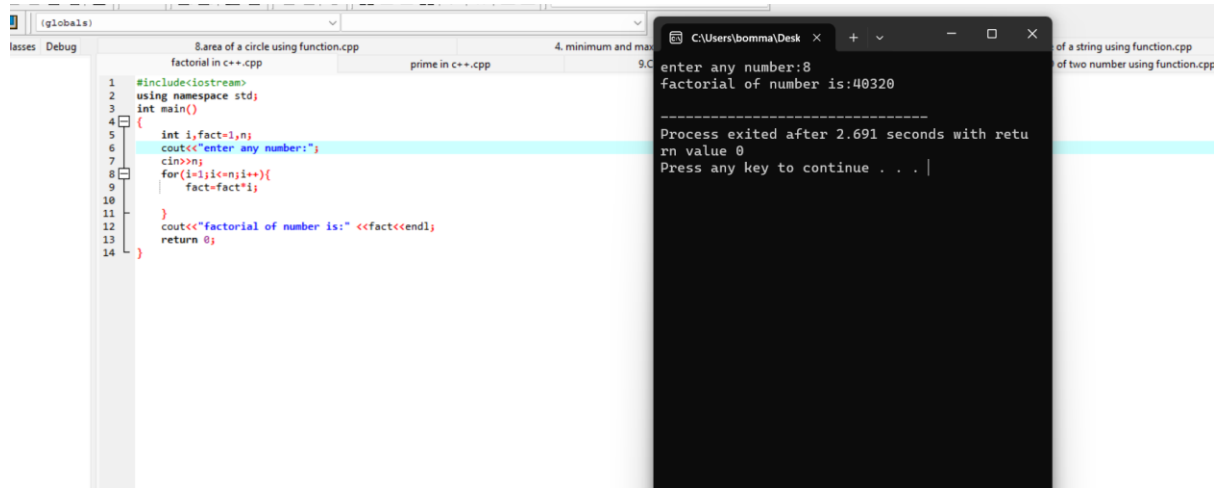


1. Factorial using function



The screenshot shows a C++ IDE with a file named 'factorial in c++.cpp'. The code defines a function to calculate the factorial of a number. The output window shows the program running with input 8, resulting in the factorial 40320.

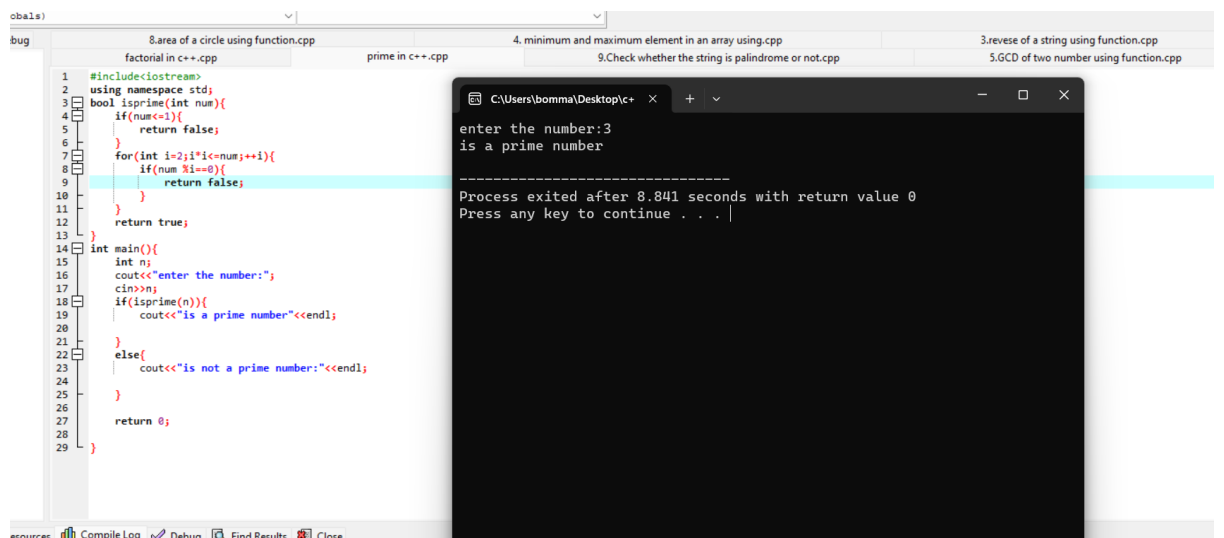
```
1 #include<iostream>
2 using namespace std;
3 int main()
4 {
5     int i,fact=1,n;
6     cout<<"enter any number:";
7     cin>>n;
8     for(i=1;i<=n;i++){
9         fact=fact*i;
10    }
11    cout<<"factorial of number is:" <<fact<<endl;
12    return 0;
13 }
```

Output:

```
enter any number:8
factorial of number is:40320

-----
Process exited after 2.691 seconds with return value 0
Press any key to continue . . .
```

2. Find prime number using function



The screenshot shows a C++ IDE with a file named 'prime in c++.cpp'. The code defines a function to check if a number is prime. The output window shows the program running with input 3, resulting in the output '3 is a prime number'.

```
1 #include<iostream>
2 using namespace std;
3 bool isprime(int num){
4     if(num<=1){
5         return false;
6     }
7     for(int i=2;i<=num;i++){
8         if(num%i==0){
9             return false;
10        }
11    }
12    return true;
13 }
14 int main(){
15     int n;
16     cout<<"enter the number:";
17     cin>>n;
18     if(isprime(n)){
19         cout<<"is a prime number"<<endl;
20     }
21     else{
22         cout<<"is not a prime number"<<endl;
23     }
24     return 0;
25 }
```

Output:

```
enter the number:3
3 is a prime number

-----
Process exited after 8.841 seconds with return value 0
Press any key to continue . . .
```

3. Find the reverse of a string using function

```
1 |
2 | #include <iostream>
3 | using namespace std;
4 | void reverseStr(string& str)
5 | {
6 |     int n = str.length();
7 |     for (int i = 0; i < n / 2; i++)
8 |         swap(str[i], str[n - i - 1]);
9 | }
10 | int main()
11 | {
12 |     string str = "geeksforgeeks";
13 |     reverseStr(str);
14 |     cout << str;
15 |     return 0;
16 | }
17 |
```

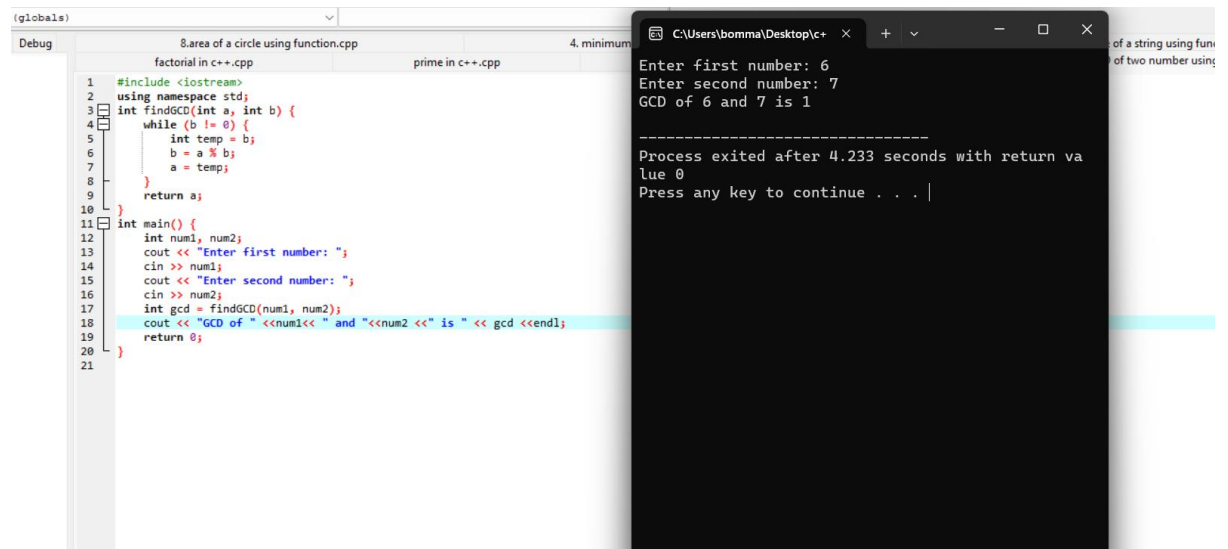
```
Process exited after 1.162 seconds with return value 0
Press any key to continue . . . |
```

4. Find minimum and maximum element in an array using function

```
1 | #include <bits/stdc++.h>
2 | using namespace std;
3 | int main()
4 | {
5 |     int arr[] = { 1, 45, 54, 71, 76, 12 };
6 |     int n = sizeof(arr) / sizeof(arr[0]);
7 |     cout << "Array: ";
8 |     for (int i = 0; i < n; i++)
9 |         cout << arr[i] << " ";
10 |    cout << "\nMin Element = "
11 |         << "min_element(arr, arr + n);
12 |         << "\nMax Element = "
13 |         << "max_element(arr, arr + n);
14 |    int &min = "min_element(arr, arr + n);
15 |    int &max = "max_element(arr, arr + n);
16 |    cout << "\nFinding the Element using address variable";
17 |    cout << "\nMin Element = " << min;
18 |    cout << "\nMax Element = " << max;
19 |    return 0;
20 | }
21 |
```

```
Array: 1 45 54 71 76 12
Min Element = 1
Max Element = 76
Finding the Element using address variable
Min Element = 1
Max Element = 76
Process exited after 0.4701 seconds with return value 0
Press any key to continue . . . |
```

5. Find GCD of two number using function



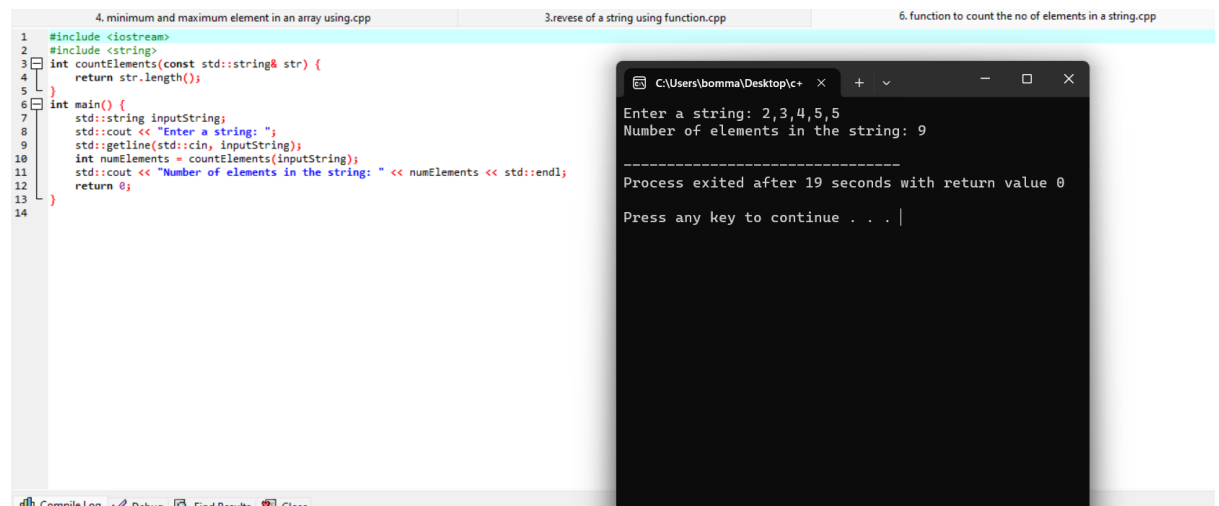
The screenshot shows a C++ IDE with a file named '8.area of a circle using function.cpp'. The code defines a function `findGCD` that uses the Euclidean algorithm to find the GCD of two numbers. The `main` function prompts the user to enter two numbers, 6 and 7, and then calls `findGCD` to calculate their GCD, which is 1. The output window shows the program's execution, including the input, the calculated GCD, and the exit time.

```
1 #include <iostream>
2 using namespace std;
3 int findGCD(int a, int b) {
4     while (b != 0) {
5         int temp = b;
6         b = a % b;
7         a = temp;
8     }
9     return a;
10 }
11 int main() {
12     int num1, num2;
13     cout << "Enter first number: ";
14     cin >> num1;
15     cout << "Enter second number: ";
16     cin >> num2;
17     int gcd = findGCD(num1, num2);
18     cout << "GCD of " << num1 << " and " << num2 << " is " << gcd << endl;
19     return 0;
20 }
21
```

Enter first number: 6
Enter second number: 7
GCD of 6 and 7 is 1

Process exited after 4.233 seconds with return value 0
Press any key to continue . . . |

6. Function to count the no of elements in a string



The screenshot shows a C++ IDE with a file named '6. function to count the no of elements in a string.cpp'. The code defines a function `countElements` that returns the length of a string. The `main` function prompts the user to enter a string, '2,3,4,5,5', and then calls `countElements` to count the number of elements, which is 9. The output window shows the program's execution, including the input, the counted number of elements, and the exit time.

```
1 #include <iostream>
2 #include <string>
3 int countElements(const std::string& str) {
4     return str.length();
5 }
6 int main() {
7     std::string inputString;
8     std::cout << "Enter a string: ";
9     std::getline(std::cin, inputString);
10    int numElements = countElements(inputString);
11    std::cout << "Number of elements in the string: " << numElements << std::endl;
12    return 0;
13 }
14
```

Enter a string: 2,3,4,5,5
Number of elements in the string: 9

Process exited after 19 seconds with return value 0
Press any key to continue . . . |

7. Convert Celsius and Fahrenheit using function

The screenshot shows a C++ IDE with a project named 'ebug'. The active file is '4. minimum and maximum element in an array using.cpp'. The code defines two functions: `celsiusToFahrenheit` and `fahrenheitToCelsius`, and a `main` function that prompts the user for temperature in Celsius and Fahrenheit, then converts them. The output window shows the program's execution: 'Enter temperature in Celsius: 33', '33 degrees Celsius is equal to 91.4 degrees Fahrenheit', 'Enter temperature in Fahrenheit: 98', '98 degrees Fahrenheit is equal to 36.6667 degrees Celsius', and 'Process exited after 10.73 seconds with return value 0'. The compilation results show 0 errors and 0 warnings.

```
1 #include <iostream>
2 double celsiusToFahrenheit(double celsius) {
3     return (celsius * 9.0 / 5.0) + 32.0;
4 }
5 double fahrenheitToCelsius(double fahrenheit) {
6     return (fahrenheit - 32.0) * 5.0 / 9.0;
7 }
8 int main() {
9     double celsiusTemperature;
10    std::cout << "Enter temperature in Celsius: ";
11    std::cin >> celsiusTemperature;
12    double fahrenheitTemperature = celsiusToFahrenheit(celsiusTemperature);
13    std::cout << celsiusTemperature << " degrees Celsius is equal to " << fahrenheitTemperature << " degrees Fahrenheit\n";
14    double fahrenheitInput;
15    std::cout << "Enter temperature in Fahrenheit: ";
16    std::cin >> fahrenheitInput;
17    double celsiusResult = fahrenheitToCelsius(fahrenheitInput);
18    std::cout << fahrenheitInput << " degrees Fahrenheit is equal to " << celsiusResult << " degrees Celsius\n";
19    return 0;
20 }
```

Resources: Compile Log, Debug, Find Results, Close

Compilation results...

Errors: 0

Warnings: 0

8.Find the area of a circle using function

The screenshot shows a C++ IDE with a project named 'ebug'. The active file is '4. minimum and maximum element in an array using.cpp'. The code defines a function `calculateCircleArea` that takes a radius and returns the area. The `main` function prompts the user for the radius, checks if it is negative, and then calculates the area. The output window shows the program's execution: 'Enter the radius of the circle: 45', 'The area of the circle with radius 45 is: 6361.73', and 'Process exited after 3.12 seconds with return value 0'. The compilation results show 0 errors and 0 warnings.

```
1 #include <iostream>
2 #include <cmath>
3 double calculateCircleArea(double radius) {
4     return M_PI * pow(radius, 2);
5 }
6 int main() {
7     double radius;
8     std::cout << "Enter the radius of the circle: ";
9     std::cin >> radius;
10    if (radius < 0) {
11        std::cerr << "Error: Radius cannot be negative." << std::endl;
12        return 1;
13    }
14    double area = calculateCircleArea(radius);
15    std::cout << "The area of the circle with radius " << radius << " is: " << area << std::endl;
16    return 0;
17 }
```

Resources: Compile Log, Debug, Find Results, Close

Compilation results...

Errors: 0

Warnings: 0

9.Check whether the string is palindrome or not

```
factorial in c++.cpp  prime in c++.cpp  9.Check whether the string is palindrom...  ing function.cpp
1  #include <iostream>
2  #include <string>
3  using namespace std;
4  bool isPalindrome(string str)
5  {
6      int left = 0;
7      int right = str.length() - 1;
8      while (left < right) {
9          if (str[left] != str[right])
10             return false;
11             left++;
12             right--;
13         }
14         return true;
15     }
16
17 int main() {
18     string str;
19     cout << "Enter a string: ";
20     getline(cin, str);
21
22     if (isPalindrome(str))
23         cout << str << " is a palindrome." << endl;
24     else
25         cout << str << " is not a palindrome." << endl;
26
27     return 0;
28 }
29
```

Enter a string: 34
34 is not a palindrome.

Process exited after 2.685 seconds with return value 0
Press any key to continue . . .

es Compile Log Debug Find Results Close
Compilation results...