**Name:Deepanshu Rawat**

**SAP ID:500097692**

**Batch: B-46**

**Subject: DS LAB**

**EXPERIMENT-1**

**Title:** Array & Structure

**Objective:** To apply the concept of array, structure and experiment on nested array and array of structures.

**List of Lab Activities:** Write C program, compile, execute and test the code using Linux C compiler with suitable test cases.

1. Find sum of all array elements using recursion.

CODE:

#include<stdio.h>

int sum(int\*,int);

int main(){

    int n;

    printf("\n Enter the size of the array:");

    scanf("%d",&n);

    int a[n];

    printf(" Enter the array elements:");

    for (int i = 0; i < n; ++i)

    {

        scanf("%d",&a[i]);

    }

    printf("\n The sum of the array elements: %d",sum(a,n));

    return 0;

}

int sum(int *a*[],int *n*){

    if(*n*<=0){

        return 0;

    }

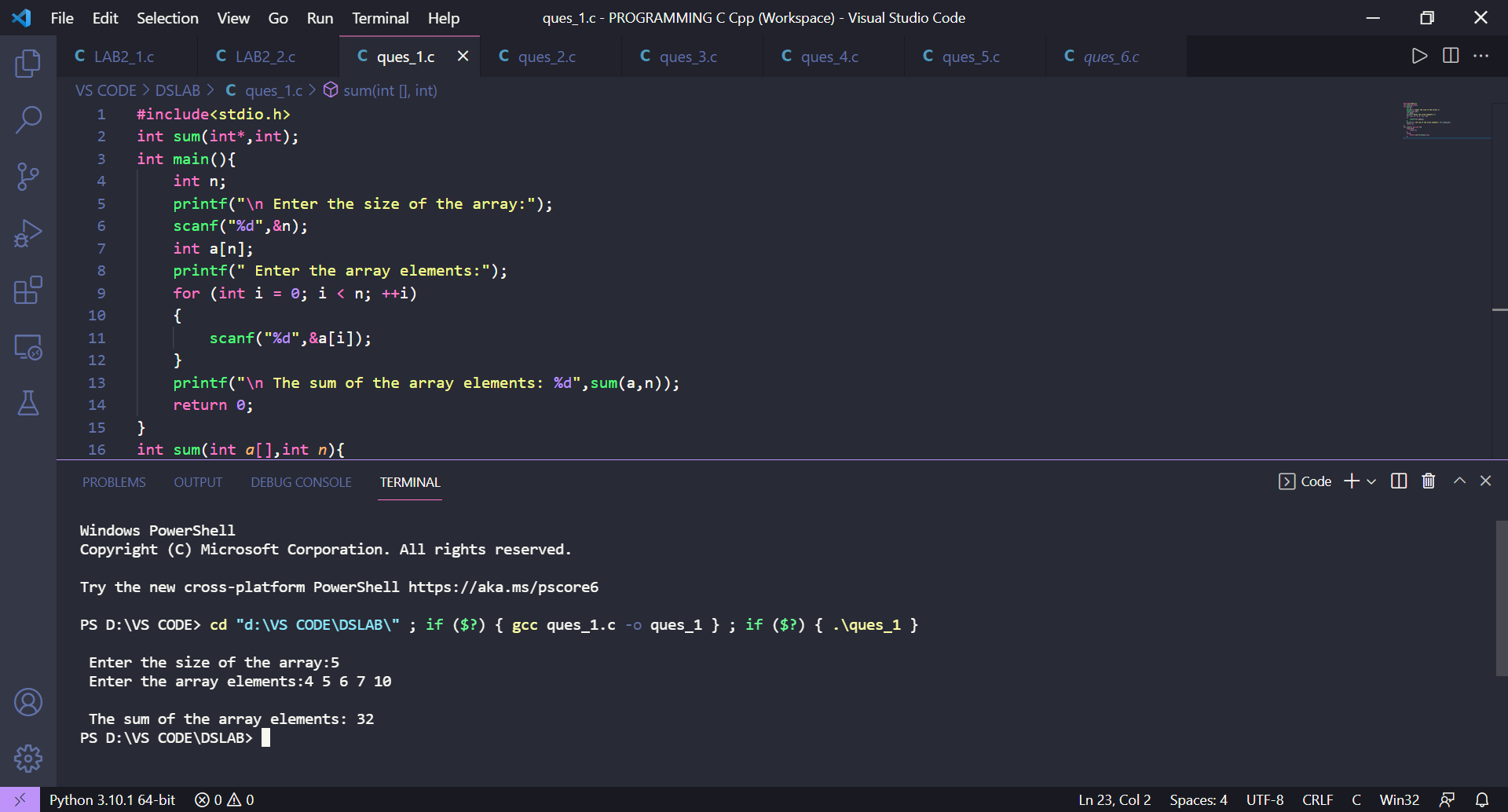
    else{

        return *a*[*n*-1]+sum(*a*,*n*-1);

    }

}

OUTPUT for 1st program:



2. Create an array ‘a1’ with ‘n’ elements. Insert an element in ith position of ‘a1’ and also delete an element from jth position of ‘a1’.

CODE:

#include<stdio.h>

void insert(int\*,int);

void delete(int\*,int);

int main(){

    int n;

    printf("\n Enter the size of the array:");

    scanf("%d",&n);

    int arr[100];

    printf("\n Enter the array elements:");

    for(int i=0;i<n;i++){

        scanf("%d",&arr[i]);

    }

    int choice;

    printf("Enter 1 for insertion and 2 for deletion:");

    scanf("%d",&choice);

    if(choice==1){

        insert(arr,n);

    }else{

        delete(arr,n);

    }

    return 0;

}

void insert(int *arr*[],int *n*){

    int insert\_index;

    int a;

    printf("\n Enter the insert index:");

    scanf("%d", &insert\_index);

    printf("\n Enter the element to be inserted:");

    scanf("%d", &a);

    for (int i = *n* - 1; i >= insert\_index - 1; i--){

*arr*[i+1] = *arr*[i];

    }

*arr*[insert\_index-1] = a;

    printf("\n Array after the operation is:");

    for (int i = 0; i <= *n*; i++){

        printf("%d ", *arr*[i]);

    }

}

void delete(int *arr*[],int *n*){

    int delete\_index;

    int a;

    printf("\n Enter the delete index:");

    scanf("%d", &delete\_index);

    if(delete\_index>*n* || delete\_index<0){

        printf("\n Operation cannot be performed!");

    }

    else{

        for (int i =delete\_index-1; i <*n*-1; i++){

*arr*[i] = *arr*[i+1];

        }

        printf("\n Array after the operation is:");

        for (int i = 0; i < *n*-1; i++){

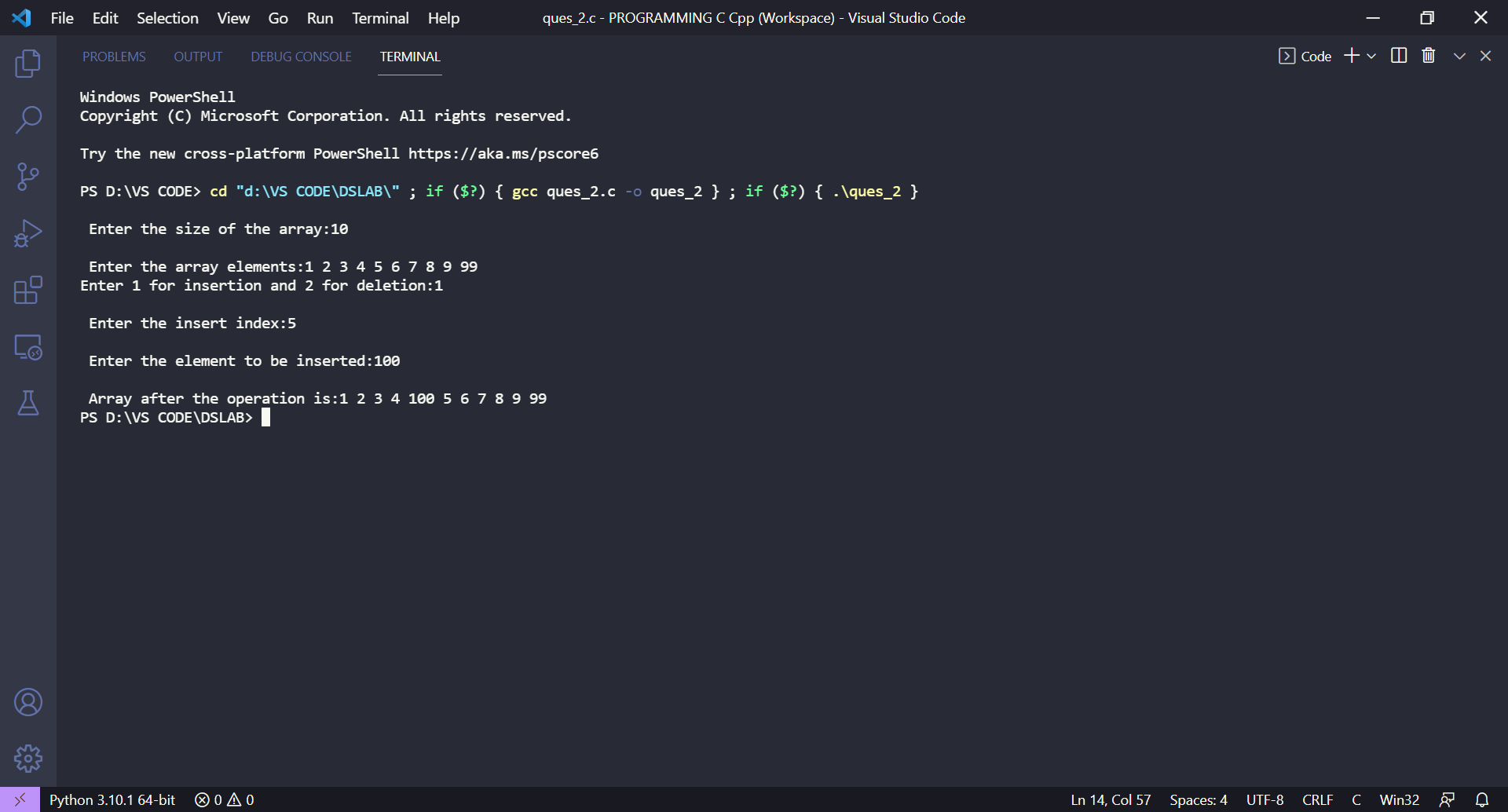
            printf("%d ", *arr*[i]);

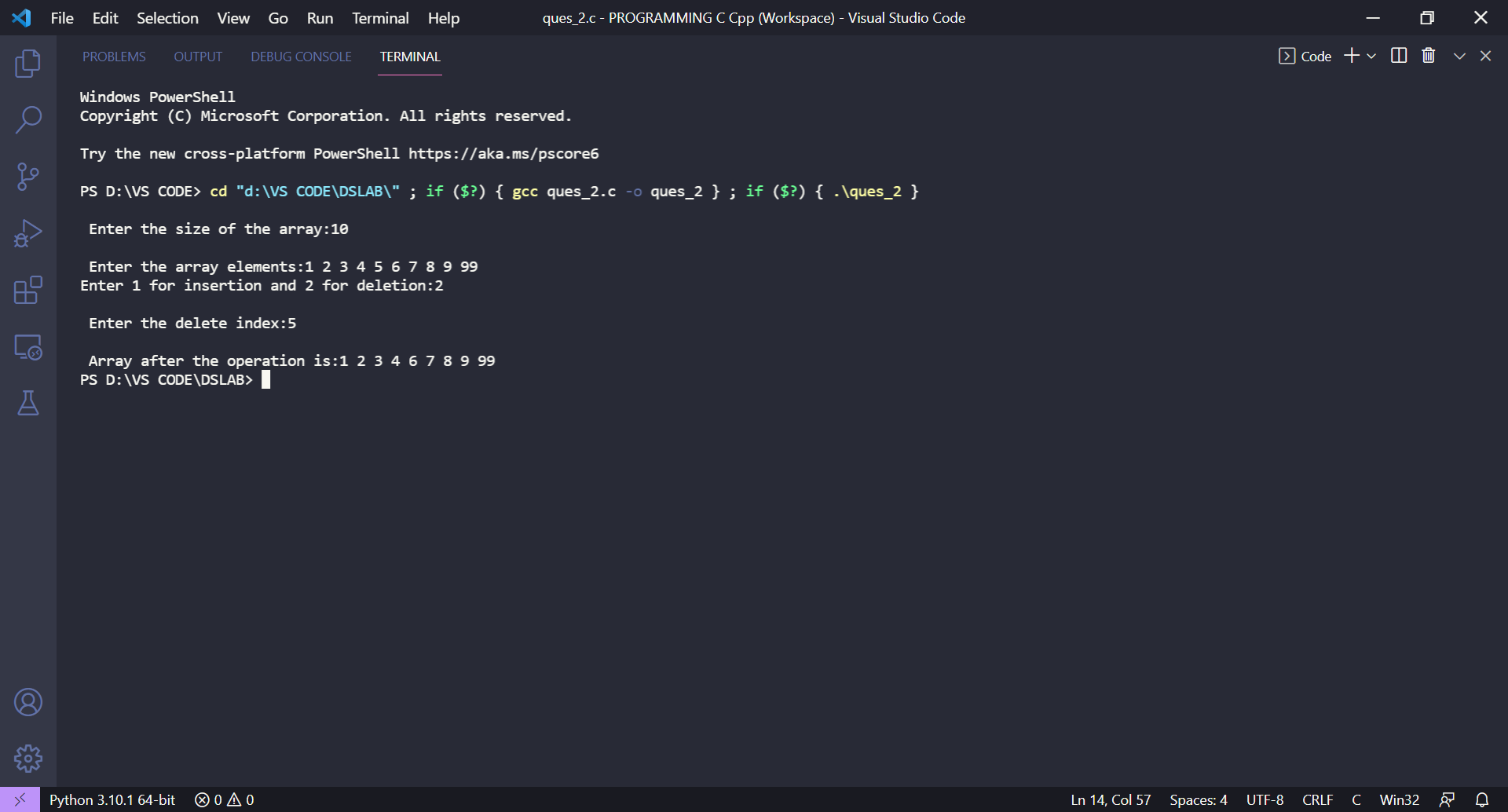
        }

    }

}

OUTPUT for 2nd program:





3. Convert uppercase string to lowercase using for loop.

CODE:

#include <stdio.h>

int main()

{

    char str[1000];

    printf("\n Enter the string in UPPER case: ");

    gets(str);

    for(int i=0; str[i]!='\0'; i++)

    {

        if(str[i]>='A' && str[i]<='Z')

        {

            str[i] = str[i] + 32;

        }

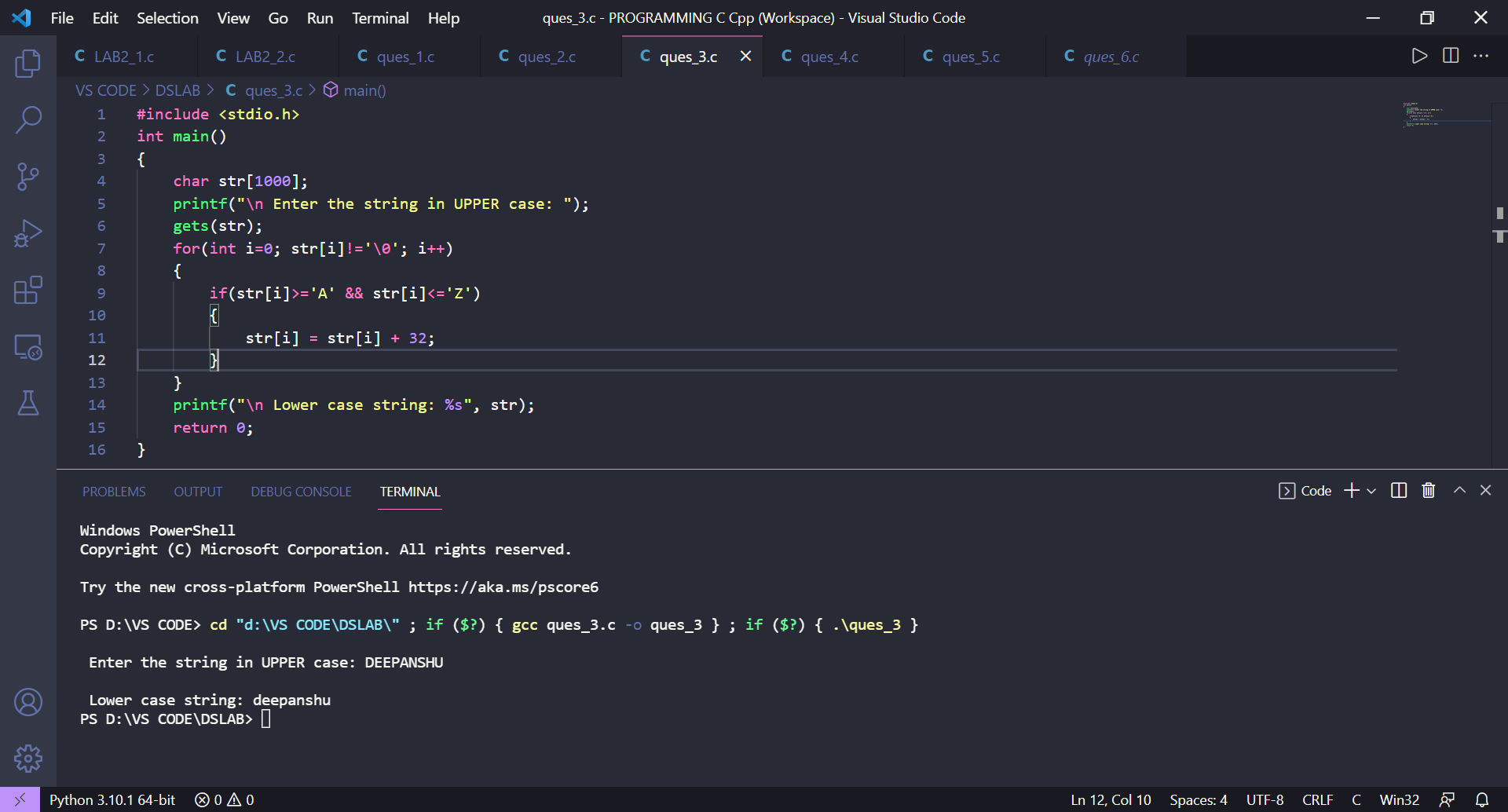
    }

    printf("\n Lower case string: %s", str);

    return 0;

}

OUTPUT for 3rd program:



4. Find the sum of rows and columns of matrix of given order (row x column).

CODE:

#include<stdio.h>

int main(){

    int r,c,sumr=0,sumc=0;

    printf("Enter the rows and columns of the matrix:");

    scanf("%d",&r);

    scanf("%d",&c);

    int num[r][c];

    printf("\n Enter the array elements:");

    for(int i=0;i<r;i++){

        for(int j=0;j<c;j++){

            scanf("%d",&num[i][j]);

        }

    }

    for(int i=0;i<r;i++){

        for(int j=0;j<c;j++){

            sumr+=num[i][j];

        }

        printf("\n The sum of %d row is: %d",i+1,sumr);

        sumr=0;

    }

    for(int i=0;i<r;i++){

        for(int j=0;j<c;j++){

            sumc+=num[j][i];

        }

        printf("\n The sum of %d column is: %d",i+1,sumc);

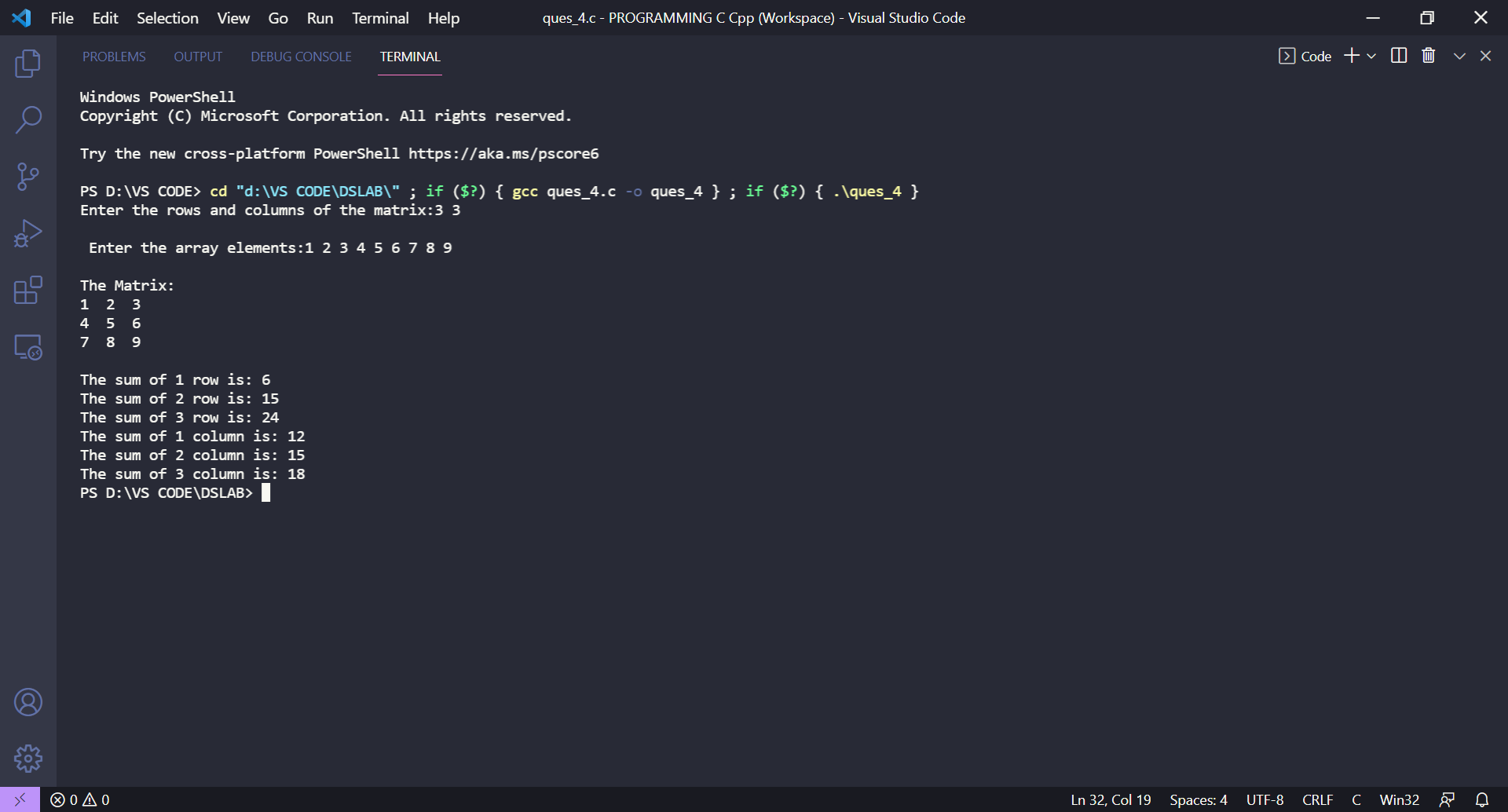
        sumc=0;

    }

    return 0;

}

OUTPUT for 4th program:



5. Find the product of two matrices using pointers.

CODE:

#include <stdio.h>

#define row 3

#define col 3

void MAT\_input(int *arr*[][col]) {

   int i,j;

   printf("\nEnter the matrix elements:\n");

   for (i = 0; i < row; i++) {

      for (j = 0; j < col; j++) {

         scanf("%d", &(\*(\*(*arr* + i) + j)));

      }

   }

}

void MAT\_print(int *arr*[][col]) {

   int i,j;

   for (i = 0; i < row; i++) {

      for (j = 0; j < col; j++) {

         printf("%d ", (\*(\*(*arr* + i) + j)));

      }

      printf("\n");

   }

}

void MAT\_multi(int *arr1*[][col], int *arr2*[][col], int *prod*[][col]) {

   int i, j, k;

   int sum=0;

   for (i = 0; i < row; i++) {

      for (j = 0; j < col; j++) {

         for (k = 0; k < col; k++) {

            sum += (\*(\*(*arr1* + i) + k)) \* (\*(\*(*arr2* + k) + j));

         }

         \*(\*(*prod* + i) + j) = sum;

         sum=0;

      }

   }

}

int main() {

   int arr1[row][col];

   int arr2[row][col];

   int prod[row][col];

   printf("\nEnter elements in first matrix of size %dx%d\n", row, col);

   MAT\_input(arr1);

   printf("\nEnter elements in second matrix of size %dx%d\n", row, col);

   MAT\_input(arr2);

   printf("\nMATRIX 1:\n");

   MAT\_print(arr1);

   printf("\nMATRIX 2:\n");

   MAT\_print(arr2);

   MAT\_multi(arr1, arr2, prod);

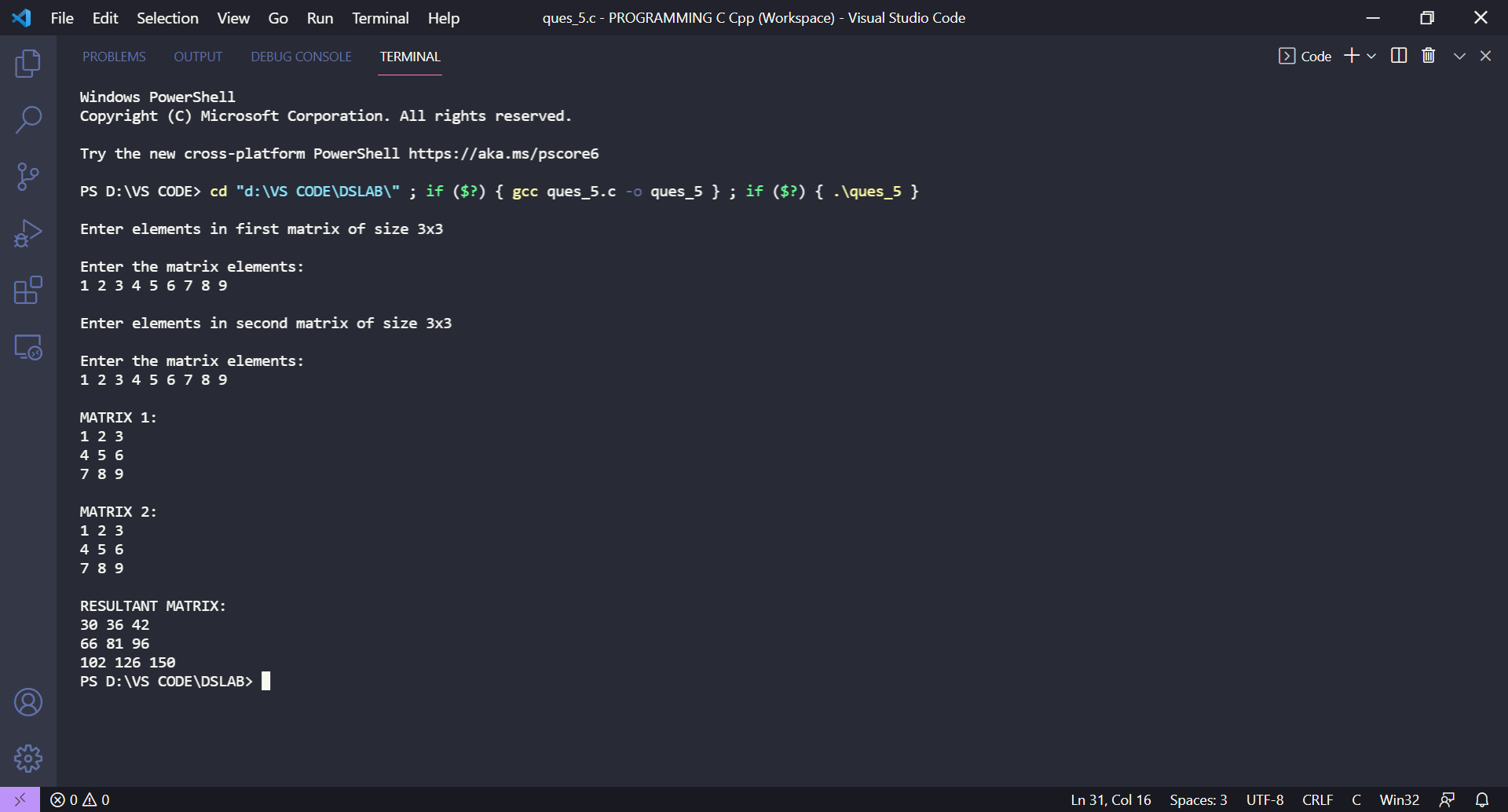
   printf("\nRESULTANT MATRIX: \n");

   MAT\_print(prod);

   return 0;

}

OUTPUT for 5th program:



6. Store ‘n’ numbers (integers or real) in an array. Conduct a linear search for a given number and report success or failure in the form of a suitable message.

CODE:

  #include<stdio.h>

    int search(float\*,int,float);

    int main(){

        int n;

        float target;

        printf("\n Enter the size of the array:");

        scanf("%d",&n);

        float num[n];

        printf("\n Enter the array elements:");

        for(int i=0;i<n;i++){

            scanf("%f",&num[i]);

        }

        printf("\n Enter the target element:");

        scanf("%f",&target);

        int val=search(num,n,target);

        if(val==-1){

            printf("\n Search element not found!!");

        }else{

            printf("\n Target element found at %dth position",val);

        }

        return 0;

    }

    int search(float *num*[],int *n*,float *target*){

        for(int i=0;i<*n*;i++){

            if(*target*==*num*[i]){

                return i+1;

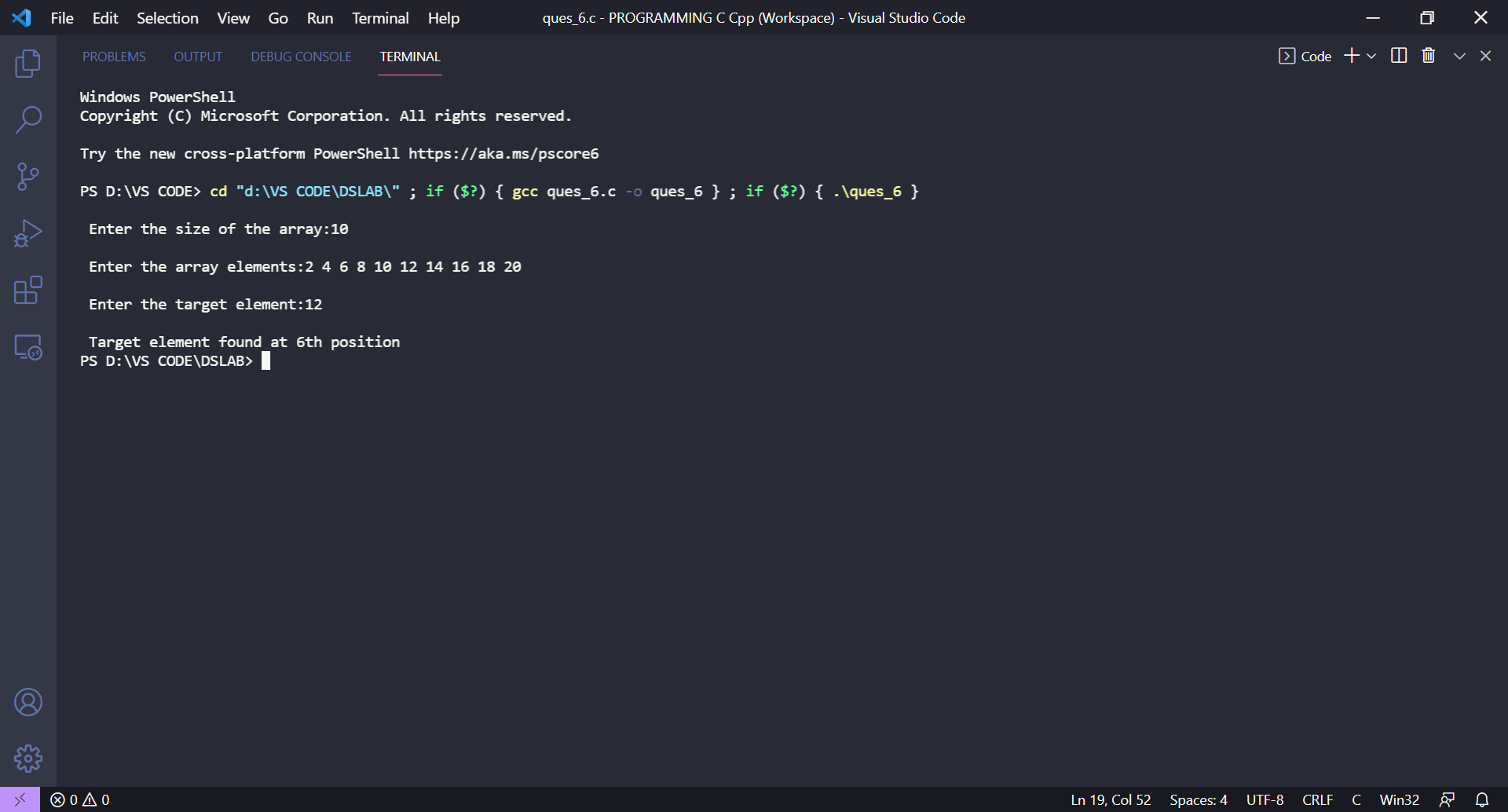
            }

        }

        return -1;

    }

OUTPUT for 6th program:



**List of Practice Activities:** Write C program, compile, execute and test the code using Linux C compiler with suitable test cases.

1. Store ‘n’ numbers (integers or real) in an array in ascending or descending order. Conduct a binary search for a given number and report success or failure in the form of a suitable message.

CODE:

#include<stdio.h>

#include<stdbool.h>

int search(float *arr*[],int *lb*,int *ub*,float *target*){

    bool isDec=*arr*[*lb*]>*arr*[*ub*];

    while(*lb*<=*ub*){

        int mid=*lb*+(*ub*-*lb*)/2;

        if(*arr*[mid]==*target*){

           return mid;

        }

        if(isDec==false){

            if(*arr*[mid]>*target*){

*ub*=mid-1;

            }else{

*lb*=mid+1;

            }

        }else{

            if(*arr*[mid]>*target*){

*lb*=mid+1;

            }else{

*ub*=mid-1;

            }

        }

    }

    return -1;

}

int main(){

    int n;

    float target;

    printf("\nEnter the size of the array:");

    scanf("%d",&n);

    float arr[n];

    printf("\nEnter the array elements in either Ascending or Descending order:");

    for(int i=0;i<n;i++){

        scanf("%f",&arr[i]);

    }

    printf("\nEnter the target element:");

    scanf("%f",&target);

    int ans=search(arr,0,n-1,target);

    if(ans!=-1){

        printf("\nTarget element is found at %dth position",ans+1);

    }else{

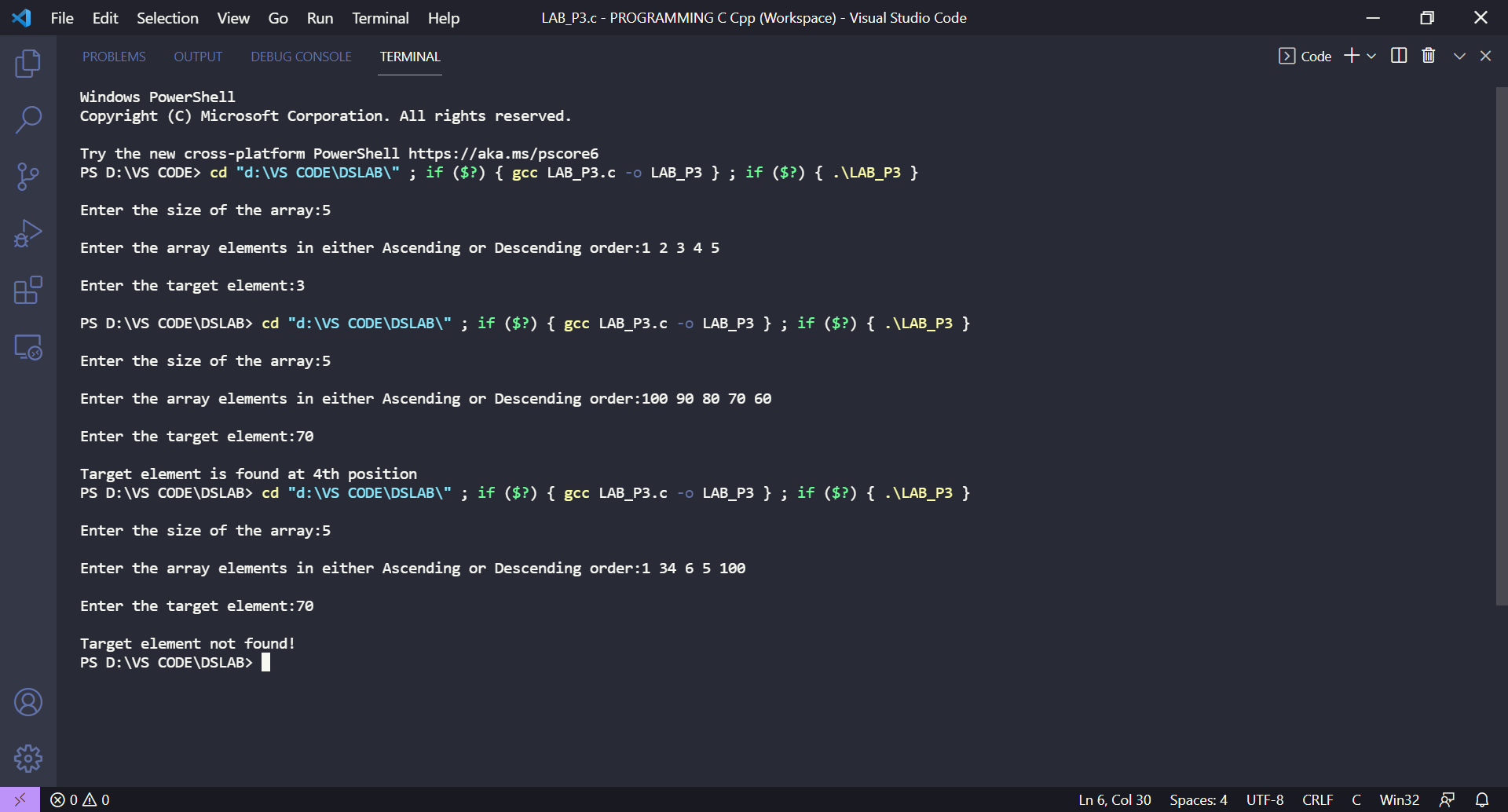
        printf("\nTarget element not found!");

    }

    return 0;

}

**OUTPUT of 1st program:**

****

1. Find if the given matrix of order (m x n) is a Sparse matrix or not. [Assume that a matrix can

become a sparse matrix if more than half the total number of its elements have the value zero]

CODE:

#include<stdio.h>

int main(){

    int m,n;

    printf("\nEnter the number of rows:");

    scanf("%d",&m);

    printf("\nEnter the number of columns:");

    scanf("%d",&n);

    int arr[m][n];

    int checko=(m\*n)/2;

    int c=0;

    printf("\nEnter the array elements:");

    for(int i=0;i<m;i++){

        for(int j=0;j<n;j++){

            scanf("%d",&arr[i][j]);

        }

    }

    for(int i=0;i<m;i++){

        for(int j=0;j<n;j++){

            if(arr[i][j]==0){

                c++;

            }

        }

    }

    for(int i=0;i<m;i++){

        for(int j=0;j<n;j++){

            printf("%d ",arr[i][j]);

        }

        printf("\n");

    }

    if(c>checko){

        printf("\nThe given matrix is a Sparse Matrix");

    }else{

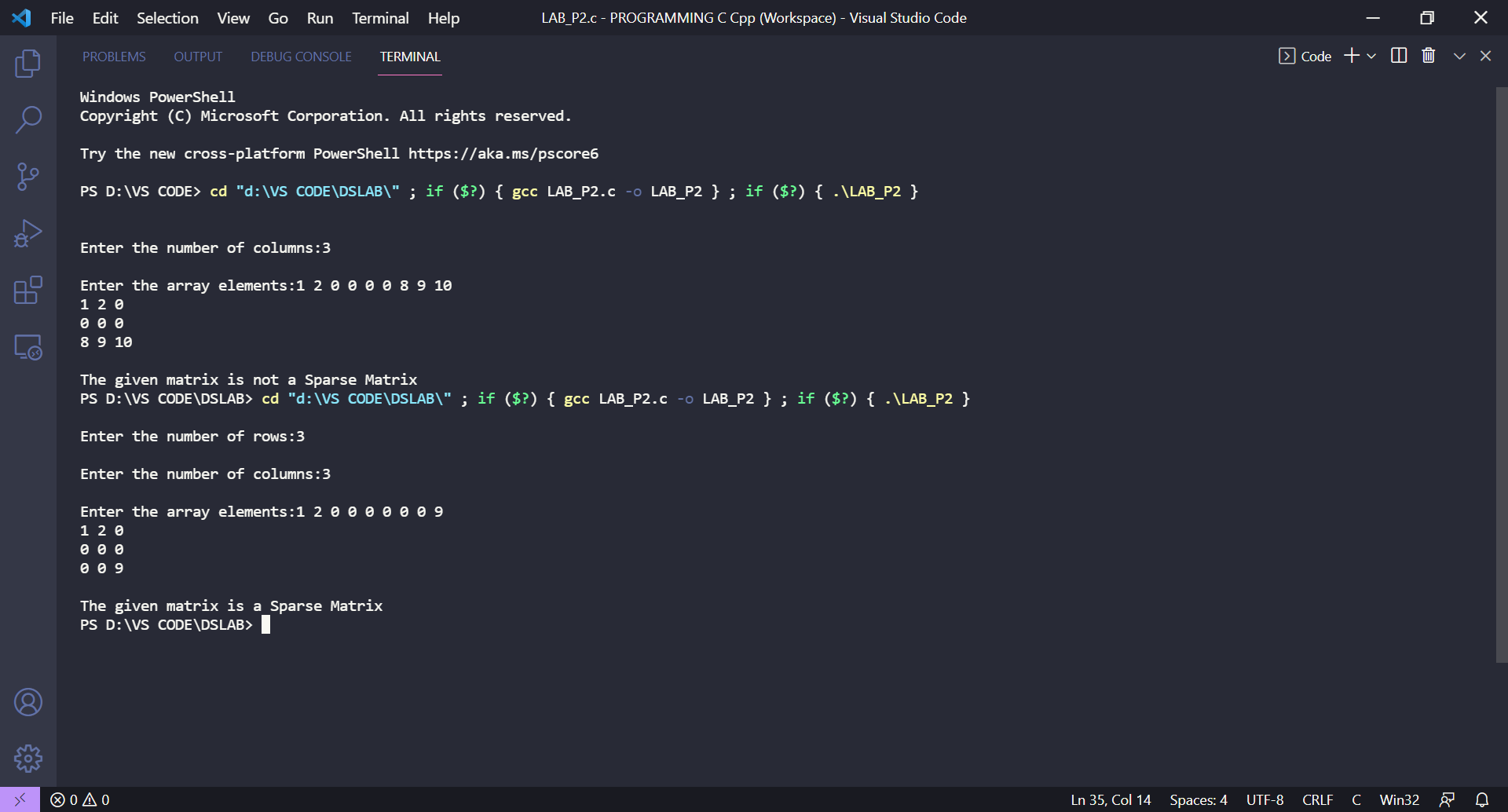
        printf("\nThe given matrix is not a Sparse Matrix");

    }

    return 0;

}

**OUTPUT of 2nd program:**

****

1. Find out the largest and smallest number in a given array.

CODE:

#include<stdio.h>

int sort(int *arr*[],int *len*){ //bubble sort

    for(int i=0;i<*len*;i++){

        for(int j=0;j<*len*-1-i;j++){

            if(*arr*[j]>*arr*[j+1]){

                int temp=*arr*[j];

*arr*[j]=*arr*[j+1];

*arr*[j+1]=temp;

            }

        }

    }

}

int main(){

    int n;

    printf("\nEnter the size of the array:");

    scanf("%d",&n);

    int arr[n];

    printf("\nEnter the array elemenets:");

    for(int i=0;i<n;i++){

        scanf("%d",&arr[i]);

    }

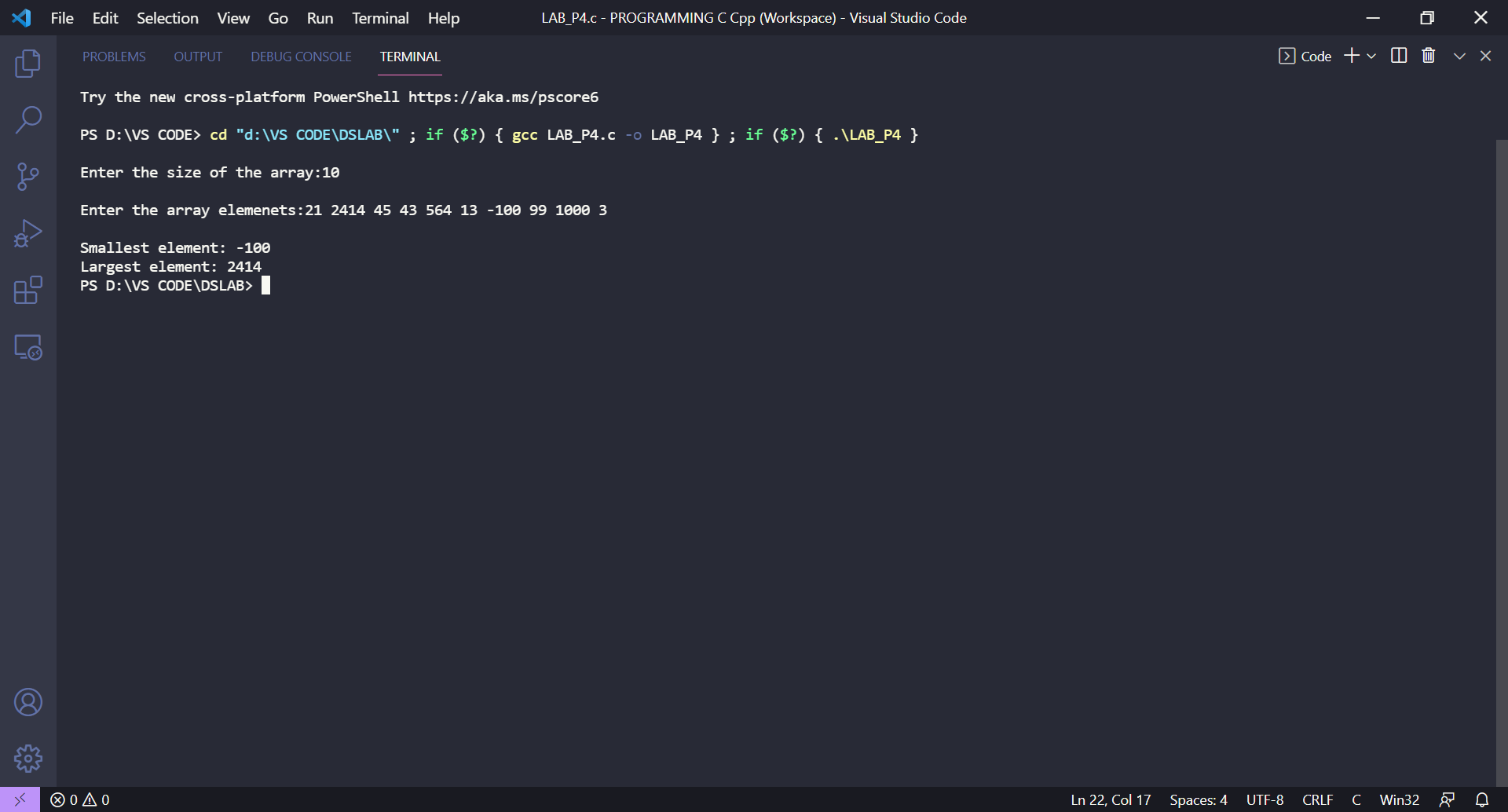
    sort(arr,n);

    printf("\nSmallest element: %d",arr[0]);

    printf("\nLargest element: %d",arr[n-1]);

}

**OUTPUT for 3rd program:**

****

**EXPERIMENT-2**

**Title**: Union and Dynamic Memory Allocation.

**Objective:** To implement the concept of union and experiment on dynamic memory allocation.

**List of Lab Activities:** Write C program, compile, execute and test the code using Linux C compiler with suitabletest cases.

1. Design a union ‘product’ to store the details of the product purchased like product name, price per unit, number of quantities purchased, and amount spent. Get the name, price per unit, and number of quantities of the product purchased. Calculate the amount spent on the product and then display all the details of the procured product.

CODE:

#include<stdio.h>

#include<string.h>

union *product*{

    char p\_name[100];

    float ppu;

    int nou;

};

int main(){

    union *product* var;

    char name[100];

    float price;

    int no;

    printf("\n The size occupied by the union Product:%d",sizeof(var));

    printf("\n");

    printf("\nEnter the product name:");

    gets(name);

    printf("\nEnter the price of a single unit(in ₹₹₹):");

    scanf("%f",&price);

    printf("\nEnter the no of items you want to buy:");

    scanf("%d",&no);

    printf("\n");

    printf("Displaying the product information....");

    strcpy(var.p\_name, name);

    printf("\nProduct name: %s",var.p\_name);

    var.ppu=price;

    printf("\nPrice of a single unit: %.2f",var.ppu);

    var.nou=no;

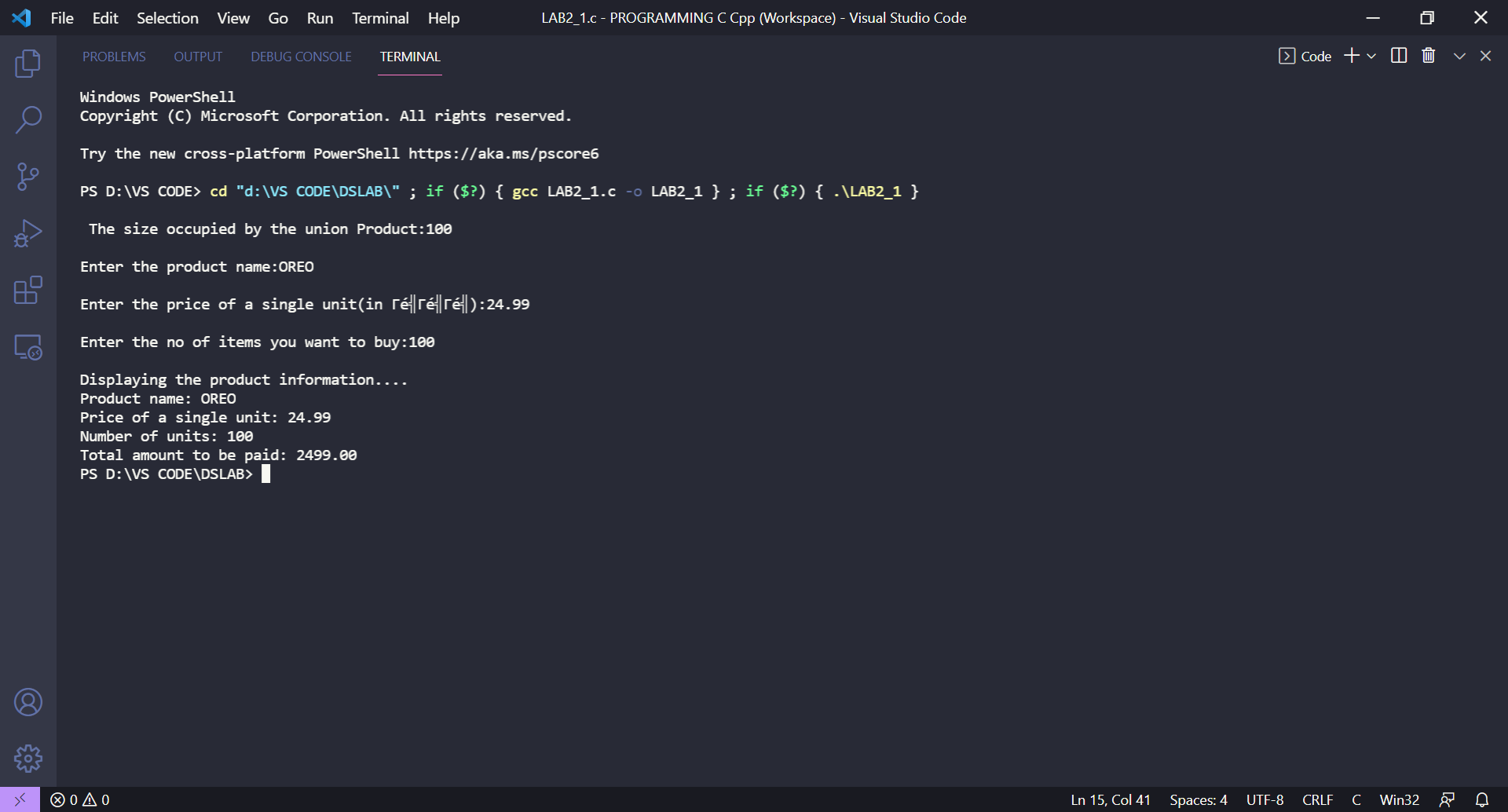
    printf("\nNumber of units: %d",var.nou);

    printf("\nTotal amount to be paid: %.2f",price\*(float)var.nou);

    return 0;

}

OUTPUT for 1st program:



2. Design a structure ‘subject’ to store the details of the subject like subject name and subject code. Using structure pointer allocate memory for the structure dynamically so as to obtain details of ‘n’ subjects using for loop.

CODE:

#include<stdio.h>

#include<stdlib.h>

struct *subject*{

  char sub\_name[100];

  int sub\_code;

}\*sub;

int main(){

  int n,m;

  printf("\nEnter the number of students:");

  scanf("%d",&m);

  for(int k=0;k<m;k++){

    printf("\nDetails of Student %d",k+1);

    printf("\nEnter the number of subjects whose details have to be stored  : ");

    scanf("%d",&n);

    sub = (struct *subject*\*)malloc(n\*sizeof(struct *subject*));

    for(int i=0;i<n;i++){

      printf("\n Enter the details of the subject %d ",i+1);

      printf("\n Enter the name of the subject : ");

      while(getchar()!='\n');

      fgets((sub+i)->sub\_name,100,stdin);

      printf(" Enter the code of the subject : ");

      scanf("%d",&(sub+i)->sub\_code);

    }

    printf("\n\tSubject Records");

    for(int i=0;i<n;i++){

      printf("\n\tSubject %d\n\tName = %s",i+1,(sub+i)->sub\_name);

      printf("\tSubject code = %d\n",(sub+i)->sub\_code);

    }

  }

  return 0;

}

OUTPUT for 2nd program:

