

Meeting 7

Avicena Yosanta Muchtar 24/536710/PA/22764

22 October 2024

1 Question 1:

Question:

Write a program with some function to calculate simple mathematic operation, namely addition or difference!

Answer:

```
1
2  #include <iostream>
3  using namespace std;
4
5  int add(int a, int b) {
6      return a + b;
7  }
8
9  int subtract(int a, int b) {
10     return a - b;
11 }
12
13 int main() {
14     int num1, num2, choice;
15
16     cout << "Enter two numbers: ";
17     cin >> num1 >> num2;
18
19     cout << "Choose the operation:\n1. Addition\n2.
    ↳ Subtraction\n";
20     cin >> choice;
21
22     if (choice == 1) {
23         cout << "Result: " << add(num1, num2) << endl;
24     } else if (choice == 2) {
```

```

25     cout << "Result: " << subtract(num1, num2) << endl;
26 } else {
27     cout << "Invalid choice" << endl;
28 }
29
30     return 0;
31 }

```

2 Explanation

```

1  int add(int a, int b) {
2      return a + b;
3  }
4
5  int subtract(int a, int b) {
6      return a - b;
7  }
8

```

- Add(int a, int b): A function that takes two integer number (a and b) as input and returns the sum of the number.
- subtract(int a, int b): A function that takes two integer number as input and returns their difference (a - b).

```

1  int main() {
2      int num1, num2, choice;

```

Three integer variables are declared:

- num1: First number input entered by the user.
- num2: Second number input entered by the user.
- choice: Used to select the choice for addition or subtraction.

```

1      cout << "Enter two numbers: ";
2      cin >> num1 >> num2;

```

The program ask the user to enter two numbers.

cin >> num1 >> num2; store the numbers and stores them in the variables num1 and num2.

```

1      if (choice == 1) {
2          cout << "Result: " << add(num1, num2) << endl;
3      } else if (choice == 2) {

```

```
4         cout << "Result: " << subtract(num1, num2) << endl;
5     } else {
6         cout << "Invalid choice" << endl;
7     }
```

The if statement checks the value of choice:

- If choice is 1, the program use the add() function with num1 and num2 as arguments and prints the result.
- If choice is 2, the program use the subtract() function and prints the result.
- If the user enters anything other than 1 or 2, the program will print \Invalid choice."

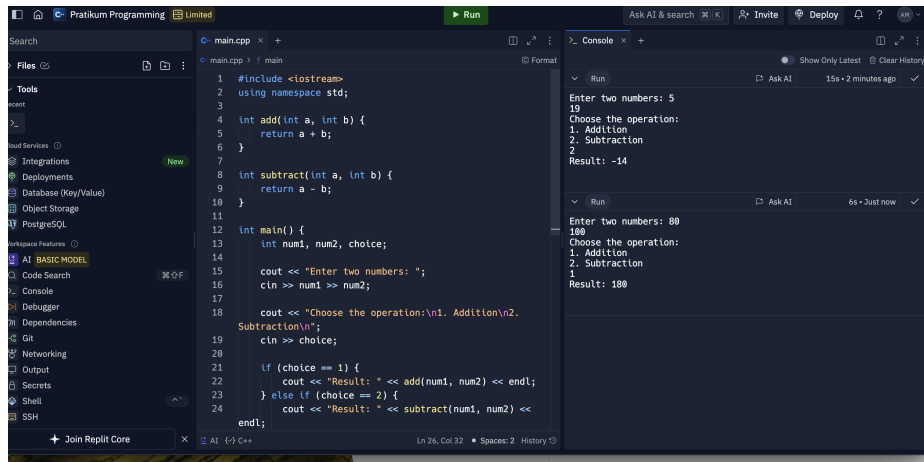


Figure 1: Ouput

3 Questio 2:

Question:

Write a program with recurrence function to calculate $x!$ (x factorial)!

```

1 #include <iostream>
2 using namespace std;
3
4 int factorial(int x) {
5     if (x == 0 || x == 1) {
6         return 1;
7     } else {
8         return x * factorial(x - 1);
9     }
10 }
11
12 int main() {
13     int num;
14
15     cout << "Enter a number to calculate its factorial: ";
16     cin >> num;
17
18     if (num < 0) {
19         cout << "Factorial is not defined for negative numbers."
20         << endl;
21     } else {
22         cout << "Factorial of " << num << " is: " <<
23         << factorial(num) << endl;
24     }
25 }

```

```

22     }
23
24     return 0;
25 }

```

4 Explanation

```

1     int factorial(int x) {
2     if (x == 0 || x == 1) {
3         return 1;
4     } else {
5         return x * factorial(x - 1);
6     }
7 }

```

- When `x == 0` or `x == 1`, the function returns 1. This is the stopping point for the recursion.
- If `x` is greater than 1, the function use (`factorial(x - 1)`) and multiplies `x` by the factorial of `x - 1`. This will go on until the base case of the program is reached.

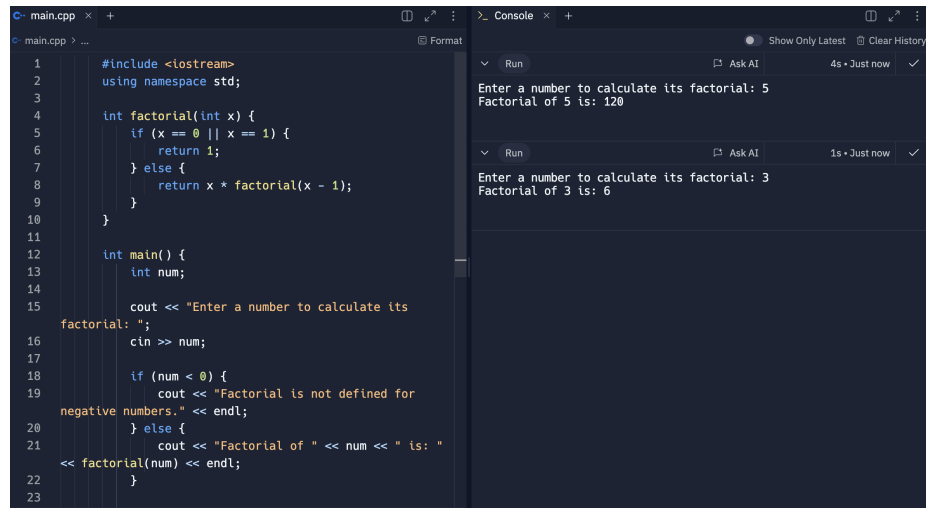
```

1     int main() {
2     int num;
3
4     cout << "Enter a number to calculate its factorial: ";
5     cin >> num;
6
7     if (num < 0) {
8         cout << "Factorial is not defined for negative numbers."
9         ↵ << endl;
10    } else {
11        cout << "Factorial of " << num << " is: " <<
12        ↵ factorial(num) << endl;
13    }
14    return 0;
15 }

```

User input:

- `cout << "Enter a number to calculate its factorial: ";`: Displays text asking the user to enter a number.
- `cin >> num;`: The user input is stored in the `num` variable.



The screenshot shows a C++ IDE with a file named `main.cpp` and a console window. The code in `main.cpp` defines a recursive function `factorial` and a `main` function. The `main` function prompts the user to enter a number, reads it, and then prints the factorial of that number. The console window shows two runs of the program. In the first run, the user enters 5, and the output is "Factorial of 5 is: 120". In the second run, the user enters 3, and the output is "Factorial of 3 is: 6".

```
1  #include <iostream>
2  using namespace std;
3
4  int factorial(int x) {
5      if (x == 0 || x == 1) {
6          return 1;
7      } else {
8          return x * factorial(x - 1);
9      }
10 }
11
12 int main() {
13     int num;
14
15     cout << "Enter a number to calculate its
16 factorial: ";
17     cin >> num;
18
19     if (num < 0) {
20         cout << "Factorial is not defined for
21 negative numbers." << endl;
22     } else {
23         cout << "Factorial of " << num << " is: "
24         << factorial(num) << endl;
25     }
26 }
```

Console Output:

```
Run
Enter a number to calculate its factorial: 5
Factorial of 5 is: 120

Run
Enter a number to calculate its factorial: 3
Factorial of 3 is: 6
```

Figure 2: Output

5 Question 3

Question:

Write a program with recurrence function to calculate GCD (Greater Common Division)!

```
1  #include <iostream>
2  using namespace std;
3
4  int gcd(int a, int b) {
5      if (b == 0) {
6          return a;
7      } else {
8          return gcd(b, a % b);
9      }
10 }
11
12 int main() {
13     int num1, num2;
14     cout << "Enter two numbers to calculate their GCD: ";
15     cin >> num1 >> num2;
16
17     cout << "The GCD of " << num1 << " and " << num2 << " is: "
18     << gcd(num1, num2) << endl;
19
20     return 0;
21 }
```

20 }

6 Explanation

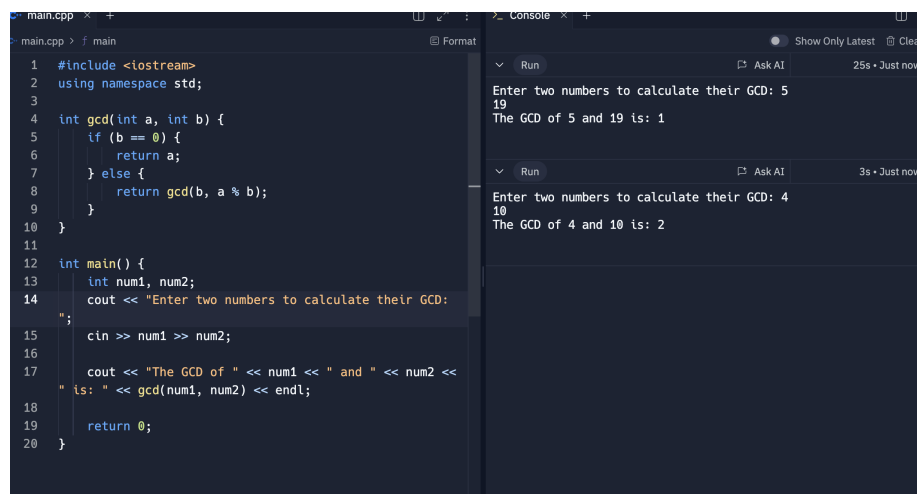
```
1 int gcd(int a, int b) {  
2     if (b == 0) {  
3         return a;  
4     } else {  
5         return gcd(b, a % b);  
6     }  
7 }
```

This program calculates GCD using Euclid's algorithm:

- The GCD of two numbers a and b can be calculated using the relation $\text{gcd}(a, b) = \text{gcd}(b, a \% b)$, where % is the modulo operation.
- This continues until b becomes 0, at which point a will be the GCD.
- When $b == 0$, the GCD is a. The number stops at this point.
- If b is not 0, the function will use the arguments $\text{gcd}(b, a \% b)$. This reduces the problem to a simple form by applying the operation.

```
1 int main() {  
2     int num1, num2;  
3     cout << "Enter two numbers to calculate their GCD: ";  
4     cin >> num1 >> num2;  
5  
6     cout << "The GCD of " << num1 << " and " << num2 << " is: "  
7     ↪ << gcd(num1, num2) << endl;  
8  
9     return 0;  
}
```

- The program asks the user to enter two number, num1 and num2, using `cin >> num1 >> num2;`.
- The function `gcd(num1, num2)` is used to compute the GCD of the two numbers inputted by the user.



The image shows a C++ IDE with a code editor on the left and a console on the right. The code editor contains a C++ program that calculates the Greatest Common Divisor (GCD) of two numbers. The program includes `<iostream>` and uses the `std` namespace. It defines a `gcd` function that uses a recursive algorithm. The `main` function prompts the user to enter two numbers, reads them, and then prints the GCD of the two numbers.

```
1 #include <iostream>
2 using namespace std;
3
4 int gcd(int a, int b) {
5     if (b == 0) {
6         return a;
7     } else {
8         return gcd(b, a % b);
9     }
10 }
11
12 int main() {
13     int num1, num2;
14     cout << "Enter two numbers to calculate their GCD: ";
15     cin >> num1 >> num2;
16     cout << "The GCD of " << num1 << " and " << num2 <<
17     " is: " << gcd(num1, num2) << endl;
18     return 0;
19 }
```

The console on the right shows two test runs. The first run shows the user entering 5 and 19, and the program outputting "The GCD of 5 and 19 is: 1". The second run shows the user entering 4 and 10, and the program outputting "The GCD of 4 and 10 is: 2".

Figure 3: Test Output