**Reg. No: 21BCE1297 Name: Vidhi Shah Date: 10/03/22**

**PPS6**

**Q1**

**Aim:**

Write a ‘C’ Program to check whether the i/p number is palindrome or not?

**Procedure:**

**Input:**

A number, n

**Output:**

Palindrome or not a palindrome

**Algorithm:**

**Method 1**

Step 1: Initialise variables and read n as an integer

Step 2: Let reverse number, r = 0 and n1 = n

Step 3: While n1 > 0

r = r\*10 + (n1%10)

Integer division of n1 by 10

Step 4: If the reverse, r is equal to n

Print n is a palindrome

Step 5: If the reverse, r is not equal to n

Print n is not a palindrome

**Method 2**

Step 1: Initialise variables and read n as a string

Step 2: Copy the string from n to r (reverse string)

Step 3: Reverse the string r

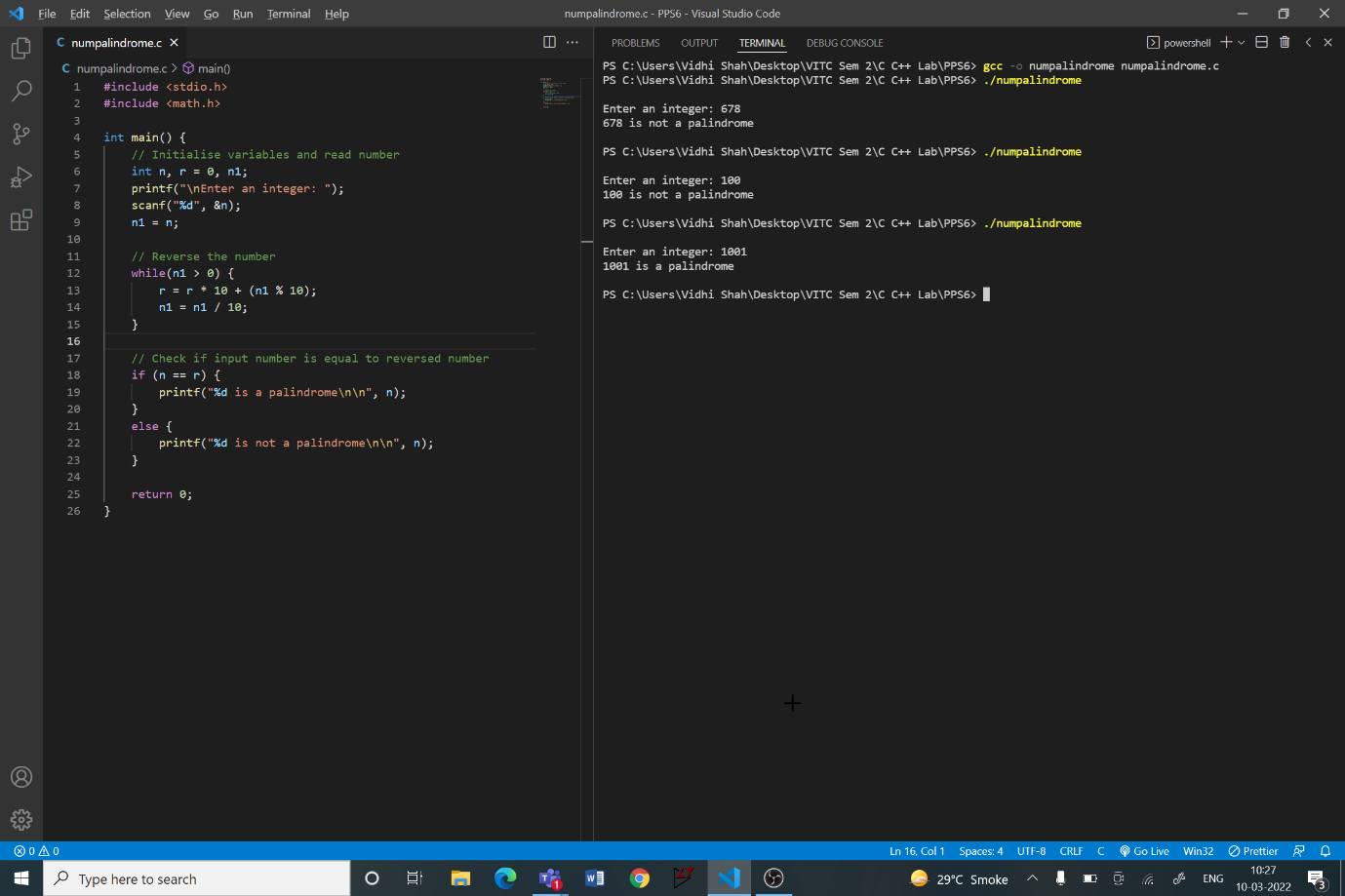
Step 4: Compare the 2 strings n and r and store the integer result in a variable, i

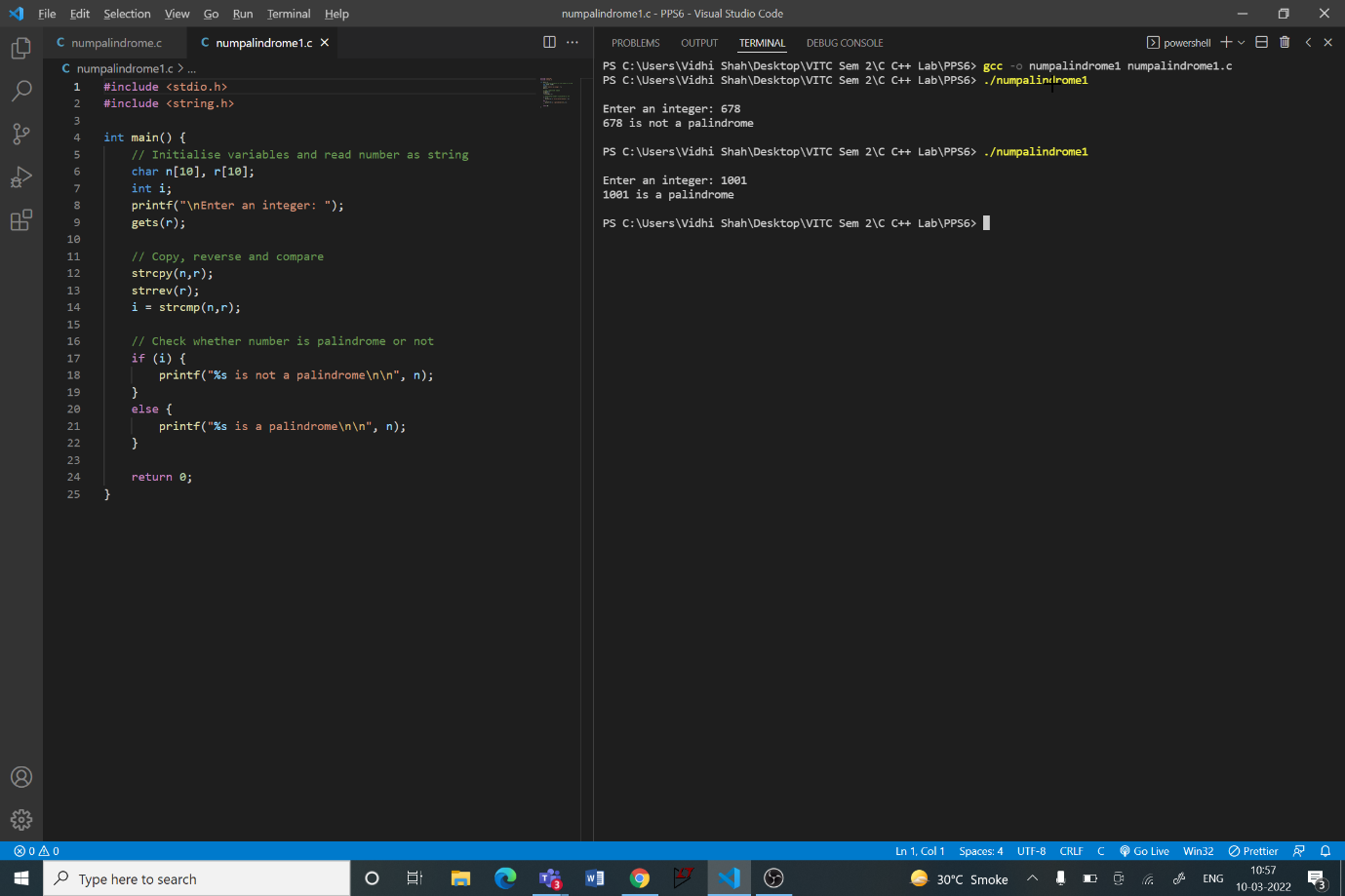
Step 5: If i is equal 0

Print n is a palindrome

Step 6: If i is not equal to 0

Print n is not a palindrome

**Code:**

****

**Q2**

**Aim:**

Write a program to display the following pattern:

1

2 2

3 3 3

4 4 4 4

**Procedure:**

**Input:**

A number, n

**Output:**

Pattern

**Algorithm:**

Step 1: Initialise variables and read n

Step 2: Initialise i to 1. Until i is less than or equal to n

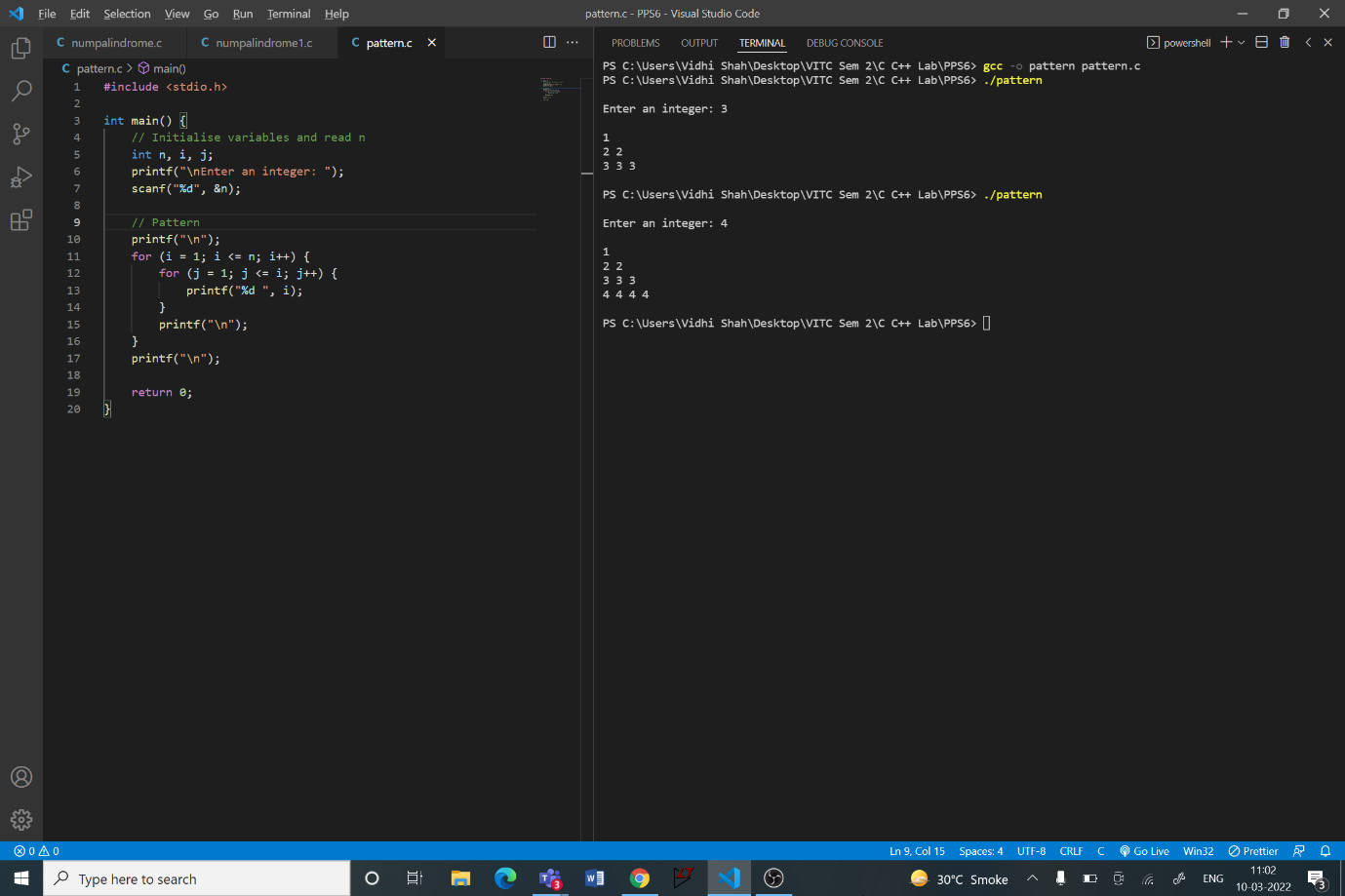
Initialise j to 1. Until j is less than or equal to i

Print i

Increment j by 1

Go to next line

Increment i by 1

**Code:**

**Q3**

**Aim:**

Given n number of elements containing positive, negative, Zero’s, odd and even numbers, write a C program to print the number of positive, negative, Zero’s, odd and even numbers in the array.

**Procedure:**

**Input:**

Number of elements, n

Next n lines contain n numbers

**Output:**

Number of positive, negative, zero, odd and even elements

**Algorithm:**

Step 1: Initialise variables and read n

Step 2: Initialise array of size n

Step 3: Using for loop read each number and store it in the array

Step 4: For each number in the array

If number is greater than 0

Increment positive count by 1

If number is less than 0

Increment negative count by 1

If number is equal to 0

Increment zero count by 1

If number is not divisible by 2

Increment odd count by 1

If number is divisible by 2

Increment even count by 1

Step 5: Display all the counts in separate lines

**Code:**

#include <stdio.h>

int main() {

    // Initialise variables and read number

    int n, i, p = 0, ne = 0, z = 0, o = 0, e = 0;

    printf("\nEnter number of elements: ");

    scanf("%d", &n);

    // Initialise array and store numbers in the array

    int arr[n];

    for (i = 0; i < n; i++) {

        printf("Enter Number %d: ", i + 1);

        scanf("%d", &arr[i]);

    }

    // Classify numbers

    for (i = 0; i < n; i++) {

        if (arr[i] > 0)

        p++;

        if (arr[i] < 0)

        ne++;

        if (arr[i] == 0)

        z++;

        if (arr[i] % 2 != 0)

        o++;

        if (arr[i] % 2 == 0)

        e++;

    }

    // Display classification count

    printf("\nPositive Numbers: %d\n", p);

    printf("Negative Numbers: %d\n", ne);

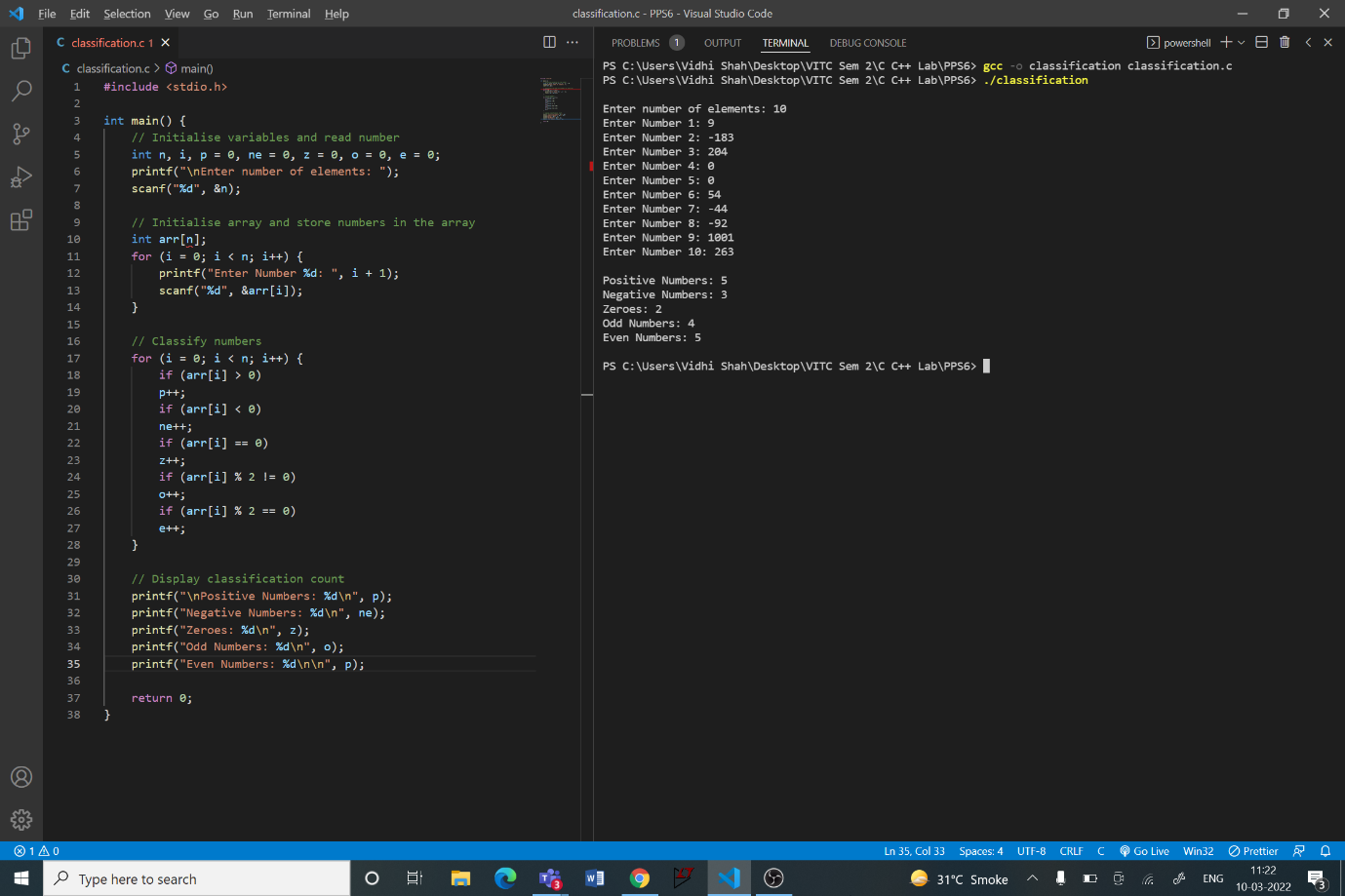
    printf("Zeroes: %d\n", z);

    printf("Odd Numbers: %d\n", o);

    printf("Even Numbers: %d\n\n", p);

    return 0;

}

****

**Q4**

**Aim:**

In an orphanage there were about 5 to 7 babies in the age group of 2months to 1 year. The founder of the orphanage wanted to name all these babies with special names. The names were special since all the names to be selected are palindromes. The founder’s friend gave him a list of baby names. The founder needs to select the palindrome names out of the list. Help the founder by implementing a C code in identifying the palindrome names out of the list. Also your program needs to count the number of palindrome names present in the list. If there is no palindrome string print 0.

Eg: If the list is

sachin

eve

arora

pilip

kamal

kumar

joshua

jones

nitin

emme

Your program has to print all palindrome strings.

eve

arora

pilip

nitin

emme

5 (number of palindrome strings present in the list)

**Procedure:**

**Input:**

Number of names, n

Next n lines contain n names

**Output:**

List of palindromic names

Number of palindromic names

**Algorithm:**

Step 1: Initialise variables and read n, initialise count to 0

Step 2: Initialise a character array (size n) of pointers (array of strings)

Step 3: Use for loop. For each iteration

Read name

Copy the string from name to reverse

Reverse the string reverse

Compare name and reverse and store the integer result in a variable, j

If j is equal 0

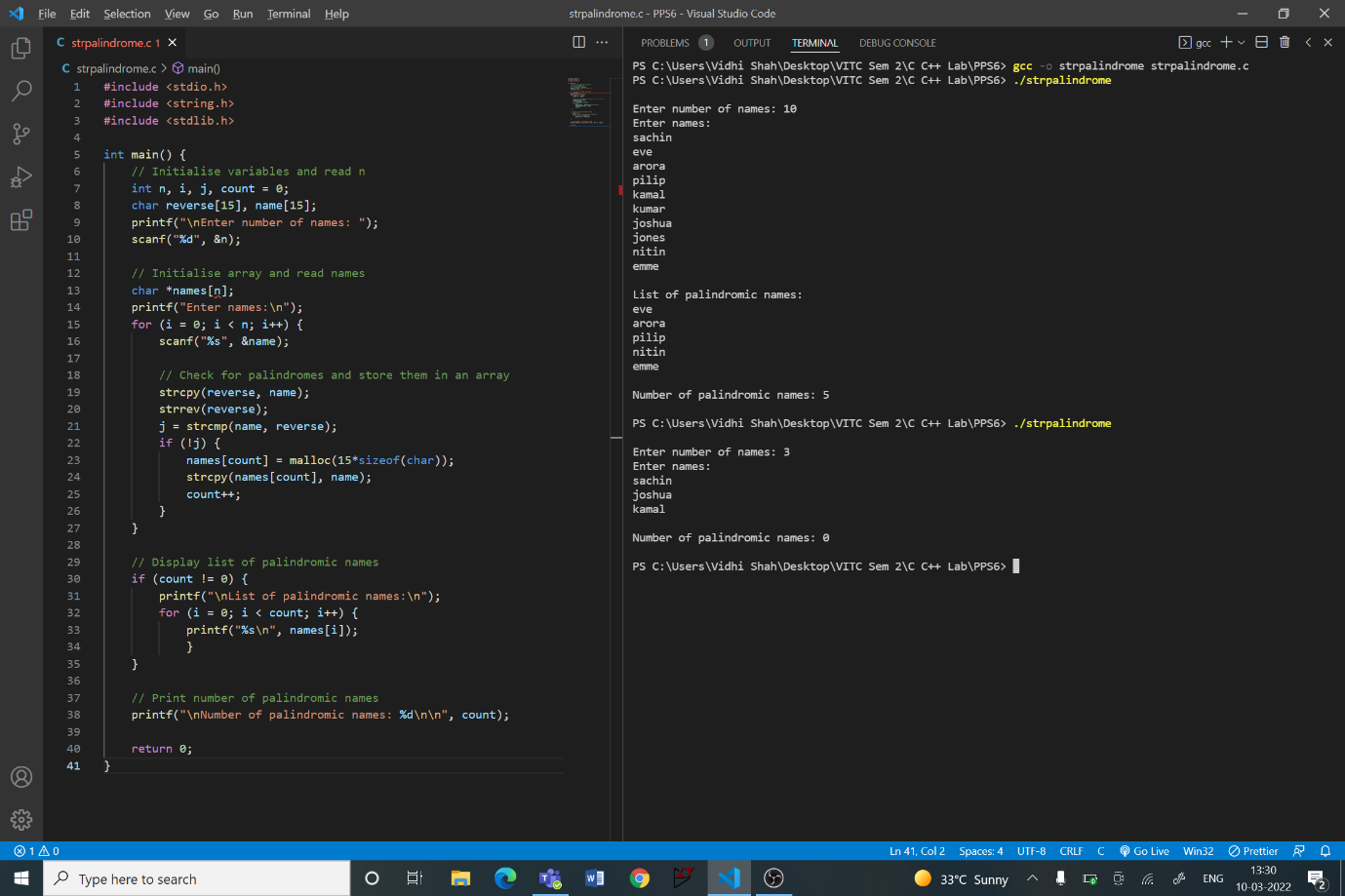
Allocate memory to countth position of the array to store a string

Copy the string name in countth position of the array

Increment count by 1

Step 6: Print all names stored in the array using for loop

Step 7: Print number of names printed

**Code:**