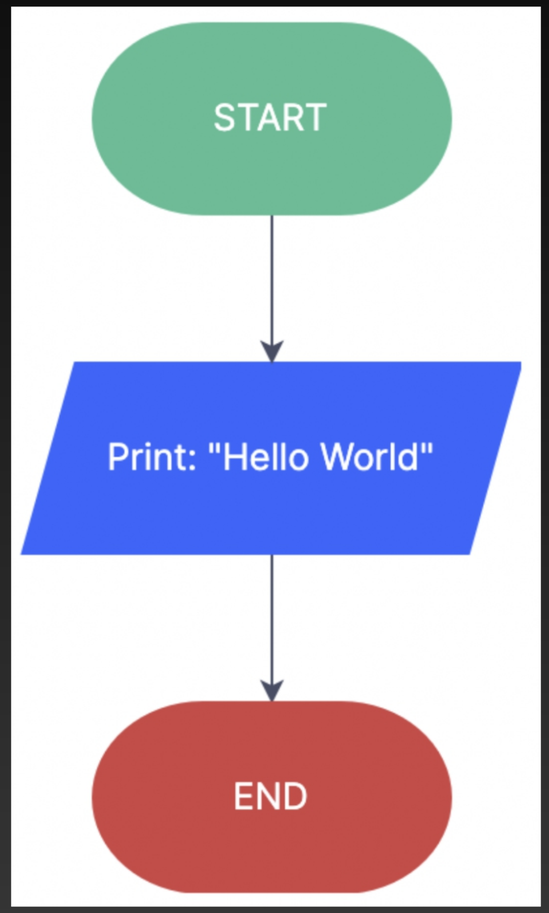
Lab f ile

C Programming

Arpit Tomar
590018317

Print"Hello World"

Flowchart



Print”Hello world”

Algorithm

Step 1:Start
STEP 2: Initialise: No specific initialisation is required for this task.
STEP 3: Display: Output the string "Hello, World!" to the screen.
STEP 4: End

Print"Hello world"

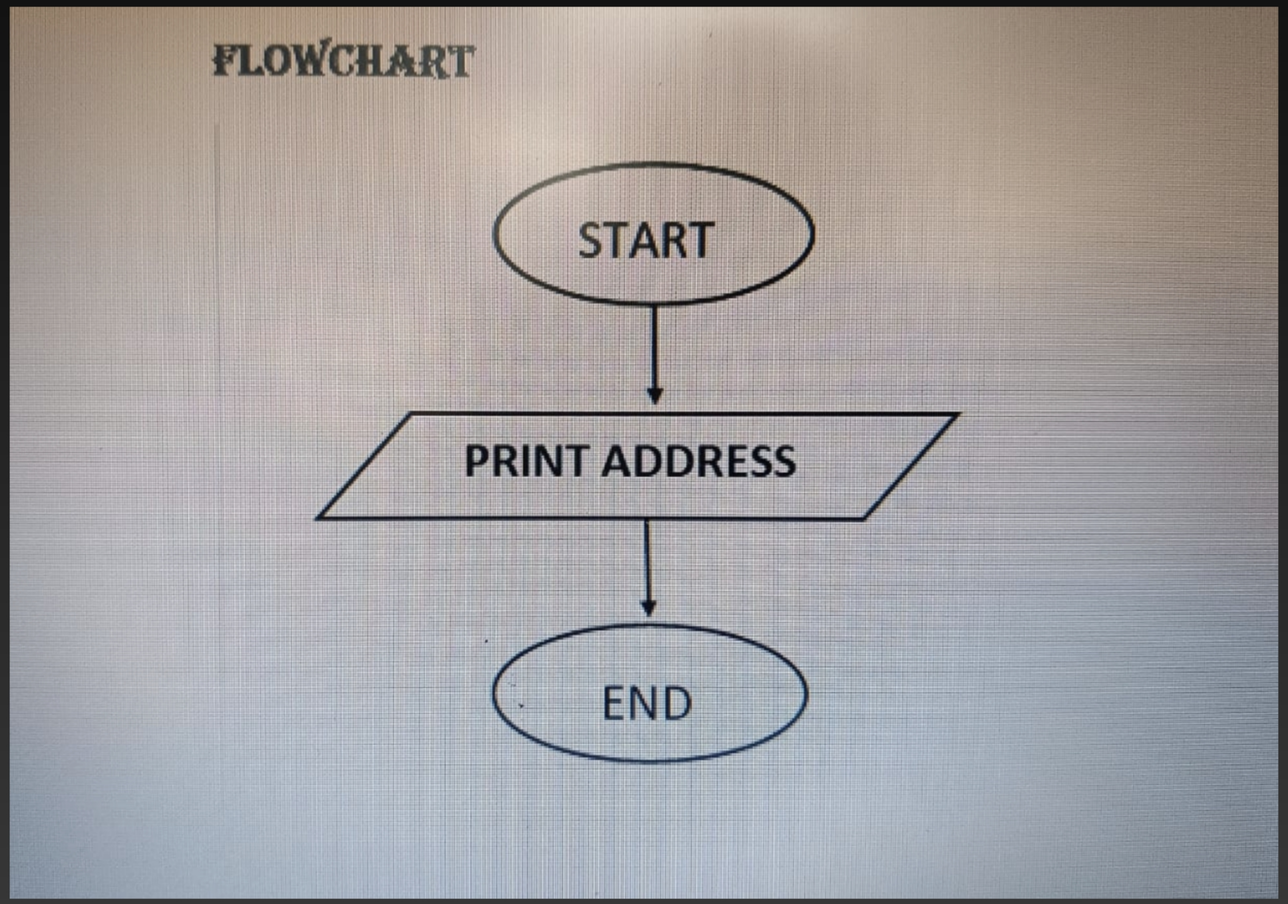
Syntax

```c
#include <stdio.h>

int main() {
    printf("Hello World");
    return 0;
}

```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

tanishq@Arpits-MacBook-Air T % cd "/var/folders/3y/blsgfgyd2rqgvhz7jn6ggpl80000gn/T/" && gcc tempCodeRunnerFile.c -o tempCodeRunnerFile && "/var/folders/3y/
blsgfgyd2rqgvhz7jn6ggpl80000gn/T/"tempCodeRunnerFile
Hello World%
tanishq@Arpits-MacBook-Air T %
```

Printing the address in multiple lines

Flowchart



Printing the address in multiple lines

Algorithm

Step 1: Start

Step 2: Include necessary header

Step 3: Declare a character array

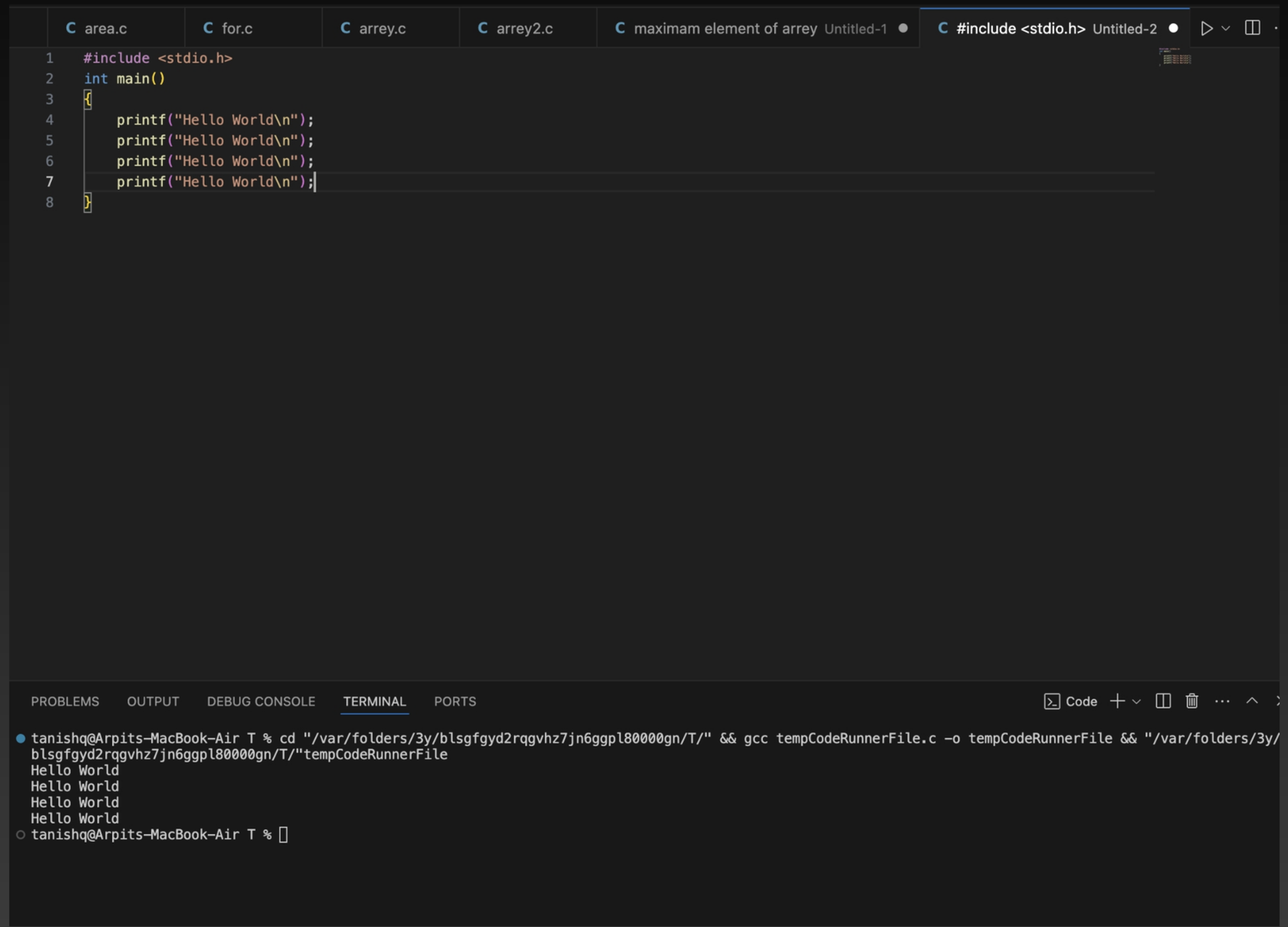
Step 4: Prompt for address input

Step 5: Print address in multiple lines

Step 6: End

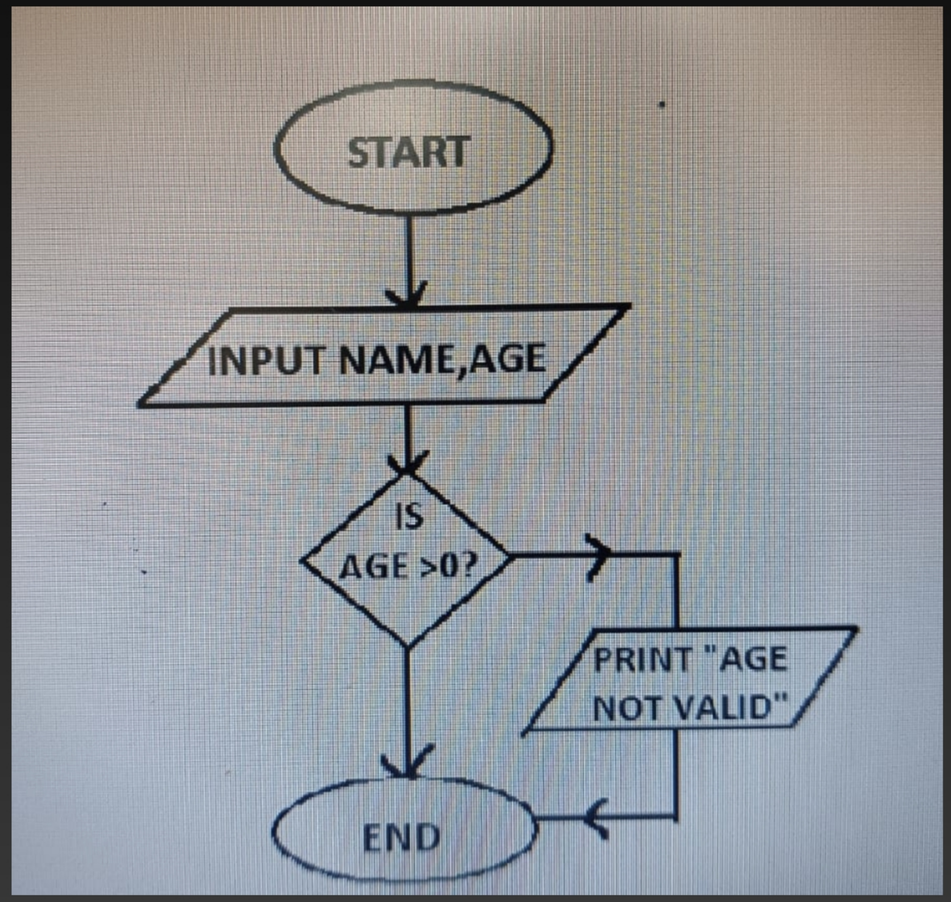
Printing the address in multiple lines

Syntax



That prompts the user to enter their name and age.

Flowchart



That prompts the user to enter their name and age.

Algorithm

Step 1: Start

Step 2: Declare variables

Step 3: Prompt for name

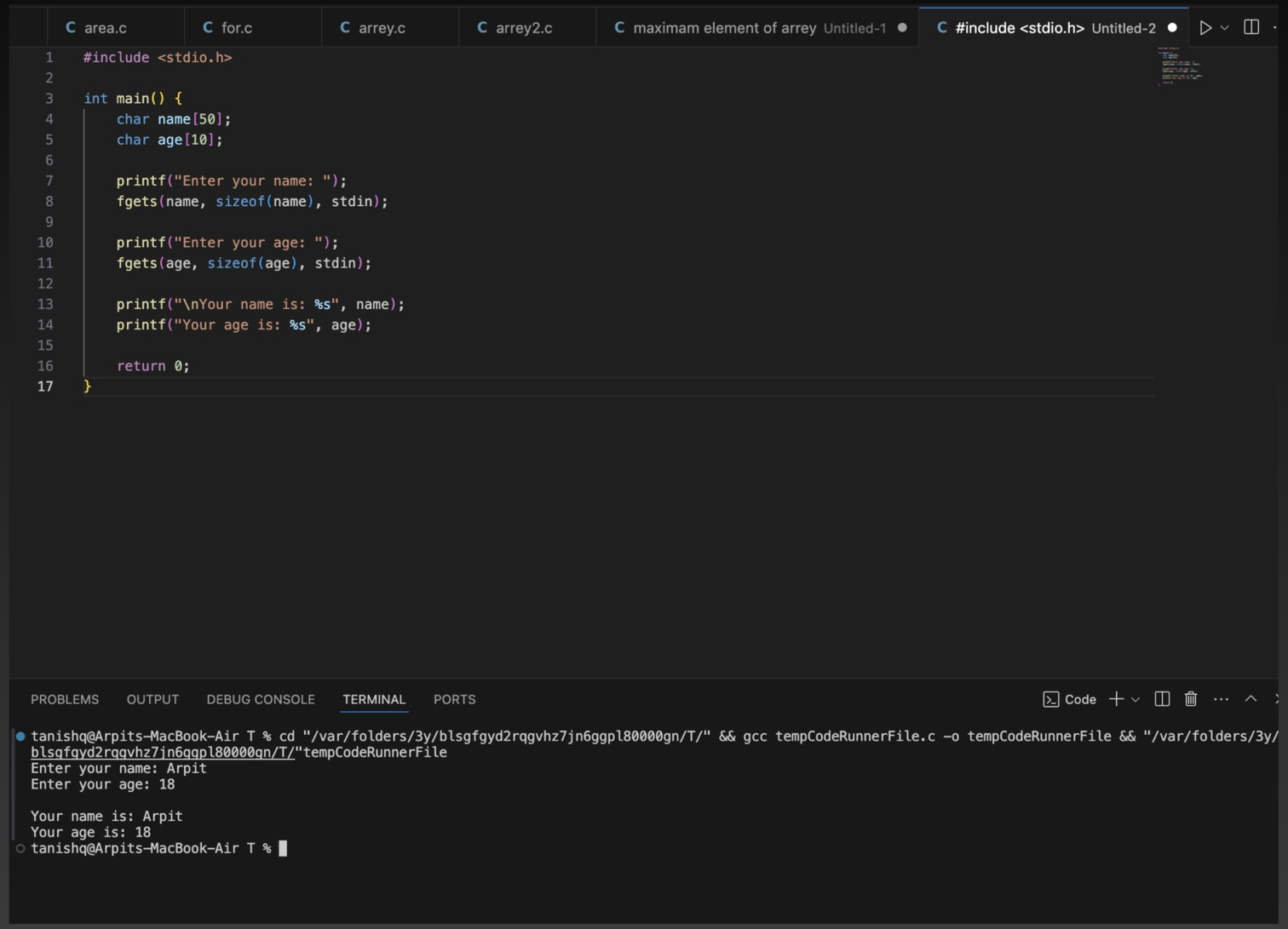
Step 4: Prompt for age

Step 5: Print the entered information

Step 6: End

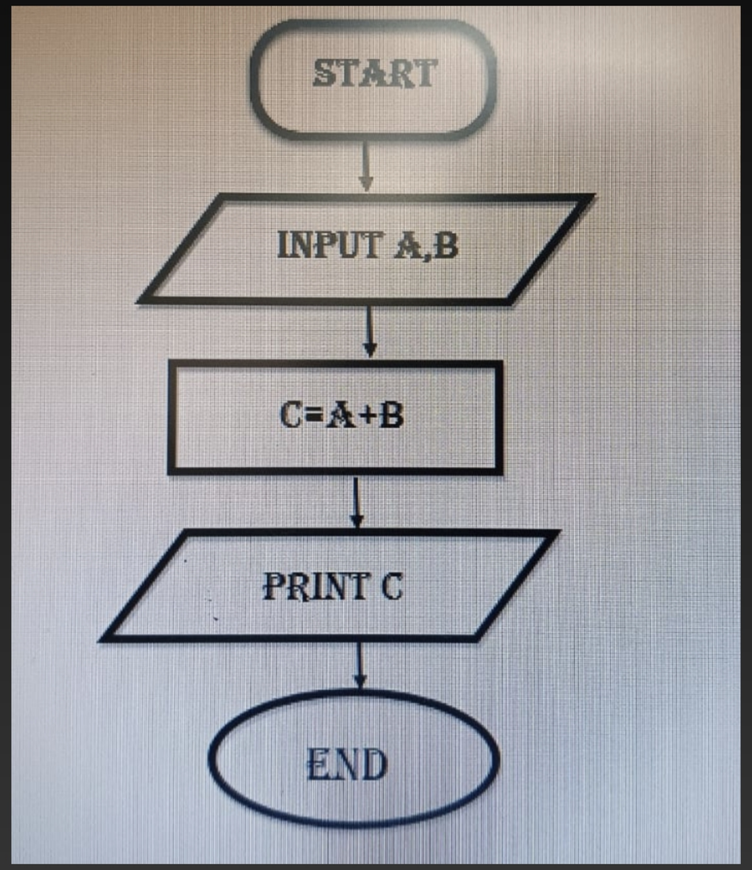
That prompts the user to enter their name and age.

Syntax



Add two numbers, take number from user.

Flowchart



Add two numbers, take number from user.

Algorithm

Step 1: Start

Step 2: Declare variables

Step 3: Prompt for the fi rst number

Step 4: Prompt for the second number

Step 5: Calculate the sum

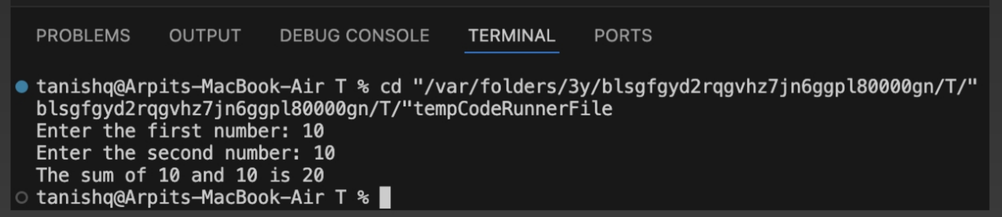
Step 6: Print the sum

Step 7: End

Add two numbers, take number from user.

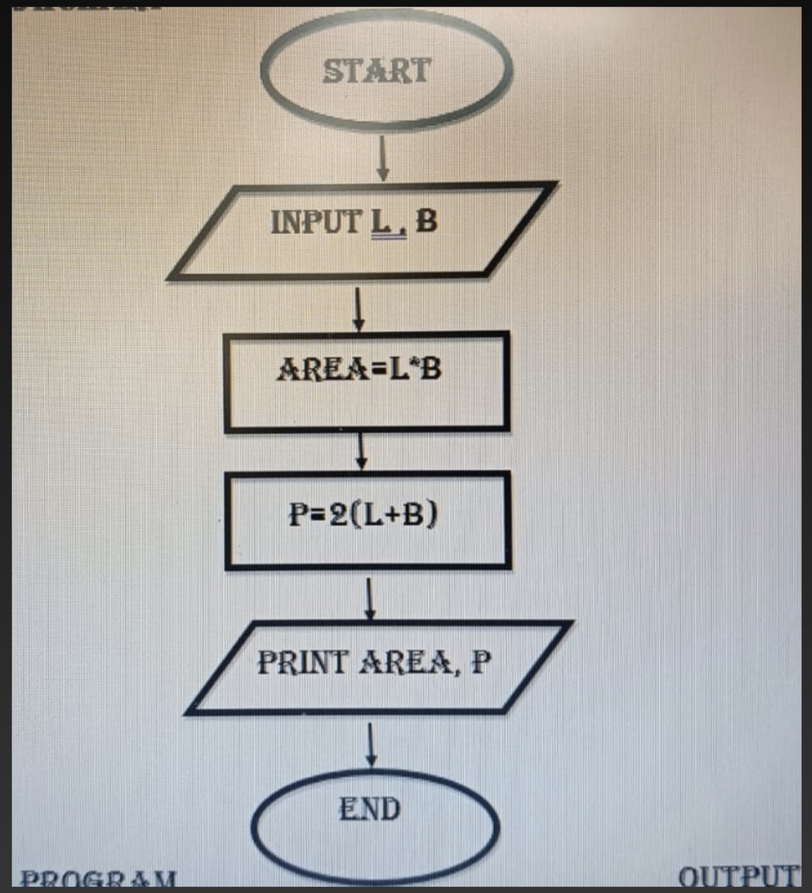
Syntax

```c
#include <stdio.h>

int main() {
    int num1, num2, sum;
    printf("Enter the first number: ");
    scanf("%d", &num1);

    printf("Enter the second number: ");
    scanf("%d", &num2);
    sum = num1 + num2;
    printf("The sum of %d and %d is %d\n", num1, num2, sum);

    return 0;
}
```



Calculate the area and perimeter of a rectangle based on the input length and width.

Flowchart



Calculate the area and perimeter of a rectangle based on the input length and width.

Algorithm

Step 1: Start

Step 2: Declare variables

Step 4: Calculate area and perimeter

Step 5: Print the results

Step 6: End

Calculate the area and perimeter of a rectangle based on the input length and width.

Syntax

```c
#include <stdio.h>

int main() {
    int length, width, area, perimeter;

    printf("Enter the length of the rectangle: ");
    scanf("%d", &length);
    printf("Enter the width of the rectangle: ");
    scanf("%d", &width);
    area = length * width;
    perimeter = 2 * (length + width);
    printf("Area of the rectangle = %d square units\n", area);
    printf("Perimeter of the rectangle = %d units\n", perimeter);

    return 0;
}
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
tanishq@Arpits-MacBook-Air T % cd "/var/folders/3y/blsgfgyd2rqgvhz7jn6ggpl80000gn/T/" && gcc tempCodeRunnerF
blsgfgyd2rqgvhz7jn6ggpl80000gn/T/"tempCodeRunnerFile
Enter the length of the rectangle: 10
Enter the width of the rectangle: 20
Area of the rectangle = 200 square units
Perimeter of the rectangle = 60 units
tanishq@Arpits-MacBook-Air T %
```

convert temperature from Celsius to Fahrenheit using the formula : F= (C*9/5)+32

Algorithm

Step 1: Start

Step 2: Declare variablesStep 3: Prompt for input

Step 4: Convert Celsius to Fahrenheit

· Use the formula Fahrenheit = (celsius * 9/5) + 32 to calculate the temperature in Fahrenheit.

Step 5: Print the result

Step 6: End

convert temperature from Celsius to Fahrenheit using the formula : F= (C*9/5)+32

Syntax

```c
#include <stdio.h>

int main() {
    float celsius, fahrenheit;
    printf("Enter temperature in Celsius: ");
    scanf("%f", &celsius);
    fahrenheit = (celsius * 9 / 5) + 32;
    printf("%.2f Celsius is equal to %.2f Fahrenheit\n", celsius, fahrenheit)

    return 0;
}
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

tanishq@Arpits-MacBook-Air T % cd "/var/folders/3y/blsgfgyd2rqgvhz7jn6ggpl80000gn/T/"
blsgfgyd2rqgvhz7jn6ggpl80000gn/T/"tempCodeRunnerFile
Enter temperature in Celsius: 36
36.00 Celsius is equal to 96.80 Fahrenheit
tanishq@Arpits-MacBook-Air T %
```

Write a C program to apply bitwise OR, AND, and NOT operators on bit level.

Algorithm

1. Start
2. Include necessary header
3. Defi ne the function
4. Perform bitwise OR
5. Print bitwise OR result
6. Perform bitwise AND
7. Print bitwise AND result
8. Perform bitwise NOT on num1
9. Print bitwise NOT result on num1
10. Perform bitwise NOT on num2
11. Print bitwise NOT result on num2
12. End

Write a C program to apply bitwise OR, AND, and NOT operators on bit level.

Syntax

iu.c > main()

```c
#include <stdio.h>

void bitwiseOperations(int num1, int num2) {
    int orResult = num1 | num2;
    printf("Bitwise OR: %d\n", orResult);
    int andResult = num1 & num2;
    printf("Bitwise AND: %d\n", andResult);
    int notResult1 = ~num1;
    printf("Bitwise NOT on num1: %d\n", notResult1);
    int notResult2 = ~num2;
    printf("Bitwise NOT on num2: %d\n", notResult2);
}

int main() {
    int num1, num2;

    printf("Enter the first number: ");
    scanf("%d", &num1);

    printf("Enter the second number: ");
    scanf("%d", &num2);

    bitwiseOperations(num1, num2);

    return 0;
}
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
cd "/Users/tanishq/c totorials/" && gcc iu.c -o iu && "/Users/tanishq/c totorials/"iu
tanishq@Arpits-MacBook-Air c totorials % cd "/Users/tanishq/c totorials/" && gcc iu.c -o iu && "/Users/tanishq/c totorials/"iu
Enter the first number: 2
Enter the second number: 3
Bitwise OR: 3
Bitwise AND: 2
Bitwise NOT on num1: -3
Bitwise NOT on num2: -4
tanishq@Arpits-MacBook-Air c totorials %
```

Write a C program to apply left and right shift operators.

Algorithm

Step1-Start.

Step2-Initialize an integer variable with a value.

Step3-Apply the left shift operator (<<) on the variable and store the result.

Step4-Apply the right shift operator (>>) on the variable and store the result.

Step5-Print the original value, the result of the left shift, and the result of the right shift.

Step6-End.

Write a C program to apply left and right shift operators.

Syntax

iu.c > main()

```c
#include <stdio.h>

int main() {
    int num = 4;
    int leftShift, rightShift;

    leftShift = num << 1;
    rightShift = num >> 1;

    printf("Original number: %d\n", num);
    printf("After left shift by 1: %d\n", leftShift);
    printf("After right shift by 1: %d\n", rightShift);

    return 0;
}
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
tanishq@Arpits-MacBook-Air c totorials % cd "/Users/tanishq/c totorials/" && gcc iu.c -o iu && "/Users/tanishq/c totorials/"iu
Original number: 4
After left shift by 1: 8
After right shift by 1: 2
tanishq@Arpits-MacBook-Air c totorials %
```

WAP to read a list of integers and store it in a 1D array. Write a C program to print the second largest integer in a list of integers.

Algorithm

Step1-Start.
Step2-Input the number of elements, n.
Step3-Create an array of size n.
Step4-Read the n elements into the array.
Step5-Initialize two variables, f irstLargest and secondLargest.
Step6-Compare the f irst two elements of the array:
Step7-If the f irst element is greater than the second element:
Step8-Set f irstLargest to the f irst element.
Step9-Set secondLargest to the second element.
Step10-Set f irstLargest to the second element.
Step11-Set secondLargest to the f irst element.
Step12-Iterate through the remaining elements in the array from the third element onward:
Step13-If the current element is greater than f irstLargest:
Step14-Set secondLargest to f irstLargest.
Step15-Set f irstLargest to the current element.
Step16-Else if the current element is greater than secondLargest and not equal to f irstLargest:
Step17-Set secondLargest to the current element.
Step18-Output the value of secondLargest.
Step19-End.

WAP to read a list of integers and store it in a 1D array. Write a C program to print the second largest integer in a list of integers.

Syntax

iu.c > main()

```c
#include <stdio.h>

int main() {
    int n, i;

    printf("Enter the number of elements: ");
    scanf("%d", &n);

    int arr[n];
    printf("Enter the elements:\n");
    for (i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }

    int firstLargest, secondLargest;

    if (arr[0] > arr[1]) {
        firstLargest = arr[0];
        secondLargest = arr[1];
    } else {
        firstLargest = arr[1];
        secondLargest = arr[0];
    }

    for (i = 2; i < n; i++) {
        if (arr[i] > firstLargest) {
            secondLargest = firstLargest;
            firstLargest = arr[i];
        } else if (arr[i] > secondLargest && arr[i] != firstLargest) {
            secondLargest = arr[i];
        }
    }

    printf("The second largest element is: %d\n", secondLargest);

    return 0;
}
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
tanishq@Arpits-MacBook-Air c totorials % cd "/Users/tanishq/c totorials/" && gcc iu.c -o iu && "/Users/tanishq/c totorials/"iu
Enter the number of elements: 2
Enter the elements:
3
4
```

WAP to read a list of integers and store it in a 1D array. Write a C program to count and diaplay positive, negative, odd, and even numbers in an array.

Algorithm

1.Start.
2.Input the number of elements, n.
3.Create an array of size n.
4.Read the n elements into the array.
5.Initialize counters for positive, negative, odd, and even numbers to 0.
6.Iterate through each element of the array:
7.If the element is positive, increment the positive counter.
8.If the element is negative, increment the negative counter.
9.If the element is odd, increment the odd counter.
10.If the element is even, increment the even counter.
11.Output the values of positive, negative, odd, and even counters.
12.End.

Write a C program to apply bitwise OR, AND, and NOT operators on bit level.

Syntax

iu.c > main()

```c
#include <stdio.h>

int main() {
    int n, i, positive = 0, negative = 0, odd = 0, even = 0;

    printf("Enter the number of elements: ");
    scanf("%d", &n);

    int arr[n];
    printf("Enter the elements:\n");
    for (i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }

    for (i = 0; i < n; i++) {
        if (arr[i] > 0) {
            positive++;
        } else if (arr[i] < 0) {
            negative++;
        }

        if (arr[i] % 2 == 0) {
            even++;
        } else {
            odd++;
        }
    }

    printf("Positive numbers: %d\n", positive);
    printf("Negative numbers: %d\n", negative);
    printf("Odd numbers: %d\n", odd);
    printf("Even numbers: %d\n", even);
    }
```

PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
tanishq@Arpits-MacBook-Air c totorials % cd "/Users/tanishq/c totorials/" && gcc iu.c -o iu && "/Users/tanishq/c totorials/"iu
Enter the number of elements: 2
Enter the elements:
4
5
Positive numbers: 2
Negative numbers: 0
Odd numbers: 1
Even numbers: 1
tanishq@Arpits-MacBook-Air c totorials %
```

WAP to read a list of integers and store it in a 1D array. Write a C program to find the frequency of a particular number in a list of integers.

Algorithm

1.Start.

2.Input the number of elements, n.

3.Create an array of size n.

4.Read the n elements into the array.

5.Input the number whose frequency needs to be found, searchNum.

6.Initialize a counter count to 0.

7.Iterate through the array:

8.If the current element is equal to searchNum, increment count.

9.Output the value of count.

10.End.

WAP to read a list of integers and store it in a 1D array. Write a C program to find the frequency of a particular number in a list of integers.

Syntax

iu.c > ...

```c
#include <stdio.h>

int main() {
    int n, i, searchNum, count = 0;

    printf("Enter the number of elements: ");
    scanf("%d", &n);

    int arr[n];
    printf("Enter the elements:\n");
    for (i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }

    printf("Enter the number to find the frequency of: ");
    scanf("%d", &searchNum);

    for (i = 0; i < n; i++) {
        if (arr[i] == searchNum) {
            count++;
        }
    }

    printf("The frequency of %d is: %d\n", searchNum, count);

    return 0;
}
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
tanishq@Arpits-MacBook-Air c totorials % cd "/Users/tanishq/c totorials/" && gcc iu.c -o iu && "/Users/tanishq/c totorials/"iu
Enter the number of elements: 1
Enter the elements:
2
Enter the number to find the frequency of: 3
The frequency of 3 is: 0
tanishq@Arpits-MacBook-Air c totorials %
```

WAP that reads two matrices A(m×n) and B(pxq) and computes the product of A and B. Read matrix A and matrix B in row major order reapectively. Print both the input matrices and resultant matrix with suitable headings and output should be in matrix format only. Program must check the compatibility of orders of the matrices for multiplication. Report appropriate message in case of incompatibility.

Algorithm

1.Start.

2.Input the dimensions of matrix A (m x n) and matrix B (p x q).

3.Check if matrices are compatible for multiplication (i.e., n should be equal to p).

4.If not compatible, output an appropriate message and end.

5.Create matrix A of size m x n and matrix B of size p x q.

6.Read matrix A and matrix B in row major order.

7.Initialize the resultant matrix C of size m x q to zero.

8.Compute the product of matrix A and B:

9.For each element in the resultant matrix C:

10.Calculate the sum of the products of corresponding elements from row of A and column

11.Output matrix A, matrix B, and the resultant matrix C with suitable headings.

12.End.

WAP that reads two matrices A(m×n) and B(pxq) and computes the product of A and B. Read matrix A and matrix B in row major order reapectively. Print both the input matrices and resultant matrix with suitable headings and output should be in matrix format only. Program must check the compatibility of orders of the matrices for multiplication. Report appropriate message in case of incompatibility.

Syntax

iu.c > main()

```c
#include <stdio.h>

int main() {
    int m, n, p, q, i, j, k;
    printf("Enter rows and columns of matrix A: ");
    scanf("%d %d", &m, &n);
    printf("Enter rows and columns of matrix B: ");
    scanf("%d %d", &p, &q);
    if (n != p) {
        printf("Matrices are not compatible for multiplication.\n");
        return 0;
    }

    int A[m][n], B[p][q], C[m][q];
    printf("Enter elements of matrix A:\n");
    for (i = 0; i < m; i++) {
        for (j = 0; j < n; j++) {
            scanf("%d", &A[i][j]);
        }
    }
    printf("Enter elements of matrix B:\n");
    for (i = 0; i < p; i++) {
        for (j = 0; j < q; j++) {
            scanf("%d", &B[i][j]);
        }
    }
    for (i = 0; i < m; i++) {
        for (j = 0; j < q; j++) {
            C[i][j] = 0;
        }
    }
    for (i = 0; i < m; i++) {
        for (j = 0; j < q; j++) {
            for (k = 0; k < n; k++) {
                C[i][j] += A[i][k] * B[k][j];
            }
        }
    }
    printf("Matrix A:\n");
    for (i = 0; i < m; i++) {
        for (j = 0; j < n; j++) {
            printf("%d ", A[i][i]);
        }
        printf("\n");
    }
    printf("Matrix B:\n");
    for (i = 0; i < p; i++) {
        for (j = 0; j < q; j++) {
            printf("%d ", B[i][j]);
        }
        printf("\n");
    }
    printf("Resultant Matrix C (Product of A and B):\n");
    for (i = 0; i < m; i++) {
        for (j = 0; j < q; j++) {
            printf("%d ", C[i][j]);
        }
        printf("\n");
    }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
tanishq@Arpits-MacBook-Air c totorials % cd "/Users/tanishq/c totorials/" && gcc iu.c -o iu && "/Users/tanishq/c totorials/"iu
Enter rows and columns of matrix A: 2
2
Enter rows and columns of matrix B: 2
2
Enter elements of matrix A:
1
1
1
1
Enter elements of matrix B:
1
1
1
1
Matrix A:
1 1
1 1
Matrix B:
1 1
1 1
Resultant Matrix C (Product of A and B):
2 2
2 2
```

Develop a C function ISPRIME(num) that accepts an integer argument and returns 1 if the argument is prime, a 0 otherwise. Write a C program that invokes this function to generate prime numbers between the given ranges.

Algorithm

1.Start.

2.Def ine a function ISPRIME(num):

3.If num is less than 2, return 0.

4.For each integer i from 2 to the square root of num:

5.If num is divisible by i, return 0.

6.If no divisors are found, return 1.

7.Input the lower and upper bounds of the range.

8.Iterate through each number in the range:

9.Use ISPRIME(num) to check if the number is prime.

10.If the number is prime, print it.

11.End.

Develop a C function ISPRIME(num) that accepts an integer argument and returns 1 if the argument is prime, a 0 otherwise. Write a C program that invokes this function to generate prime numbers between the given ranges.

Syntax

iu.c > ...

```c
#include <stdio.h>
#include <math.h>
int ISPRIME(int num) {
    if (num < 2) {
        return 0;
    }
    for (int i = 2; i <= sqrt(num); i++) {
        if (num % i == 0) {
            return 0;
        }
    }
    return 1;
}
int main() {
    int lower, upper;
    printf("Enter the lower bound of the range: ");
    scanf("%d", &lower);
    printf("Enter the upper bound of the range: ");
    scanf("%d", &upper);

    printf("Prime numbers between %d and %d are:\n", lower, upper);
    for (int num = lower; num <= upper; num++) {
        if (ISPRIME(num)) {
            printf("%d ", num);
        }
    }
    printf("\n");

    return 0;
}
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
tanishq@Arpits-MacBook-Air c totorials % cd "/Users/tanishq/c totorials/" && gcc iu.c -o iu && "/Users/tanishq/c totorials/"iu
Enter the lower bound of the range: 2
Enter the upper bound of the range: 3
Prime numbers between 2 and 3 are:
2 3
tanishq@Arpits-MacBook-Air c totorials %
```

Develop a function REVERSE(str) that accepts a string argument. Write a C program that invokes this function to find the reverse of a given string.

Algorithm

1.Start.

2.Def ine the REVERSE(str) function:

3.Initialize two indexes, start at 0 and end at the length of the string minus 1.

4.Swap the characters at start and end.

5.Increment start and decrement end.

6.Repeat until start is greater than or equal to end.

7.Input the string to be reversed.

8.Call the REVERSE(str) function with the input string.

9.Output the reversed string.

10.End.

Develop a recursive and non-recursive function FACT(num) to find the factorial of a number, n!, defined by FACT(n) = 1, if n=0. Otherwise, FACT(n) = n*FACT(n-1). Using the function, write a C program to compute the binomial coefficient. Tabulate the results for different of n and r with suitable messages.

Algorithm

1.Start.
2.If num is 0, return 1.
*3.Otherwise, return num * FACT(num - 1).*
4.End.
5.Algorithm for Non-Recursive FACT(num):
6.Start.
7.Initialize result to 1.
8.For each integer i from 1 to num, multiply result by i.
9.Return result.
10.End.
11.C Program to Compute Binomial Coeff icient:
12.Def ine the recursive and non-recursive FACT functions.
*13.Def ine a function BINOMIAL_COEFF(n, r) that calculates FACT(n) / (FACT(r) * FACT(n - r)).*
14.Input values for n and r.
15.Compute the binomial coeff icient using BINOMIAL_COEFF.
16.Output the result with suitable messages.Tabulate the results for di fferent values of n and r.
17.End.

Develop a recursive and non-recursive function FACT(num) to find the factorial of a number, n!, defined by FACT(n) = 1, if n=0. Otherwise, FACT(n) = n*FACT(n-1). Using the function, write a C program to compute the binomial coefficient. Tabulate the results for different of n and r with suitable messages.

Syntax

iu.c > main()

```c
#include <stdio.h>
int FACT_recursive(int num) {
    if (num == 0) {
        return 1;
    } else {
        return num * FACT_recursive(num - 1);
    }
}
int FACT_non_recursive(int num) {
    int result = 1;
    for (int i = 1; i <= num; i++) {
        result *= i;
    }
    return result;
}
int BINOMIAL_COEFF(int n, int r) {
    return FACT_non_recursive(n) / (FACT_non_recursive(r) * FACT_non_recursive(n - r));
}

int main() {
    int n, r;

    printf("Enter value of n: ");
    scanf("%d", &n);
    printf("Enter value of r: ");
    scanf("%d", &r);

    if (r > n) {
        printf("Invalid input: r should be less than or equal to n.\n");
        return 1;
    }

    int binomialCoefficient = BINOMIAL_COEFF(n, r);

    printf("Binomial Coefficient C(%d, %d) = %d\n", n, r, binomialCoefficient);
    printf("\nTabulated Results:\n");
    printf(" n  r  C(n, r)\n");
    printf("-------------\n");
    for (int i = 0; i <= n; i++) {
        for (int j = 0; j <= i; j++) {
            printf("%2d %2d %5d\n", i, j, BINOMIAL_COEFF(i, j));
        }
    }

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
tanishq@Arpits-MacBook-Air c totorials % cd "/Users/tanishq/c totorials/" && gcc iu.c -o iu && "/Users/tanishq/c totorials/"iu
Enter value of n: 2
Enter value of r: 2
Binomial Coefficient C(2, 2) = 1

Tabulated Results:
 n  r  C(n, r)
-------------
 0  0     1
 1  0     1
 1  1     1
 2  0     1
 2  1     2
 2  2     1
tanishq@Arpits-MacBook-Air c totorials %
```

Develop a recursive function GCD(num1, num2) that accepts two integer arguments. Write a C program that invokes this function to find the greatest common divisor of two given integers.

Algorithm

1.Start.

2.If num2 is 0, return num1.

3.Otherwise, return GCD(num2, num1 % num2).

4.End.

Develop a recursive function GCD(num1, num2) that accepts two integer arguments. Write a C program that invokes this function to find the greatest common divisor of two given integers.

Syntax

iu.c > ...

```c
#include <stdio.h>
int GCD(int num1, int num2) {
    if (num2 == 0) {
        return num1;
    } else {
        return GCD(num2, num1 % num2);
    }
}
int main() {
    int num1, num2;

    printf("Enter two integers: ");
    scanf("%d %d", &num1, &num2);

    int gcd = GCD(num1, num2);

    printf("The GCD of %d and %d is: %d\n", num1, num2, gcd);

    return 0;
}
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
tanishq@Arpits-MacBook-Air c totorials % cd "/Users/tanishq/c totorials/" && gcc iu.c -o iu && "/Users/tanishq/c totorials/"iu
Enter two integers: 2
3
The GCD of 2 and 3 is: 1
tanishq@Arpits-MacBook-Air c totorials %
```

Develop a recursive function FIBO(num) that accepts an integer argument. Write a C program that invokes this function to generate the Fibonacci sequence up to num.

Algorithm

1.Start.

2.If num is 0, return 0.

3.If num is 1, return 1.

4.Otherwise, return FIBO(num - 1) + FIBO(num - 2).

5.End.

Develop a recursive function FIBO(num) that accepts an integer argument. Write a C program that invokes this function to generate the Fibonacci sequence up to num.

Syntax

iu.c > ...

```c
#include <stdio.h>
int FIBO(int num) {
    if (num == 0) {
        return 0;
    } else if (num == 1) {
        return 1;
    } else {
        return FIBO(num - 1) + FIBO(num - 2);
    }
}
int main() {
    int num, i;

    printf("Enter an integer: ");
    scanf("%d", &num);

    printf("Fibonacci sequence up to %d:\n", num);
    for (i = 0; i <= num; i++) {
        printf("%d ", FIBO(i));
    }
    printf("\n");

    return 0;
}
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
tanishq@Arpits-MacBook-Air c totorials % cd "/Users/tanishq/c totorials/" && gcc iu.c -o iu && "/Users/tanishq/c totorials/"iu
Enter an integer: 2
Fibonacci sequence up to 2:
0 1 1
tanishq@Arpits-MacBook-Air c totorials %
```

Create a union containing 6 strings: name, home_address, hostel_address, city, state and zip. Write a C program to display your present address.

Algorithm

1.Start.

2.Def ine a union Address containing the following strings:

3.name

4.home_address

5.hostel_address

6.city

7.state

8.zip

9.Declare a variable addr of type union Address.

10.Store the name in addr.name.

11.Store the home address in addr.home_address.

12.Store the city in addr.city.

13.Store the state in addr.state.

14.Store the ZIP code in addr.zip.

15.Print the present address information:

16.Name

17.Home Address

18.City

19.State

20.ZIP

22.End.

Create a union containing 6 strings: name, home_address, hostel_address, city, state and zip. Write a C program to display your present address.

Syntax

iu.c > ...

```c
#include <stdio.h>
#include <string.h>
union Address {
    char name[50];
    char home_address[100];
    char hostel_address[100];
    char city[50];
    char state[50];
    char zip[10];
};

int main() {
    union Address addr;
    strcpy(addr.name, "John Doe");
    strcpy(addr.home_address, "123 Home St");
    strcpy(addr.city, "Springfield");
    strcpy(addr.state, "IL");
    strcpy(addr.zip, "62701");
    printf("Present Address:\n");
    printf("Name: %s\n", addr.name);
    printf("Home Address: %s\n", addr.home_address);
    printf("City: %s\n", addr.city);
    printf("State: %s\n", addr.state);
    printf("ZIP: %s\n", addr.zip);

    return 0;
}
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
tanishq@Arpits-MacBook-Air c totorials % cd "/Users/tanishq/c totorials/" && gcc iu.c -o iu && "/Users/tanishq/c totorials/"iu
Present Address:
Name: 62701
Home Address: 62701
City: 62701
State: 62701
ZIP: 62701
tanishq@Arpits-MacBook-Air c totorials %
```

Create a union containing 6 strings: name, home_address, hostel_address, city, state and zip. Write a C program to display your present address.

```c
#include <stdio.h>
#include <string.h>
union Address {
    char name[50];
    char home_address[100];
    char hostel_address[100];
    char city[30];
    char state[30];
    char zip[10];
};

int main() {
    union Address addr;
    strcpy(addr.hostel_address, "Hostel No. 4, upes");

    // Display the present address
    printf("Present Address:\n");
    printf("%s\n", addr.hostel_address);

    return 0;
}
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
tanishq@Arpits-MacBook-Air c totorials % cd "/Users/tanishq/c totorials/" && gcc m.c -o m && "/Users/tanishq/c totorials/"m
Present Address:
Hostel No. 4, upes
tanishq@Arpits-MacBook-Air c totorials %
```

Write a program to define some constant variable in preprocessor.

```c
#include <stdio.h>
#define PI 3.14159
#define MAX_LENGTH 100
#define MIN_AGE 18
#define GRAVITY 9.8
#define MESSAGE "Welcome to C Programming!"

int main() {
    printf("The value of PI is: %.5f\n", PI);
    printf("The maximum length allowed is: %d\n", MAX_LENGTH);
    printf("The minimum age to vote is: %d\n", MIN_AGE);
    printf("The value of gravity is: %.1f m/s^2\n", GRAVITY);
    printf("%s\n", MESSAGE);

    return 0;
}

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
tanishq@Arpits-MacBook-Air c totorials % cd "/Users/tanishq/c totorials/" && gcc m.c -o m && "/Users/tanishq/c totorials/"m
The value of PI is: 3.14159
The maximum length allowed is: 100
The minimum age to vote is: 18
The value of gravity is: 9.8 m/s^2
Welcome to C Programming!
tanishq@Arpits-MacBook-Air c totorials %
```

Write a program to define a function in directives.

```c
#include <stdio.h>

#define SQUARE(x) ((x) * (x))
#define MAX(a, b) ((a) > (b) ? (a) : (b))

int main() {
    int num1 = 5, num2 = 10;
    printf("The square of %d is: %d\n", num1, SQUARE(num1));
    printf("The maximum of %d and %d is: %d\n", num1, num2, MAX(num1, num2));
    printf("The square of (3 + 2) is: %d\n", SQUARE(3 + 2));
    printf("The maximum of (7 + 2) and (5 * 2) is: %d\n", MAX(7 + 2, 5 * 2));

    return 0;
}
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
tanishq@Arpits-MacBook-Air c totorials % cd "/Users/tanishq/c totorials/" && gcc m.c -o m && "/Users/tanishq/c totorials/"m
The square of 5 is: 25
The maximum of 5 and 10 is: 10
The square of (3 + 2) is: 25
The maximum of (7 + 2) and (5 * 2) is: 10
tanishq@Arpits-MacBook-Air c totorials %
```

Declare different types of pointers (int, float, char) and initialize them with the addresses of variables. Print the values of both the pointers and the variables they point to.

```c
#include <stdio.h>

int main() {
    int int_var = 10;
    float float_var = 3.14;
    char char_var = 'A';
    int *int_ptr = &int_var;
    float *float_ptr = &float_var;
    char *char_ptr = &char_var;
    printf("Address of int_var: %p\n", (void*)int_ptr);
    printf("Address of float_var: %p\n", (void*)float_ptr);
    printf("Address of char_var: %p\n", (void*)char_ptr);
    printf("Value of int_var using pointer: %d\n", *int_ptr);
    printf("Value of float_var using pointer: %.2f\n", *float_ptr);
    printf("Value of char_var using pointer: %c\n", *char_ptr);

    return 0;
}

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
tanishq@Arpits-MacBook-Air c totorials % cd "/Users/tanishq/c totorials/" && gcc m.c -o m && "/Users/tanishq/c totorials/"m
Address of int_var: 0x16af03348
Address of float_var: 0x16af03344
Address of char_var: 0x16af03343
Value of int_var using pointer: 10
Value of float_var using pointer: 3.14
Value of char_var using pointer: A
tanishq@Arpits-MacBook-Air c totorials %
```

Perform pointer arithmetic (increment and decrement) on pointers of different data types. Observe how the memory addresses change and the effects on data access.

m.c > ...

```c
#include <stdio.h>
int main() {
    char charArray[3] = {'A', 'B', 'C'};
    int intArray[3] = {10, 20, 30};
    double doubleArray[3] = {1.1, 2.2, 3.3};
    char *charPtr = charArray;
    int *intPtr = intArray;
    double *doublePtr = doubleArray;
    printf("Initial addresses and values:\n");
    printf("charPtr: %p, value: %c\n", (void *)charPtr, *charPtr);
    printf("intPtr: %p, value: %d\n", (void *)intPtr, *intPtr);
    printf("doublePtr: %p, value: %f\n", (void *)doublePtr, *doublePtr);
    charPtr++;
    intPtr++;
    doublePtr++;
    printf("\nAfter incrementing:\n");
    printf("charPtr: %p, value: %c\n", (void *)charPtr, *charPtr);
    printf("intPtr: %p, value: %d\n", (void *)intPtr, *intPtr);
    printf("doublePtr: %p, value: %f\n", (void *)doublePtr, *doublePtr);
    charPtr--;
    intPtr--;
    doublePtr--;
    printf("\nAfter decrementing back:\n");
    printf("charPtr: %p, value: %c\n", (void *)charPtr, *charPtr);
    printf("intPtr: %p, value: %d\n", (void *)intPtr, *intPtr);
    printf("doublePtr: %p, value: %f\n", (void *)doublePtr, *doublePtr);

}
```

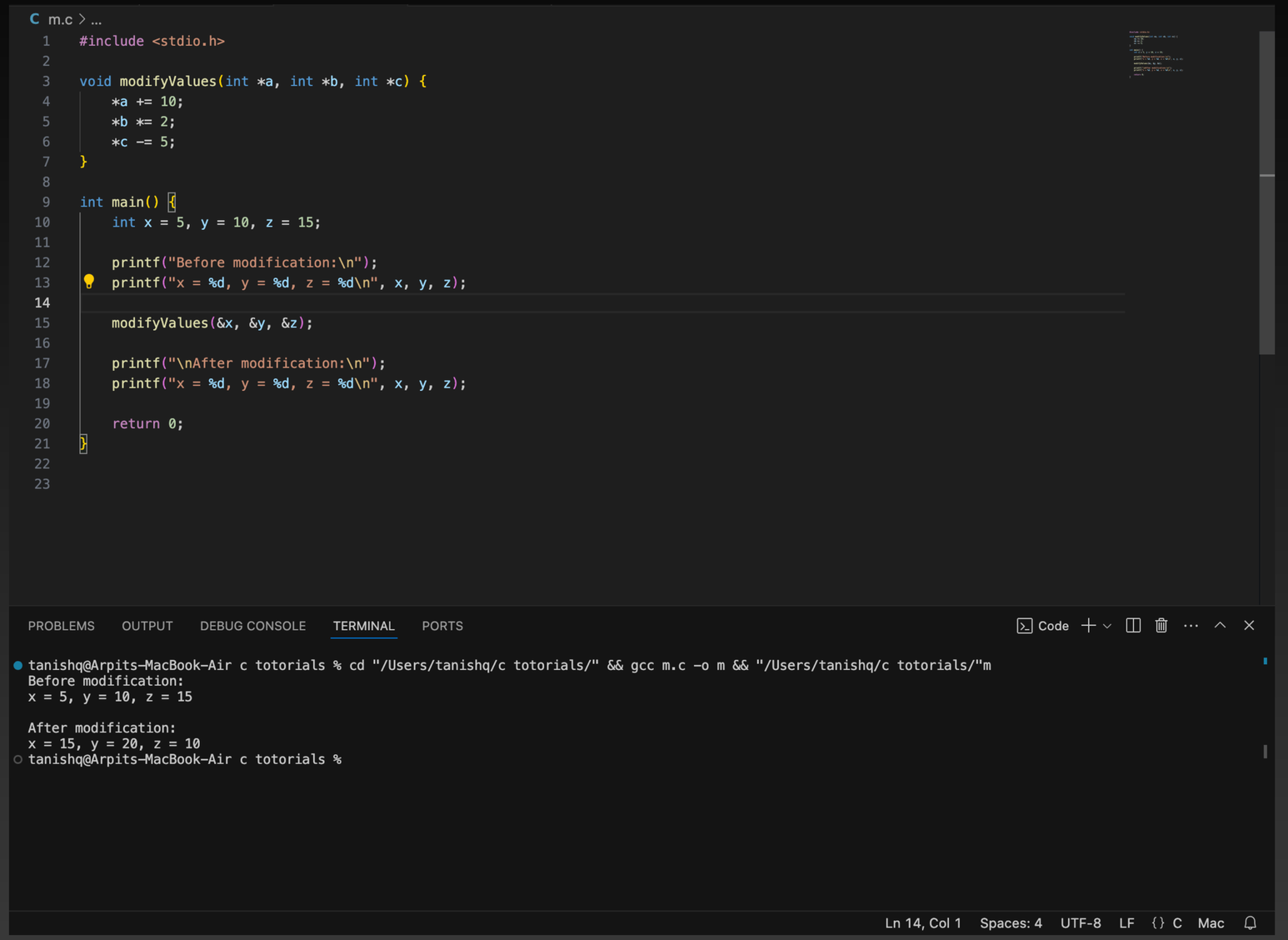
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Code

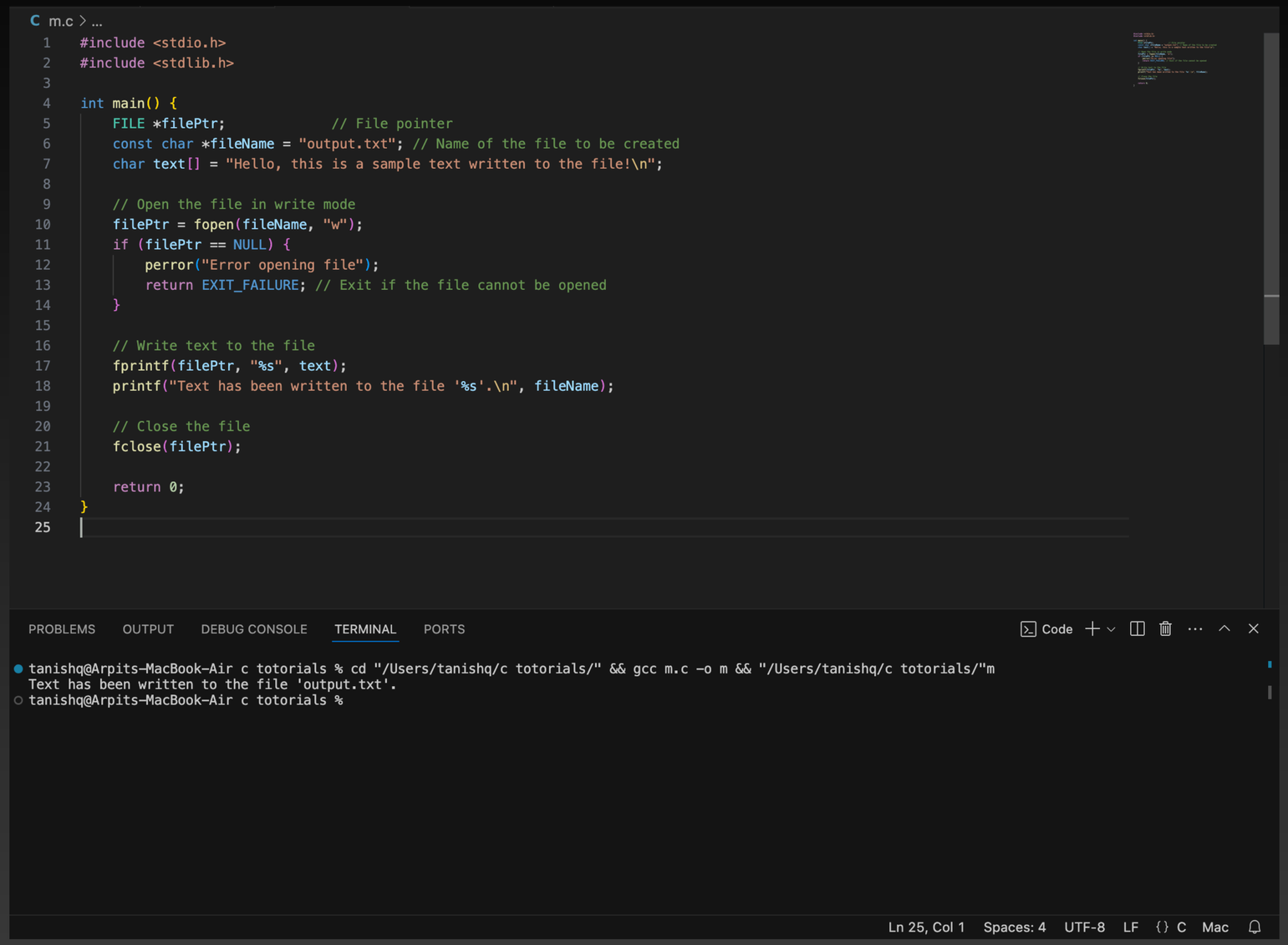
```
tanishq@Arpits-MacBook-Air c totorials % cd "/Users/tanishq/c totorials/" && gcc m.c -o m && "/Users/tanishq/c totorials/"m
Initial addresses and values:
charPtr: 0x16ce23318, value: A
intPtr: 0x16ce23338, value: 10
doublePtr: 0x16ce23320, value: 1.100000

After incrementing:
charPtr: 0x16ce23319, value: B
intPtr: 0x16ce2333c, value: 20
doublePtr: 0x16ce23328, value: 2.200000

After decrementing back:
charPtr: 0x16ce23318, value: A
intPtr: 0x16ce23338, value: 10
doublePtr: 0x16ce23320, value: 1.100000
tanishq@Arpits-MacBook-Air c totorials %
```

Ln 27, Col 1 Spaces: 4 UTF-8 LF {} C Mac

Write a function that accepts pointers as parameters. Pass variables by reference using pointers and modify their values within the function.



Write a program to create a new file and write text into it.



Open an existing file and read its content character by character, and then close the file.