**Reg. No: 21BCE1297 Name: Vidhi Shah Date: 16/03/22**

**PPS7**

**Q1**

**Aim:**

Write a function in ‘C’ to perform factorial of natural number ‘n’. Get the user input for ‘n’.

**Procedure:**

**Input:**

A natural number, n

**Output:**

Factorial of n

**Algorithm:**

Step 1: Declare ‘factorial’ function with return type ‘int’ and argument ‘int n’

**Main Function**

Step 1: Declare variables n and r

Step 2: Read a natural number ‘n’ from user

Step 3: Call ‘factorial’ function and save its return value in r

Step 4: Display ‘r’ which is the factorial of ‘n’

**Factorial Function (Recursion)**

Step 1: If n is equal to 0 or n is equal to 1

return 1

Step 2: Else call the function again in return statement

return n\*factorial(n-1)

**Code:**

#include <stdio.h>

// Function declaration

int factorial(int n);

// Main function

int main() {

    int n, r;

    printf("\nEnter a natural number: ");

    scanf("%d", &n);

    r = factorial(n);

    printf("Factorial of %d is %d\n\n", n, r);

    return 0;

}

// Factorial function

int factorial(int n) {

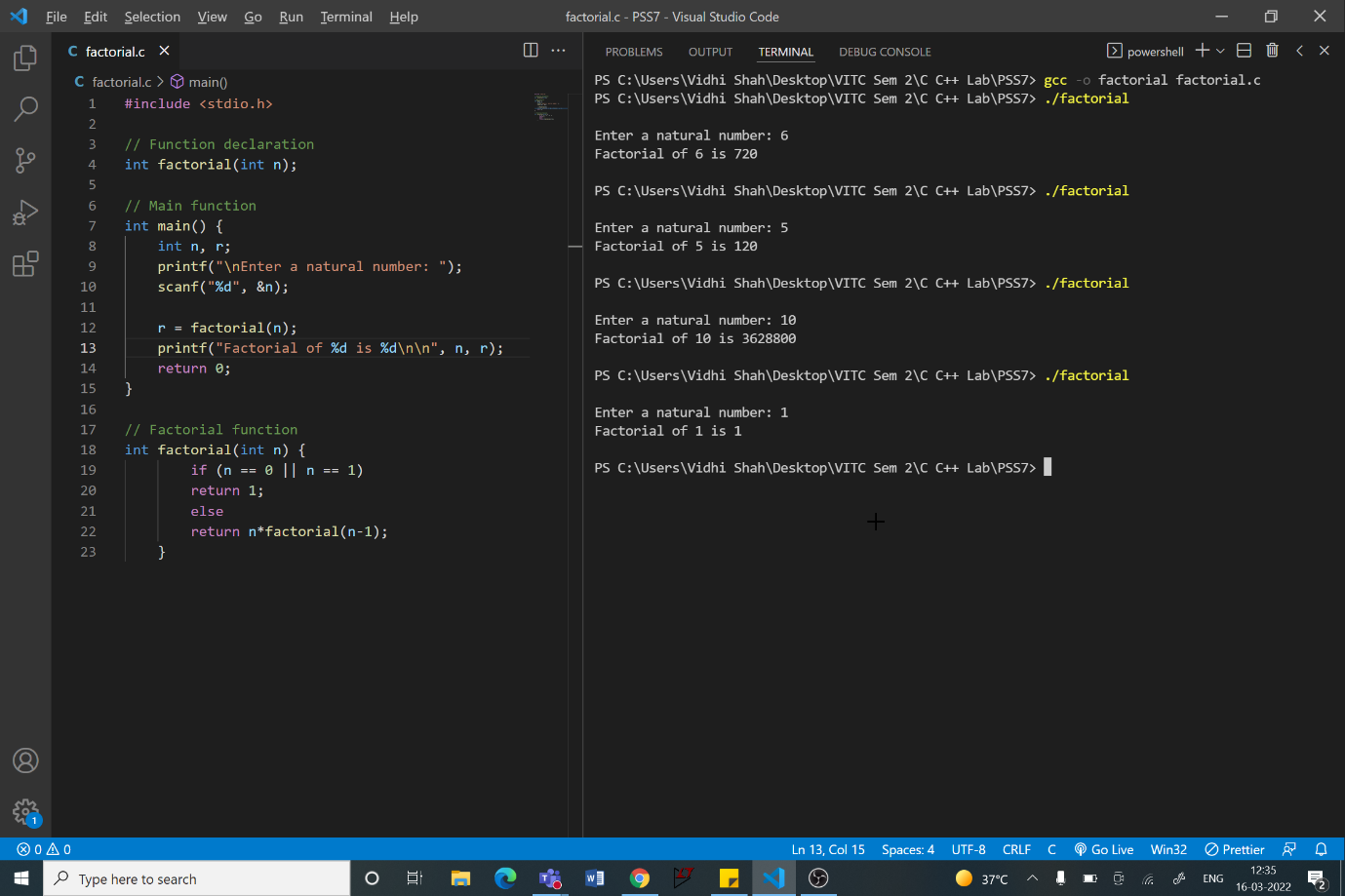
        if (n == 0 || n == 1)

        return 1;

        else

        return n\*factorial(n-1);

    }

****

**Q2**

**Aim:**

Write a function using ‘C’ to find the square of a number given by the user.

**Procedure:**

**Input:**

An integer number, n

**Output:**

Square of number n

**Algorithm:**

Step 1: Declare ‘square’ function with return type ‘int’ and argument ‘int n’

**Main Function**

Step 1: Declare variables n and r

Step 2: Read an integer number ‘n’ from user

Step 3: Call ‘square’ function and save its return value in r

Step 4: Display ‘r’ which is the square of ‘n’

**Square Function**

Step 1: Return n\*n

**Code:**

#include <stdio.h>

// Function declaration

int square(int n);

// Main function

int main() {

    int n, r;

    printf("\nEnter an integer: ");

    scanf("%d", &n);

    r = square(n);

    printf("Square of %d is %d\n\n", n, r);

    return 0;

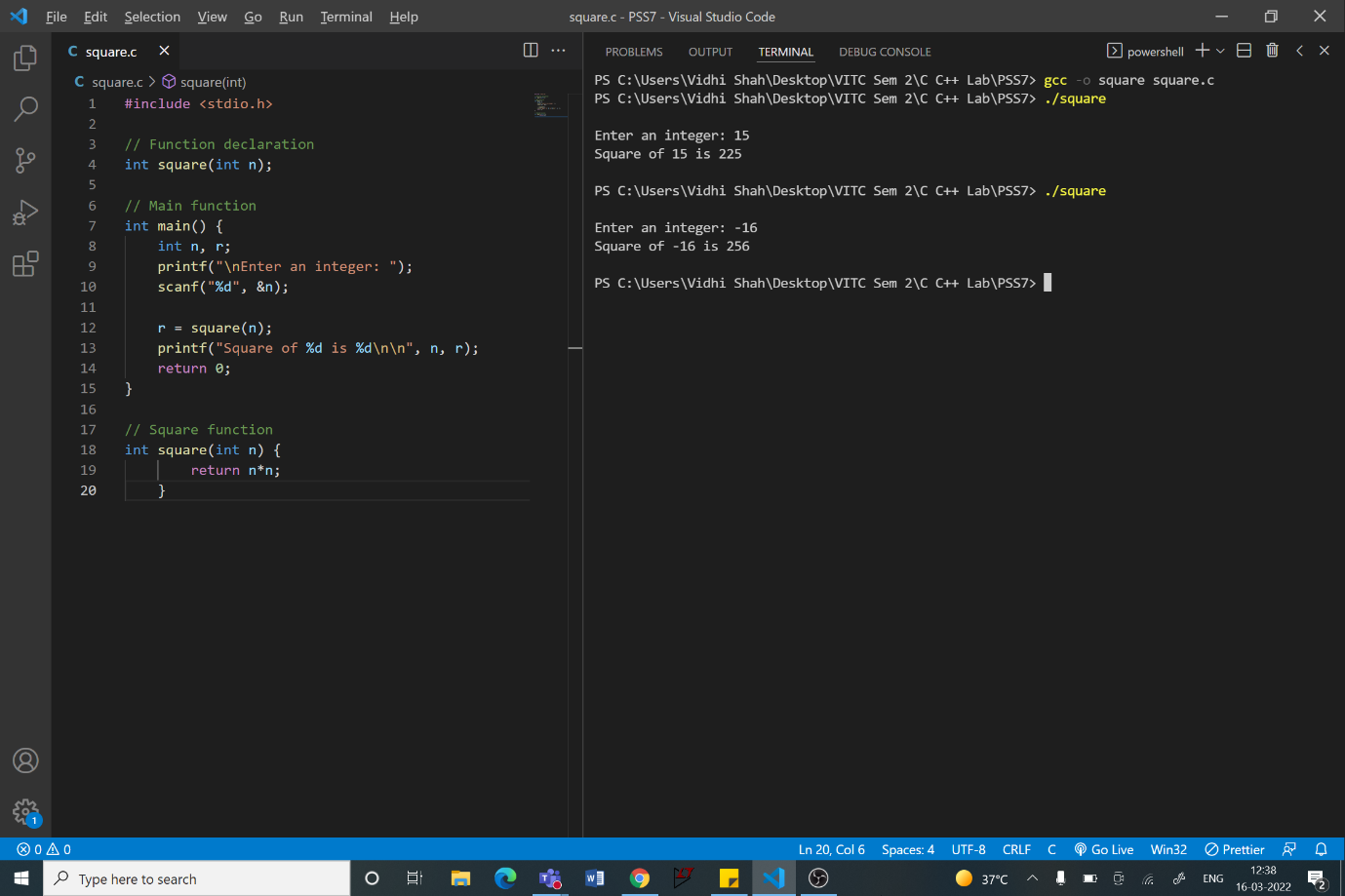
}

// Square function

int square(int n) {

        return n\*n;

    }

****

**Q3**

**Aim:**

Write a ‘C’ Program to perform Matrix Addition for n x n Matrix. Get the user input for ‘n’.

**Procedure:**

**Input:**

A natural number, n

Matrix 1 and Matrix 2 of dimension n x n

**Output:**

Matrix addition of matrix 1 and matrix 2

**Algorithm:**

Step 1: Declare variables n, i, j

Step 2: Declare ‘matrixaddn’ function with return type ‘void’ and argument of

2 integer matrices

**Main Function**

Step 1: Read a natural number ‘n’ from user

Step 2: Declare 2 integer matrices, ‘m1’ and ‘m2’, of dimension n x n

Step 2: Read and store elements of matrix 1, m1, using for loop

Step 3: Read and store elements of matrix 2, m2, using for loop

Step 4: Call ‘matrixaddn’ function

**Matrix Addition Function**

Step 1: Declare matrix ‘ma’ of dimension n x n

Step 2: Initialise i to 0. For i < n

Initialise j to 0. For j < n

ma[i][j] = m1[i][j] + m2[i][j]

Increment j

Increment i

Step 3: Print elements of matrix ‘ma’ using for loop

**Code:**

**Main Function**

#include <stdio.h>

// Function declaration

int n, i, j;

void matrixaddn(int m1[n][n], int m2[n][n]);

// Main function

int main() {

    printf("\nEnter a natural number 'n' for dimensions of the matrices: ");

    scanf("%d", &n);

    // Matrix input from user

    int m1[n][n], m2[n][n];

    printf("\nEnter elements for Matrix 1:\n");

    for (i = 0; i < n; i++) {

        for (j = 0; j < n; j++) {

            scanf("%d", &m1[i][j]);

        }

    }

    printf("\nEnter elements for Matrix 2:\n");

    for (i = 0; i < n; i++) {

        for (j = 0; j < n; j++) {

            scanf("%d", &m2[i][j]);

        }

    }

    matrixaddn(m1, m2);

    printf("\n");

    return 0;

}

**Matrix Addition Function**

// Matrix Addition function

void matrixaddn(int m1[n][n], int m2[n][n]) {

        int ma[n][n];

        for (i = 0; i < n; i++) {

            for (j = 0; j < n; j++) {

                ma[i][j] = m1[i][j] + m2[i][j];

            }

        }

        printf("Matrix Addition:\n");

        for (i = 0; i < n; i++) {

            for (j = 0; j < n; j++) {

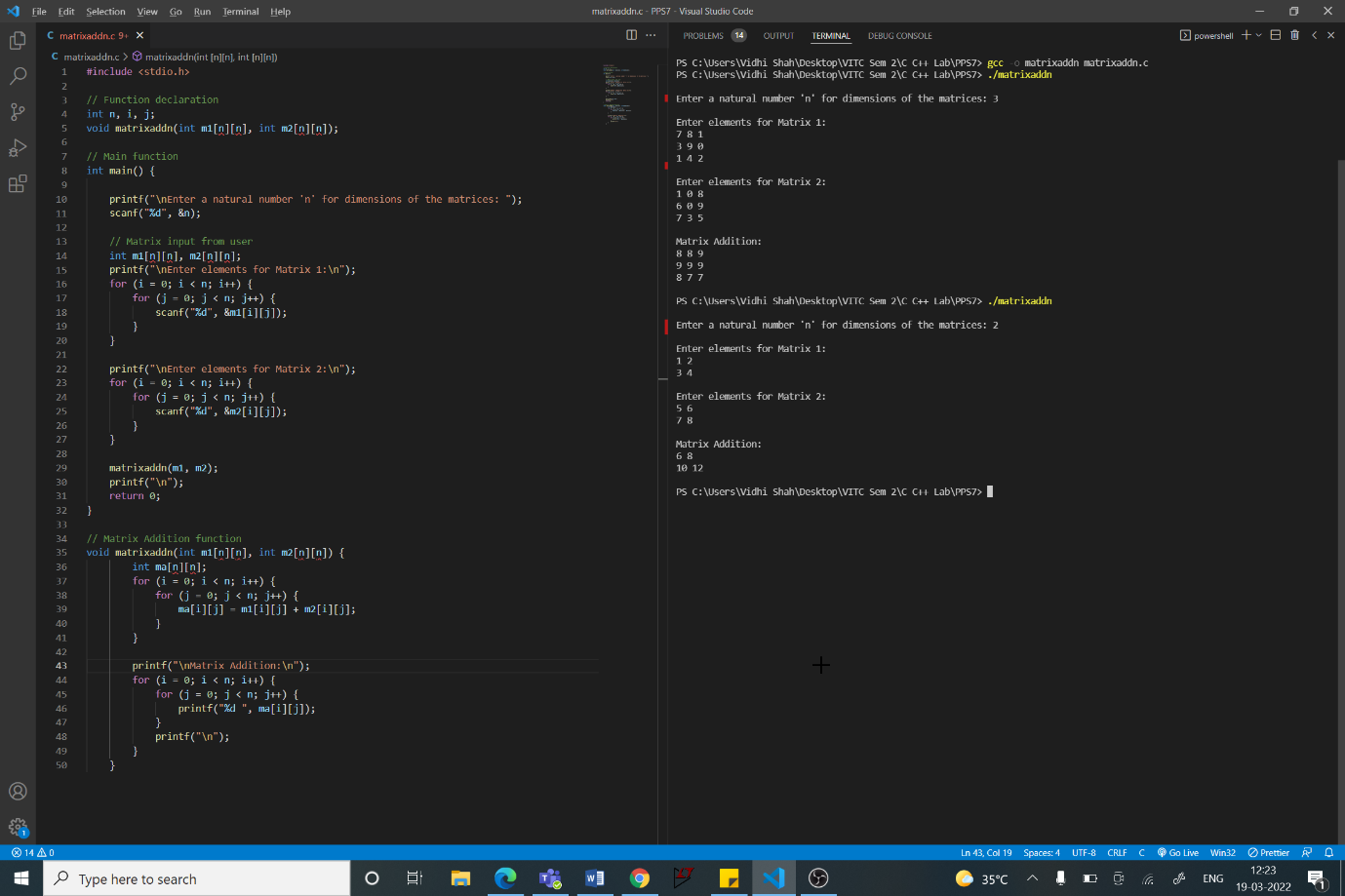
                printf("%d ", ma[i][j]);

            }

            printf("\n");

        }

    }

****