

1. Write a program to display a welcome message using a function without parameters and return value.

| main.c  | Output  |
|---|---|
| <pre>1 #include &lt;stdio.h&gt; 2 3 void displayWelcomeMessage() { 4     printf("Welcome to C Programming!\n"); 5 } 6 7 int main() { 8     displayWelcomeMessage(); 9     return 0; 10 } 11</pre> | <pre>Welcome to C Programming!  === Code Execution Successful ===</pre> |

2. Write a program where a function accepts two numbers as arguments and prints their sum. Use a function with parameters and no return value.

| main.c   | Output  |
|--|---|
| <pre>1 #include &lt;stdio.h&gt; 2 3 void printSum(int a, int b) { 4     printf("The sum is: %d\n", a + b); 5 } 6 7 int main() { 8     int num1, num2; 9     printf("Enter two numbers: "); 10    scanf("%d %d", &amp;num1, &amp;num2); 11    printSum(num1, num2); 12    return 0; 13 } 14</pre> | <pre>Enter two numbers: 7 7 The sum is: 14  === Code Execution Successful ===</pre> |

3. Write a program where a function takes two integers as parameters, compares them, and returns the maximum value.

| main.c  | Output   |
|---|--|
| <pre>1 #include &lt;stdio.h&gt; 2 3 int findMax(int a, int b) { 4     return (a &gt; b) ? a : b; 5 } 6 7 int main() { 8     int num1, num2, max; 9     printf("Enter two numbers: "); 10    scanf("%d %d", &amp;num1, &amp;num2); 11    max = findMax(num1, num2); 12    printf("The maximum value is: %d\n", max); 13    return 0; 14 } 15</pre> | <pre>Enter two numbers: 7 77 The maximum value is: 77  === Code Execution Successful ===</pre> |

4. Write a program to compute the factorial of a number using a recursive function.

| main.c  | Run | Output   |
|---|-----|--|
| <pre> 1 #include &lt;stdio.h&gt; 2 int factorial(int n) { 3     if (n==0) 4         return 1; 5     else 6         return n*factorial(n-1); 7 } 8 int main() { 9     int num; 10    printf("Enter a numbers: "); 11    scanf("%d",&amp;num); 12    printf("The factorial of %d is: %d\n",num,factorial(num)); 13    return 0; 14 } </pre> | Run | <pre> Enter a numbers: 7 The factorial of 7 is: 5040  === Code Execution Successful === </pre> |

5. Write a program to demonstrate swapping two numbers using a function with call by value.

| main.c   | Run | Output   |
|--|-----|--|
| <pre> 1 #include &lt;stdio.h&gt; 2 void swap(int a, int b) { 3     int temp = a; 4     a = b; 5     b = temp; 6     printf("After swapping: a = %d, b = %d\n", a, b); 7 } 8 int main() { 9     int num1, num2; 10    printf("Enter two numbers: "); 11    scanf("%d %d", &amp;num1, &amp;num2); 12    printf("Before swapping: a = %d, b = %d\n", num1, num2); 13    swap(num1, num2); 14    printf("After swapping in main: a = %d, b = %d\n", num1, num2); 15    return 0; 16 } </pre> | Run | <pre> Enter two numbers: 7 8 Before swapping: a = 7, b = 8 After swapping: a = 8, b = 7 After swapping in main: a = 7, b = 8  === Code Execution Successful === </pre> |

6. Write a program where a function accepts an array and its size as arguments and returns the largest value in the array.  
Code

```

#include <stdio.h>
int findLargest(int arr[], int size) {
    int max = arr[0];
    for (int i = 1; i < size; i++) {
        if (arr[i] > max) {
            max = arr[i];
        }
    }
    return max;
}
int main() {
    int n;
    printf("Enter the size of the array: ");
    scanf("%d", &n);
    if (n <= 0) {
        printf("Invalid size. Size must be greater than 0.\n");
        return 1;
    }
    int arr[100];
    if (n > 100) {
        printf("Array size exceeds limit.\n");
        return 1;
    }

```

```

    }
    printf("Enter the array elements: ");
    for (int i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }
    printf("Array elements are: ");
    for (int i = 0; i < n; i++) {
        printf("%d ", arr[i]);
    }
    printf("\n");
    printf("The largest value is: %d\n", findLargest(arr, n));
    return 0;
}

```

Output

The screenshot shows a code editor with a file named 'main.c'. The code is as follows:

```

23 }
24 printf("Enter the array elements: ");
25 for (int i = 0; i < n; i++) {
26     scanf("%d", &arr[i]);
27 }
28 printf("Array elements are: ");
29 for (int i = 0; i < n; i++) {
30     printf("%d ", arr[i]);
31 }
32 printf("\n");
33 printf("The largest value is: %d\n", findLargest(arr,
34     n));
35 return 0;
36 }

```

The 'Output' pane on the right shows the following text:

```

Enter the size of the array: 4
Enter the array elements: 1 2 3 4
Array elements are: 1 2 3 4
The largest value is: 4

=== Code Execution Successful ===

```

7. Write a program in C to print all perfect numbers in given range using the function.

Code

```

#include <stdio.h>
int isPerfect(int num) {
    int sum = 0;
    for (int i = 1; i <= num / 2; i++) {
        if (num % i == 0)
            sum += i;
    }
    return (sum == num);
}
int main() {
    int start, end;
    printf("Enter the range (start and end): ");
    scanf("%d %d", &start, &end);
    printf("Perfect numbers in the range are: ");
    for (int i = start; i <= end; i++) {
        if (isPerfect(i))
            printf("%d ", i);
    }
    printf("\n");
    return 0;
}

```

output

The screenshot shows a C program in a code editor. The code defines a function `isPerfect` that checks if a number is perfect by summing its divisors. The `main` function prompts the user for a range (start and end) and prints all perfect numbers within that range. The output window shows the user inputting '8 888' and the program outputting 'Perfect numbers in the range are: 28 496'. A status message '=== Code Execution Successful ===' is also displayed.

```
main.c
1 #include <stdio.h>
2 int isPerfect(int num) {
3     int sum = 0;
4     for (int i = 1; i <= num / 2; i++) {
5         if (num % i == 0)
6             sum += i;
7     }
8     return (sum == num);
9 }
10 int main() {
11     int start, end;
12     printf("Enter the range (start and end): ");
13     scanf("%d %d", &start, &end);
14     printf("Perfect numbers in the range are: ");
15     for (int i = start; i <= end; i++) {
16         if (isPerfect(i))
17             printf("%d ", i);
18     }
19 }
```

Output

Enter the range (start and end): 8 888  
Perfect numbers in the range are: 28 496

=== Code Execution Successful ===

8. Write a program to reverse a number using function?(Get the input from user)

The screenshot shows a C program in a code editor. The code defines a function `reverseNumber` that reverses a number by repeatedly extracting the last digit and building the reversed number. The `main` function prompts the user for a number and prints the reversed number. The output window shows the user inputting '677' and the program outputting 'The reversed number is: 776'. A status message '=== Code Execution Successful ===' is also displayed.

```
main.c
1 #include <stdio.h>
2 int reverseNumber(int num) {
3     int reverse = 0;
4     while (num != 0) {
5         reverse = reverse * 10 + (num % 10);
6         num /= 10;
7     }
8     return reverse;
9 }
10 int main() {
11     int num;
12     printf("Enter a number: ");
13     scanf("%d", &num);
14     printf("The reversed number is: %d\n", reverseNumber(num));
15     return 0;
16 }
17 }
```

Output

Enter a number: 677  
The reversed number is: 776

=== Code Execution Successful ===

9. Write a menu-driven program where each arithmetic operation is implemented using a separate function.

```
#include <stdio.h>
```

```
void add(int a, int b) {
    printf("Result: %d\n", a + b);
}
```

```
void subtract(int a, int b) {
    printf("Result: %d\n", a - b);
}
```

```
void multiply(int a, int b) {
    printf("Result: %d\n", a * b);
}
```

```
void divide(int a, int b) {
    if (b != 0)
        printf("Result: %.2f\n", (float)a / b);
    else
        printf("Division by zero is not allowed.\n");
}
```

```

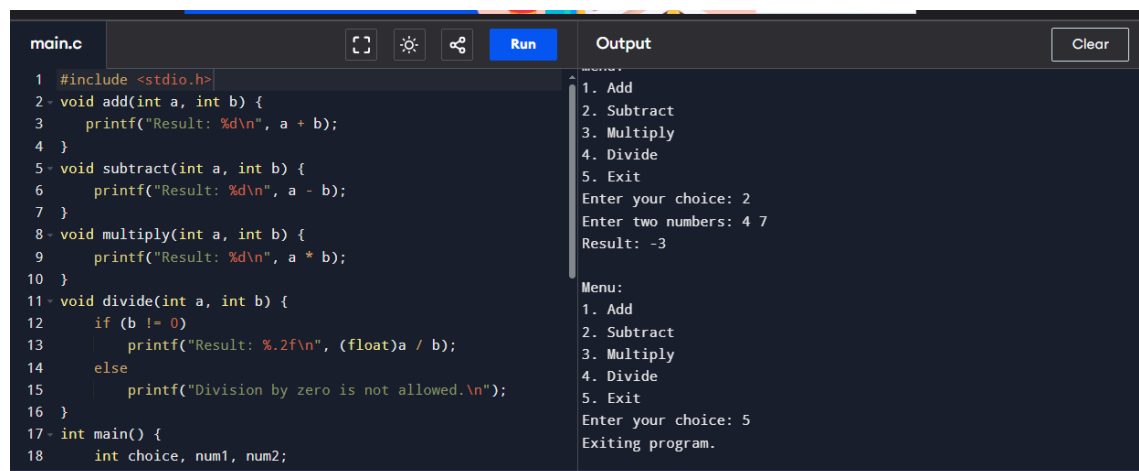
int main() {
    int choice, num1, num2;
    do {
        printf("\nMenu:\n");
        printf("1. Add\n2. Subtract\n3. Multiply\n4. Divide\n5. Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);

        if (choice >= 1 && choice <= 4) {
            printf("Enter two numbers: ");
            scanf("%d %d", &num1, &num2);
        }

        switch (choice) {
            case 1: add(num1, num2); break;
            case 2: subtract(num1, num2); break;
            case 3: multiply(num1, num2); break;
            case 4: divide(num1, num2); break;
            case 5: printf("Exiting program.\n"); break;
            default: printf("Invalid choice. Try again.\n");
        }
    } while (choice != 5);

    return 0;
}

```



The screenshot shows a C++ IDE with a file named `main.c`. The code defines a menu-based calculator with functions for addition, subtraction, multiplication, and division. The `main` function uses a `do-while` loop to repeatedly show the menu and process user input until the user chooses to exit (option 5).

The **Output** pane on the right shows the program's execution: it displays the menu, prompts for a choice (2), prompts for two numbers (4 and 7), and displays the result (-3). It then shows the menu again, prompts for a choice (5), and displays "Exiting program."

```

main.c
1 #include <stdio.h>
2 void add(int a, int b) {
3     printf("Result: %d\n", a + b);
4 }
5 void subtract(int a, int b) {
6     printf("Result: %d\n", a - b);
7 }
8 void multiply(int a, int b) {
9     printf("Result: %d\n", a * b);
10 }
11 void divide(int a, int b) {
12     if (b != 0)
13         printf("Result: %.2f\n", (float)a / b);
14     else
15         printf("Division by zero is not allowed.\n");
16 }
17 int main() {
18     int choice, num1, num2;

```

**Output**

```

1. Add
2. Subtract
3. Multiply
4. Divide
5. Exit
Enter your choice: 2
Enter two numbers: 4 7
Result: -3

Menu:
1. Add
2. Subtract
3. Multiply
4. Divide
5. Exit
Enter your choice: 5
Exiting program.

```