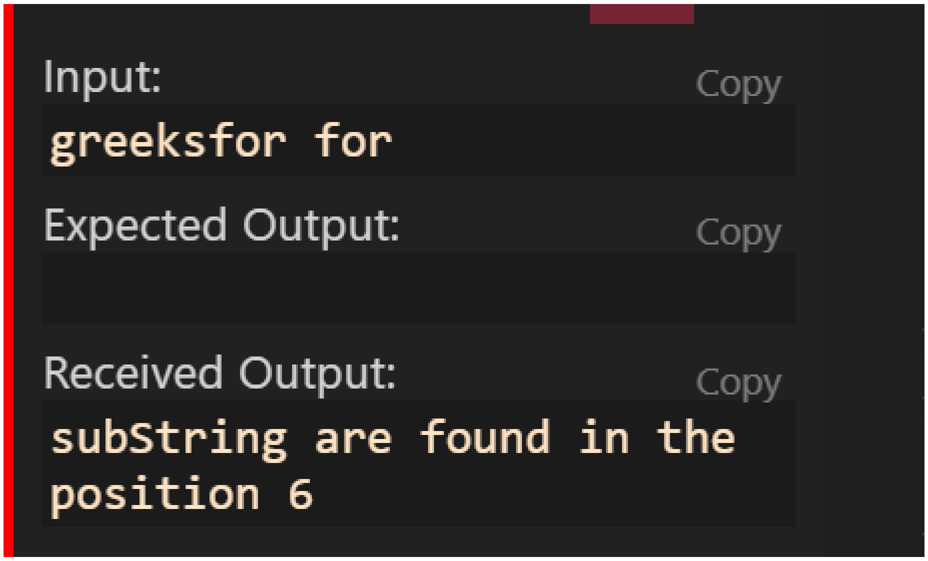
7. Write a program to find a substring within a string. If found display its starting position.



```
#include<bits/stdc++.h>
```



```
using namespace std;

int strStr(char *mainString, char *subString)
{
    if (!mainString || !subString)
        return -1; // Check for NULL pointers

    int n = strlen(mainString);
    int m = strlen(subString);
    int i = 0;         // to keep track of first matching index, so we can start checking again from i+1 th index in the next traversal
    int j = 0, k = 0; // to traverse the two strings

    while (j < n)
    {
        i = j;
        k = 0;
        if (mainString[j] == subString[k])
        {
            while (j < n && k < m && mainString[j] == subString[k])
            {
                j++;
                k++;
            }
            if (k == m)
            {
                return i;
            }
            j = i;
        }
        j++;
    }
    return -1;
}

int main(){
    char mainString[100];
    char subString[100];

    cin >> mainString >> subString;
    int res = strStr(mainString, subString);

    if(res != -1){
        cout << "subString are found in the position " << res;
    }
    else{
        cout << "subString are not found";
    }
}
```



Input:

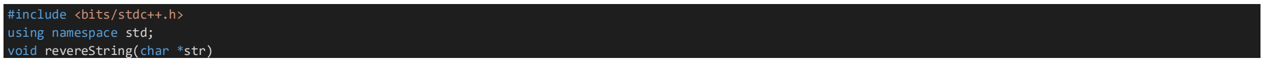
```
greeksfor for
```

Expected Output:

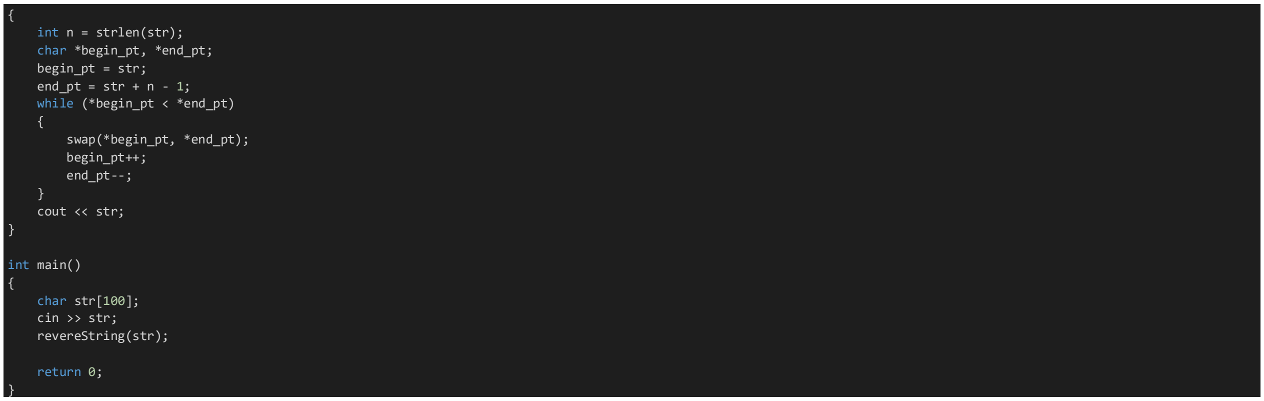
Received Output:

```
subString are found in the
position 6
```

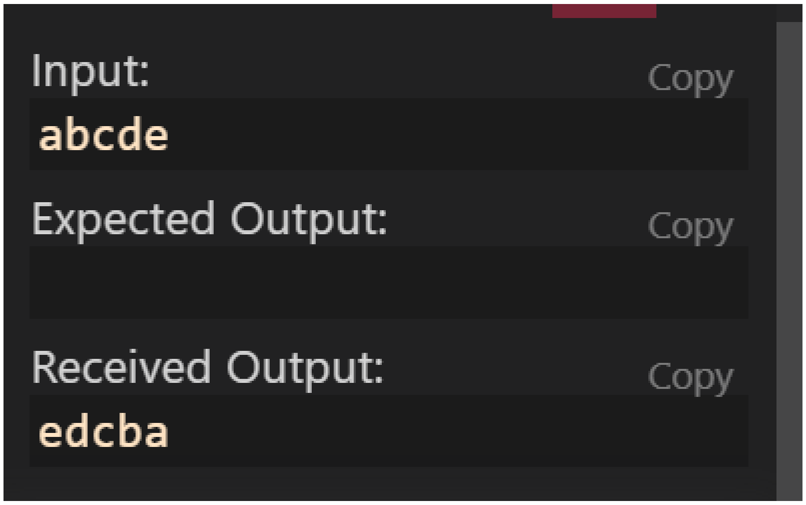
8. Write a program to reverse a string.



```
#include <bits/stdc++.h>
using namespace std;
void revereString(char *str)
```

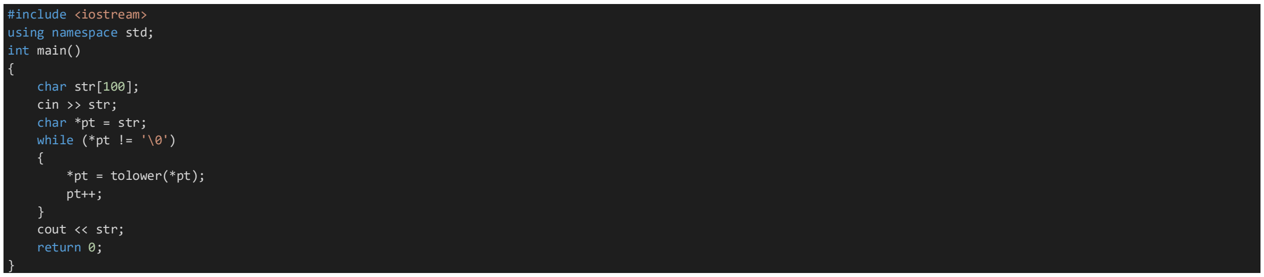


```
{
    int n = strlen(str);
    char *begin_pt, *end_pt;
    begin_pt = str;
    end_pt = str + n - 1;
    while (*begin_pt < *end_pt)
    {
        swap(*begin_pt, *end_pt);
        begin_pt++;
        end_pt--;
    }
    cout << str;
}

int main()
{
    char str[100];
    cin >> str;
    revereString(str);

    return 0;
}
```

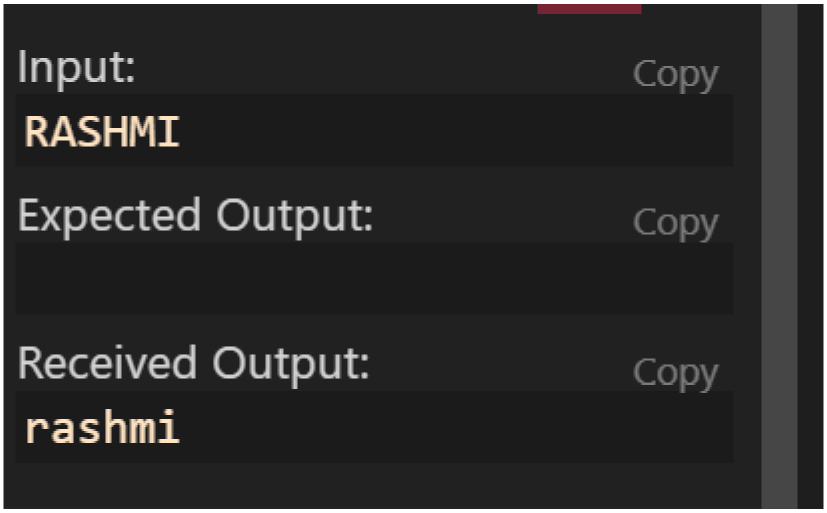


```
Input:
abcde
Expected Output:

Received Output:
edcba
```

9. Write a program to convert a string in lowercase.



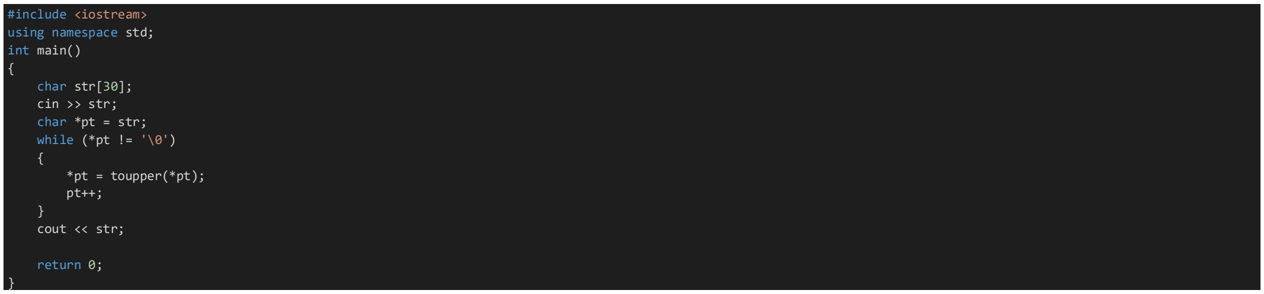
```
#include <iostream>
using namespace std;
int main()
{
    char str[100];
    cin >> str;
    char *pt = str;
    while (*pt != '\0')
    {
        *pt = tolower(*pt);
        pt++;
    }
    cout << str;
    return 0;
}
```



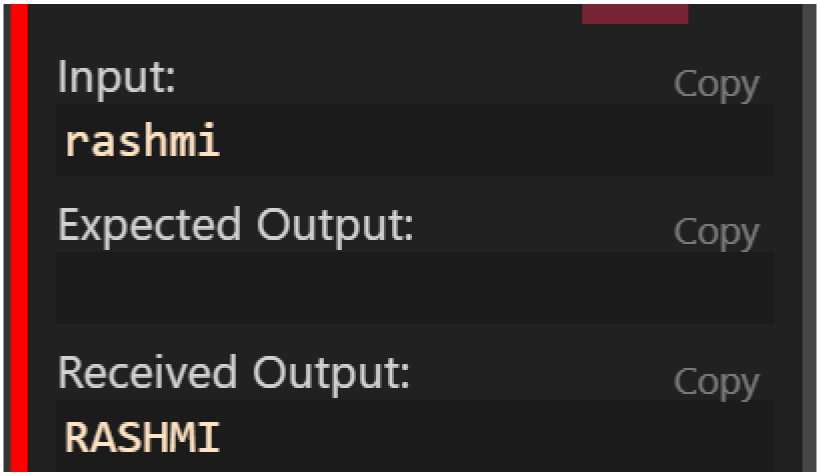
```
RASHMI
```

```
rashmi
```

10. Write a program to convert a string in uppercase.



```
#include <iostream>
using namespace std;
int main()
{
    char str[30];
    cin >> str;
    char *pt = str;
    while (*pt != '\0')
    {
        *pt = toupper(*pt);
        pt++;
    }
    cout << str;

    return 0;
}
```



```
rashmi
```

```
RASHMI
```