**CSIC 101**

**Problem Solving and Programming Skills using C (PSPS)**

For All programs, the **inputs must be read from the user**, input value should not be pre-fixed inside the code. Means that program should be able to compute results for different inputs just by re-executing the code and without changing the source code. For most programs, **functions** should be written.

1. Write a program to copy the contents of one 1 D array into another in the reverse order.

#include<stdio.h>

void input\_arr(int arr[],int n){

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

    printf("enter %d element\n",i+1);

        scanf("%d",&arr[i]);

    }

}

void print\_arr(int arr[],int n){

    for(int i=0;i<n;i++){

    printf("%d\n",arr[i]);

    }

}

void reverse\_array\_copy(int arr[],int arr2[],int n){

    for(int i=n-1;i>-1;i--){

        arr2[n-1-i]=arr[i];

    }

}

    int main(){

    int n;

    printf("enter no. of elements in array\n");

    scanf("%d",&n);

    int arr[n];

    input\_arr(arr,n);

    printf("entered array is\n");

    print\_arr(arr,n);

    int arr2[n];

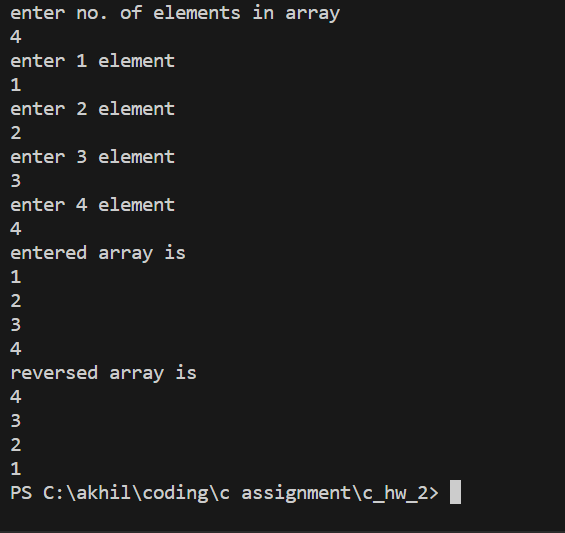
    reverse\_array\_copy(arr,arr2,n);

    printf("reversed array is\n");

    print\_arr(arr2,n);

return 0;

}



1. Write a program which performs the following tasks:

− initialize an integer array of 10 elements in main( )

− Add all the values and print the sum

− Multiply all the values by three and

− Print the tables from 1 to 10 in tabular forms

#include<stdio.h>

void input\_arr(int arr[],int n){

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

    printf("enter %d element\n",i+1);

        scanf("%d",&arr[i]);

    }

}

void print\_arr(int arr[],int n){

    for(int i=0;i<n;i++){

    printf("%d\n",arr[i]);

    }

}

    int main(){

    int arr[10];

    int n=10;

input\_arr(arr,n);

    int sum=0;

    for (int i=0;i<n;i++){

        sum+=arr[i];

    }

    printf("%d",sum);

    for(int i=0;i<n;i++){

        arr[i]=3\*arr[i];

    }

    print\_arr(arr,n);

    for(int i=1;i<11;i++){

        for(int j=1;j<11;j++){

            printf("%dX%d=%d\t",i,j,i\*j);

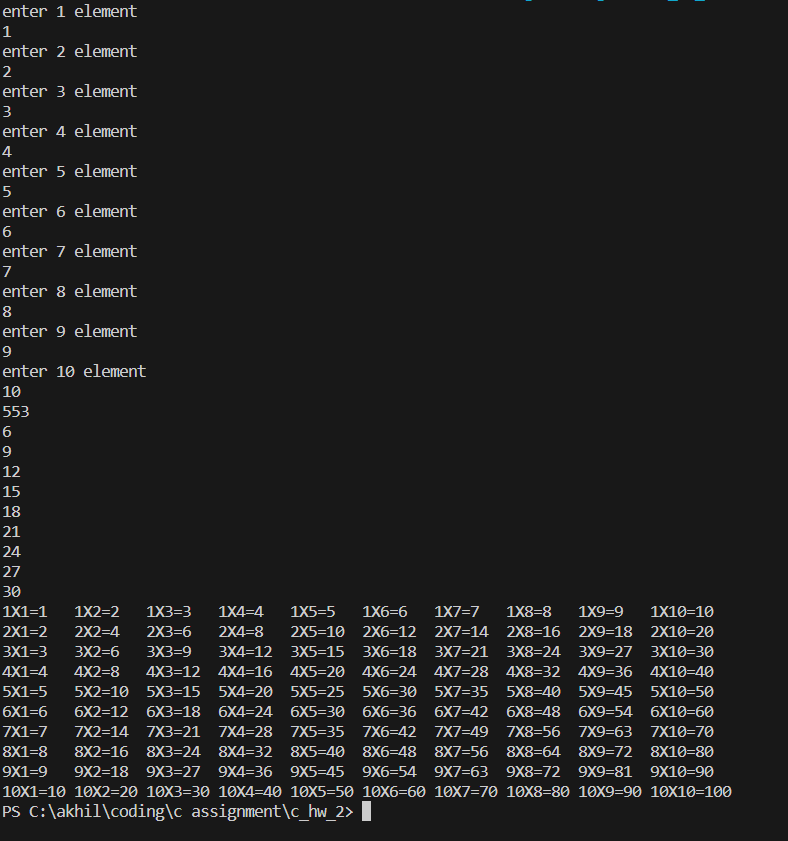
        }

        printf("\n");

    }

return 0;

}



1. Write a C program to take 10 elements in the array and print all the values in reverse order without copying in another array.[Don't use any extra array]

#include<stdio.h>

void input\_arr(int arr[],int n){

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

    printf("enter %d element\n",i+1);

        scanf("%d",&arr[i]);

    }

}

void print\_arr\_rev(int arr[],int n){

    for(int i=n-1;i>-1;i--){

    printf("%d\n",arr[i]);

    }

    }

    int main(){

        int n;

        printf("enter n");

        scanf("%d",&n);

        int arr[n];

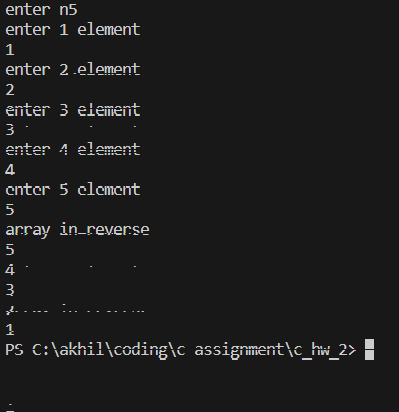
        input\_arr(arr,n);

        printf("array in reverse\n");

        print\_arr\_rev(arr,n);

return 0;

}



1. Write a program to count a total number of duplicate elements in an array.

Test Data : Input the number of elements to be stored in the array :3

Input 3 elements in the array :

element - 0 : 5

element - 1 : 1

element - 2 : 1

Expected Output : Total number of duplicate elements found in the array is : 1

#include <stdio.h>

int countDuplicates(int arr[], int size) {

    int duplicateCount = 0;

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

        for (int j = i + 1; j < size; j++) {

            if (arr[i] == arr[j]) {

                duplicateCount++;

                break;

            }

        }

    }

    return duplicateCount;

}

int main() {

    int n;

    printf("Input the number of elements to be stored in the array: ");

    scanf("%d", &n);

    int array[n];

    for (int i = 0; i <n; i++) {

        printf("element - %d : ", i);

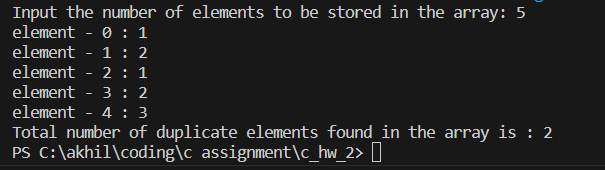
        scanf("%d", &array[i]);

    }

    printf("Total number of duplicate elements found in the array is : %d\n", countDuplicates(array, n));

    return 0;

}



1. Write a program to print all unique elements in an array.

Test Data : Input the number of elements to be stored in the array :3

Input 3 elements in the array :

element - 0 : 1

element - 1 : 5

element - 2 : 1

Expected Output : The unique elements found in the array are : 5

#include<stdio.h>

void input\_arr(int arr[],int n){

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

    printf("enter %d element\n",i+1);

        scanf("%d",&arr[i]);  }}

void print\_arr\_unique(int arr[],int n){

    for(int i=0;i<n;i++){

        int unique=1;

        for(int j=0;j<n;j++){

            if(i!=j&&arr[i]==arr[j]){

            unique=0;  break;}}

        if(unique){printf("%d\n",arr[i]);  }    }}

    int main(){

        int n;

        printf("enter no. of element in array");

        scanf("%d",&n);

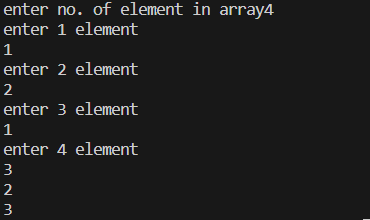
        int arr[n];

        input\_arr(arr,n);

print\_arr\_unique(arr,n);

return 0;

}



1. Use a single-subscripted array to solve the following problem.

Read in 20 numbers, each of which is between 10 and 100, inclusive. As each number is read, print it only if it‘s not a duplicate of a number already read. Provide for the ―worst case in which all 20 numbers are different. Use the smallest possible array to solve this problem.

//CAN BE DONE BY SAME CODE AS USED IN Q5.

1. A small airline has just purchased a computer for its new automated reservations system. The president has asked you to program the new system. You‘ll write a program to assign seats on each flight of the airline‘s only plane (capacity: 10 seats).

Your program should display the following menu of alternatives:

Please type 1 for "first class"

Please type 2 for "economy"

If the person types 1, then your program should assign a seat in the first class section (seats 1–5). If the person types 2, then your program should assign a seat in the economy section (seats 6–10). Your program should then print a boarding pass indicating the person's seat number and whether it‘s in the first class or economy section of the plane.

Use a single-subscripted array to represent the seating chart of the plane. Initialize all the elements of the array to 0 to indicate that all seats are empty. As each seat is assigned, set the corresponding element of the array to 1 to indicate that the seat is no longer available.

Your program should, of course, never assign a seat that has already been assigned. When the first class section is full, your program should ask the person if it‘s acceptable to be placed in the economy section (and vice versa). If yes, then make the appropriate seat assignment. If no, then

print the message "Next flight leaves in 3 hours."

#include <stdio.h>

int main() {

    int seats[10] = {0};

    int choice, seat, section;

    do {

        printf("Please type 1 for \"first class\"\n");

        printf("Please type 2 for \"economy\"\n");

        scanf("%d", &choice);

        if (choice == 1 || choice == 2) {

            if (choice == 1) {

                for (seat = 0; seat < 5; seat++) {

                    if (seats[seat] == 0) {

                        seats[seat] = 1;

                        section = 1;

                        break;

                    }

                }

            } else {

                for (seat = 5; seat < 10; seat++) {

                    if (seats[seat] == 0) {

                        seats[seat] = 1;

                        section = 2;

                        break;

                    }

                }

            }

            if (seat < 10) {

                printf("Boarding Pass\n");

                printf("Seat Number: %d\n", seat + 1);

                printf("Section: %s class\n", (section == 1) ? "First" : "Economy");

            } else {

                printf("The section is full. Would you like to be placed in the other section? (1 for yes, 0 for no): ");

                int switchSection;

                scanf("%d", &switchSection);

                if (switchSection == 1) {

                    choice = (choice == 1) ? 2 : 1;

                } else {

                    printf("Next flight leaves in 3 hours.\n");

                    break;

                }

            }

        } else {

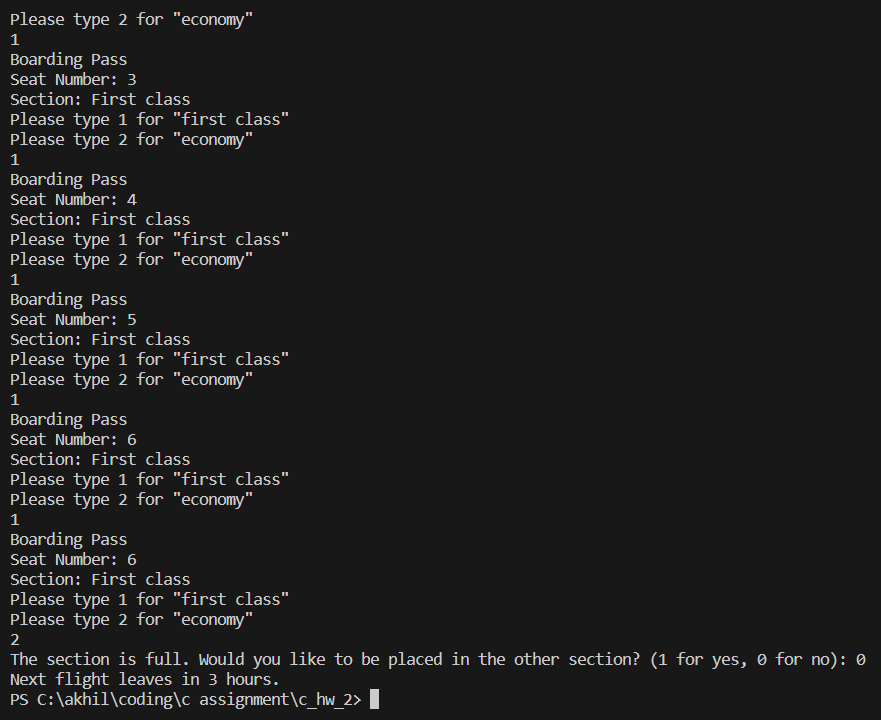
            printf("Invalid choice. Please type 1 for \"first class\" or 2 for \"economy\".\n");

        }

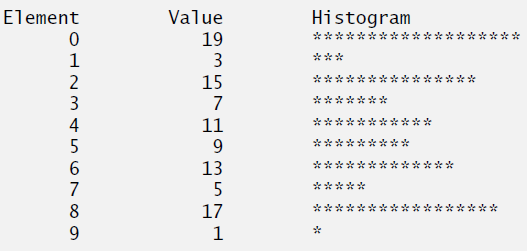
    } while (1);

    return 0;

}



1. Reads numbers from an array and graphs the information in the form of a bar chart or histogram—each number is printed, then a bar consisting of that many asterisks is printed beside the number. For Example:



#include<stdio.h>

void input\_arr(int arr[],int n){

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

    printf("enter %d element\n",i+1);

        scanf("%d",&arr[i]);

    }

}

void print\_arr\_bar(int arr[],int n){

    for(int j=0;j<n;j++){

printf("%d\t",j);

    for(int i=0;i<arr[j];i++){

    printf("\*");

    }

    printf("\n");}

}

    int main(){

        int n;

        printf("enter no. of element in array");

        scanf("%d",&n);

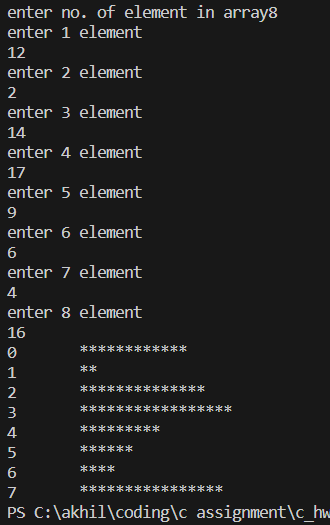
        int arr[n];

        input\_arr(arr,n);

        print\_arr\_bar(arr,n);

return 0;

}



1. Write a program to add two 6 x 6 matrices.

#include<stdio.h>

void in\_matrix(int arr[][6],int n){

    for(int i=0;i<n;i++)

    {for(int j=0;j<n;j++)

    {

        printf("enter %d,%d element",i,j);

        scanf("%d",&arr[i][j]);

    }}

}

void print\_matrix(int arr[][6],int n){

    for(int i=0;i<n;i++)

    {for(int j=0;j<n;j++)

    {

        printf("%d\t",arr[i][j]);

    }

    printf("\n");}

}

void matrix\_add(int arr1[][6],int arr2[][6],int n){

    for(int i=0;i<n;i++ ){

        for(int j=0;j<n;j++){

            printf("%d\t",arr1[i][j]+arr2[i][j]);

        }

        printf("\n");

    }

}

int main(){

    int n=6;

    printf("enter matrix 1\n");

    int arr1[6][6];

    in\_matrix(arr1,n);

    printf("matrix 1 is\n");

    print\_matrix(arr1,n);

    printf("enter 2nd matrix\n");

    int arr2[6][6];

    in\_matrix(arr2,n);

    printf("matrix 1 is\n");

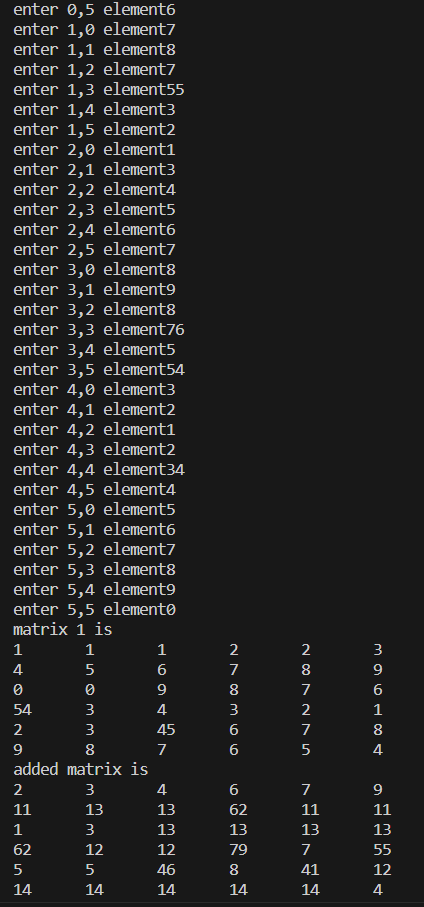
    print\_matrix(arr1,n);

    printf("added matrix is\n");

    matrix\_add(arr1,arr2,n);

    return 0;

}



1. Write a program to multiply any two 3 x 3 matrices.

#include<stdio.h>

void in\_matrix(int arr[][3],int n){

    for(int i=0;i<n;i++)

    {for(int j=0;j<n;j++)

    {

        printf("enter %d,%d element",i,j);

        scanf("%d",&arr[i][j]);

    }}

}

void print\_matrix(int arr[][3],int n){

    for(int i=0;i<n;i++)

    {for(int j=0;j<n;j++)

    {

        printf("%d\t",arr[i][j]);

    }

    printf("\n");}

}

void matrix\_multipli(int arr1[][3], int arr2[][3], int n) {

    for (int i = 0; i < n; i++) {

        for (int k = 0; k < n; k++) {

            int sum = 0;

            for (int j = 0; j < n; j++) {

                sum += arr1[i][j] \* arr2[j][k];

            }

            printf("%d\t", sum);

        }

        printf("\n");

    }

}

int main(){

     int n=3;

    printf("enter matrix 1\n");

    int arr1[3][3];

    in\_matrix(arr1,n);

    printf("enter 2nd matrix\n");

    int arr2[3][3];

    in\_matrix(arr2,n);

    printf("matrix 1 is\n");

    print\_matrix(arr1,n);

    printf("matrix 2 is\n");

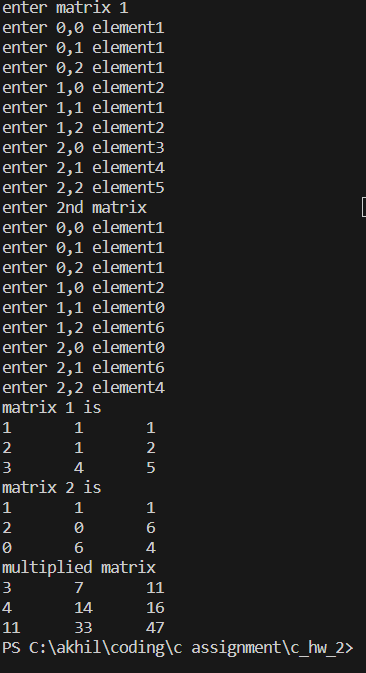
    print\_matrix(arr2,n);

    printf("multiplied matrix \n");

    matrix\_multipli(arr1,arr2,n);

    return 0;

}



1. Write a program to check whether a given matrix is an identity matrix.

#include <stdio.h>

void in\_matrix(int arr[][3],int n){

    for(int i=0;i<n;i++)

    {for(int j=0;j<n;j++)

    {

        printf("enter %d,%d element",i,j);

        scanf("%d",&arr[i][j]);

    }}

}

void print\_matrix(int arr[][3],int n){

    for(int i=0;i<n;i++)

    {for(int j=0;j<n;j++)

    {

        printf("%d\t",arr[i][j]);

    }

    printf("\n");}

}

int isIdentityMatrix(int mat[][3], int n) {

    if (n != 3) {

        return 0;

    }

    for (int i = 0; i < n; i++) {

        for (int j = 0; j < n; j++) {

            if ((i == j && mat[i][j] != 1) || (i != j && mat[i][j] != 0)) {

                return 0;

            }

        }

    }

    return 1;

}

int main() {

     int n=3;

    printf("enter matrix 1\n");

    int arr1[3][3];

    in\_matrix(arr1,n);

    printf("matrix 1 is\n");

    print\_matrix(arr1,n);

if(isIdentityMatrix(arr1,n)){

printf("matrix is identity\n");

}

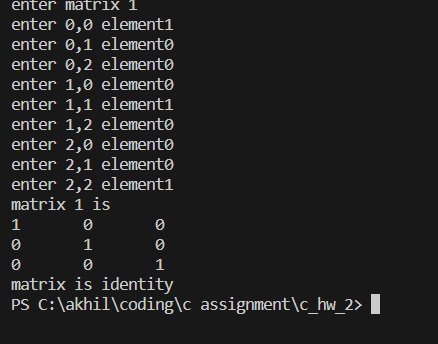
else{

    printf("matrix is mot an identity matrix\n");

}

    return 0;

}



1. Write a program to count the frequency of each element of an array.

Test Data :

Input the number of elements to be stored in the array :3

Input 3 elements in the array :

element - 0 : 25

element - 1 : 12

element - 2 : 43

Expected Output :

The frequency of all elements of an array :

25 occurs 1 times

12 occurs 1 times

43 occurs 1 times

Write the above program using only arrays, don't use any extra variable, use only arrays. You are free to use two arrays.

#include <stdio.h>

void input\_arr(int arr[], int n)

{

    for (int i = 0; i < n; i++)

    {

        printf("enter %d element\n", i + 1);

        scanf("%d", &arr[i]);

    }

}

void print\_arr(int arr[], int n)

{

    for (int i = 0; i < n; i++)

    {

        printf("%d\t", arr[i]);

    }

    printf("\n");

}

void frequency\_arr(int arr[], int n)

{

    int count[n];

    for (int i = 0; i < n; i++)

    {

        count[i] = 0;

    }

    for (int i = 0; i < n; i++)

    {

        for (int j = 0; j < n; j++)

        {

            if (arr[i] == arr[j])

            {

                count[i]++;

            }

        }

    }

    for (int i = 0; i < n; i++)

    {

        int flag = 1;

        for (int j = i-1; j > -1; j--)

        {

            if (arr[j] == arr[i])

            {

                flag = 0;

                break;

            }

        }

        if (flag)

        {

            {

                printf("frequency of %d element is: %d\n", arr[i], count[i]);

            }

        }

    }

}

int main()

{

    int n;

    printf("enter size of array\n");

    scanf("%d", &n);

    int arr[n];

    input\_arr(arr, n);

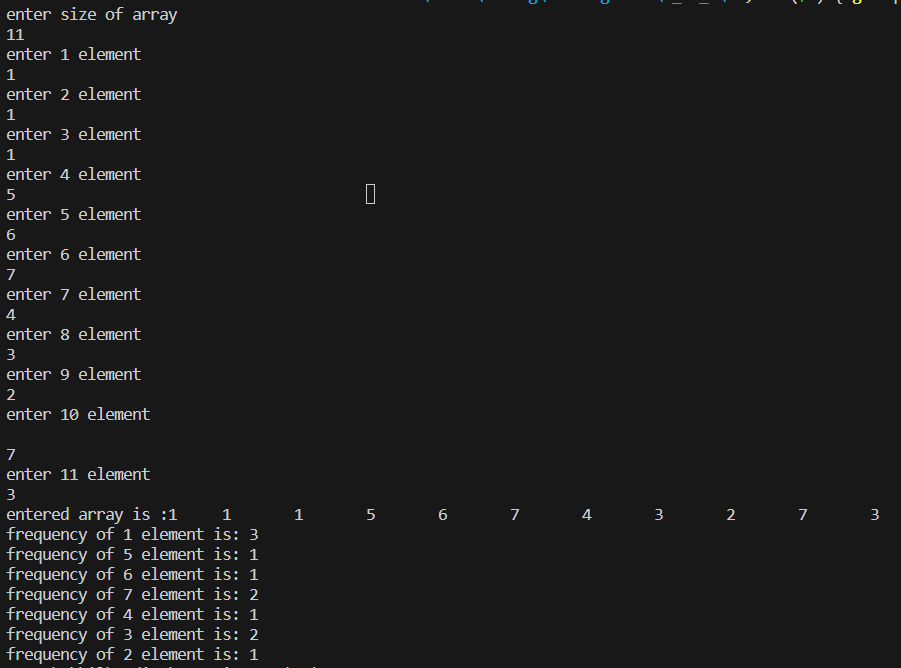
    printf("entered array is :");

    print\_arr(arr, n);

    frequency\_arr(arr, n);

    return 0;

}



1. Write a program to insert New value in the array (sorted list )..

Test Data :

Input the size of array : 3

Input 3 elements in the array in ascending order:

element - 0 : 5

element - 1 : 7

element - 2 : 9

Input the value to be inserted : 8

Expected Output :

The exist array list is : 5 7 9

After Insert the list is : 5 7 8 9

#include <stdio.h>

void input\_arr(int arr[], int n)

{

    for (int i = 0; i < n; i++)

    {

        printf("enter %d element\n", i + 1);

        scanf("%d", &arr[i]);

    }

}

void print\_arr(int arr[], int n)

{

    for (int i = 0; i < n; i++)

    {

        printf("%d\t", arr[i]);

    }

    printf("\n");

}

int index\_sorted\_arr(int arr[],int n,int a){

    int start=0;

    int end=n-1;

    int mid=(start+end)/2;

    while(start<=end){

        if(arr[mid]==a){

            return mid;

        }

        else if(arr[mid]<a){

            start=mid+1;

        }

        else{

            end=mid-1;

        }

        mid=(start+end)/2;

    }

    return mid;

}

void insert\_sorted\_arr(int arr[],int inserted\_arr[],int n,int a)

{

    int index=index\_sorted\_arr(arr,n,a);

    for(int i=0;i<=index;i++){

        inserted\_arr[i]=arr[i];

    }

    inserted\_arr[index+1]=a;

    for(int i=index+1;i<n;i++){

        inserted\_arr[i+1]=arr[i];

    }

}

int main()

{

    int n;

    printf("enter size of array\n");

    scanf("%d", &n);

    int arr[n];

    printf("enter elements of array in ascending orde\n");

    input\_arr(arr, n);

    printf("entered array is :");

    print\_arr(arr, n);

printf("enter the element you wanna insert\n");

int a;

scanf("%d",&a);

int inserted\_arr[n+1];

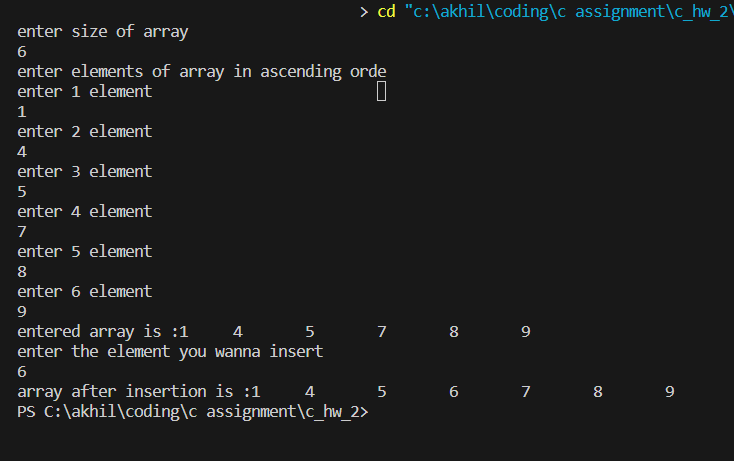
insert\_sorted\_arr(arr,inserted\_arr,n,a);

 printf("array after insertion is :");

    print\_arr(inserted\_arr, n+1);

    return 0;

}



1. Write a program to delete an element at desired position from an array.

Test Data : Input the size of array : 5

Input 5 elements in the array in ascending order:

element - 0 : 1

element - 1 : 2

element - 2 : 3

element - 3 : 4

element - 4 : 5

Input the position where to delete: 3

Expected Output :

The new list is : 1 2 4 5

#include <stdio.h>

void input\_arr(int arr[], int n)

{

    for (int i = 0; i < n; i++)

    {

        printf("enter %d element\n", i + 1);

        scanf("%d", &arr[i]);

    }

}

void print\_arr(int arr[], int n)

{

    for (int i = 0; i < n; i++)

    {

        printf("%d\t", arr[i]);

    }

    printf("\n");

}

void pop\_arr(int arr[],int n,int a){

    for(int i=a;i<n;i++){

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

    }

}

int main()

{

    int n;

    printf("enter size of array\n");

    scanf("%d", &n);

    int arr[n];

    printf("enter elements of array\n");

    input\_arr(arr, n);

    printf("entered array is :");

    print\_arr(arr, n);

printf("enter the index you wanna delete\n");

int a;

scanf("%d",&a);

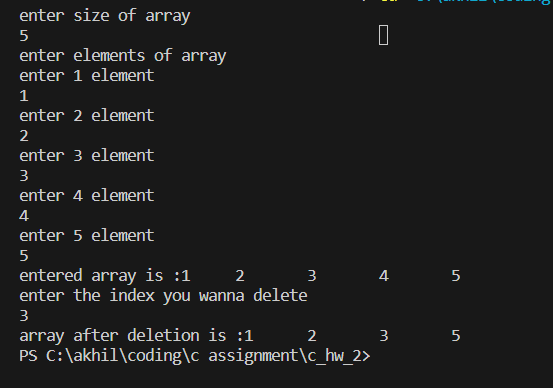
pop\_arr(arr,n,a);

 printf("array after deletion is :");

    print\_arr(arr, n-1);

    return 0;

}



1. Write a program to find transpose of a given matrix.

Test Data :

Input the rows and columns of the matrix : 2 2

Input elements in the first matrix :

element - [0][0] : 1

element - [0][1] : 2

element - [1][0] : 3

element - [1][1] : 4

Expected Output :

The matrix is :

1 2

3 4

The transpose of a matrix is :

1 3

2 4

#include<stdio.h>

void in\_matrix(int arr[][3],int n){

    for(int i=0;i<n;i++)

    {for(int j=0;j<n;j++)

    {

        printf("enter %d,%d element",i,j);

        scanf("%d",&arr[i][j]);

    }}

}

void print\_matrix(int arr[][3],int n){

    for(int i=0;i<n;i++)

    {for(int j=0;j<n;j++)

    {

        printf("%d\t",arr[i][j]);

    }

    printf("\n");}

}

void swap(int \*a,int \*b){

    \*a=\*a^\*b;

    \*b=\*a^\*b;

    \*a=\*a^\*b;

}

void matrix\_transpose(int arr1[][3], int n) {

    for (int i = 0; i < n; i++) {

            for (int j = 0; j < i; j++) {

             swap(&arr1[i][j],&arr1[j][i]);

        }

    }

}

int main(){

     int n=3;

    printf("enter matrix \n");

    int arr1[3][3];

    in\_matrix(arr1,n);

    printf("matrix is\n");

    print\_matrix(arr1,n);

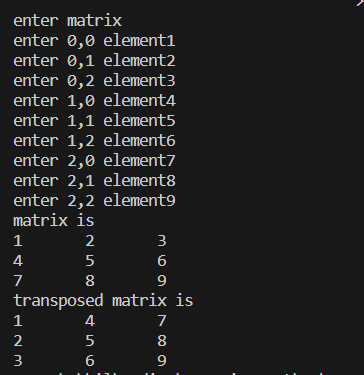
   matrix\_transpose(arr1,n);

    printf("transposed matrix is \n");

  print\_matrix(arr1,n);

    return 0;

}



1. Write a program to find sum of right diagonals of a matrix.

Test Data :

Input the size of the square matrix : 2

Input elements in the first matrix :

element - [0][0] : 1

element - [0][1] : 2

element - [1][0] : 3

element - [1][1] : 4

Expected Output :

The matrix is :

1 2

3 4

Addition of the right Diagonal elements is : 5

#include<stdio.h>

void in\_matrix(int arr[][3],int n){

    for(int i=0;i<n;i++)

    {for(int j=0;j<n;j++)

    {

        printf("enter %d,%d element",i,j);

        scanf("%d",&arr[i][j]);

    }}

}

void print\_matrix(int arr[][3],int n){

    for(int i=0;i<n;i++)

    {for(int j=0;j<n;j++)

    {

        printf("%d\t",arr[i][j]);

    }

    printf("\n");}

}

int sum\_right\_diagonal(int matrix[3][3]){

    int sum=0;

    for(int i=0;i<3;i++){

        sum+=matrix[i][i];

    }

    return sum;

}

int main(){

     int n=3;

    printf("enter matrix \n");

    int arr1[3][3];

    in\_matrix(arr1,n);

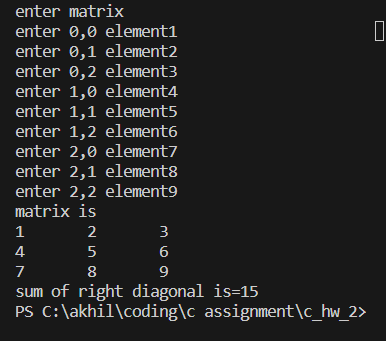
    printf("matrix is\n");

    print\_matrix(arr1,n);

   printf("sum of right diagonal is=%d",sum\_right\_diagonal(arr1));

    return 0;

}



1. Write a program to find sum of rows an columns of a Matrix.

Test Data : Input the size of the square matrix : 2

Input elements in the first matrix :

element - [0][0] : 5

element - [0][1] : 6

element - [1][0] : 7

element - [1][1] : 8

Expected Output :

The matrix is :

5 6

7 8

The sum or rows and columns of the matrix is :

5 6 11---- rows sum

7 8 15---- rows sum

12 14 --------------sum of columns

#include<stdio.h>

void in\_matrix(int arr[][3],int n){

    for(int i=0;i<n;i++)

    {for(int j=0;j<n;j++)

    {

        printf("enter %d,%d element",i,j);

        scanf("%d",&arr[i][j]);

    }}

}

void print\_matrix(int arr[][3],int n){

    for(int i=0;i<n;i++)

    {for(int j=0;j<n;j++)

    {

        printf("%d\t",arr[i][j]);

    }

    printf("\n");}

}

int main(){

     int n=3;

    printf("enter matrix \n");

    int arr1[3][3];

    in\_matrix(arr1,n);

    printf("matrix is\n");

    print\_matrix(arr1,n);

  for(int i=0;i<n;i++){

    int sum=0;

    for(int j=0;j<n;j++){

        sum+=arr1[i][j];

    }

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

  }

  for(int i=0;i<n;i++){

    int sum=0;

    for(int j=0;j<n;j++){

        sum+=arr1[j][i];

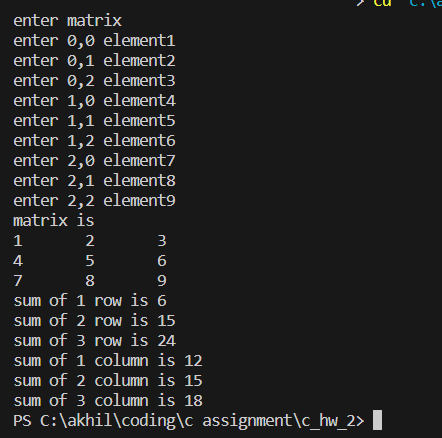
    }

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

  }

    return 0;

}



1. Write a program to print or display the lower triangular of a given matrix. For example if matrix is :

1 2 3

4 5 6

7 8 9

Setting zero in lower triangular matrix

1 2 3

0 5 6

0 0 9

#include<stdio.h>

void in\_matrix(int arr[][3],int n){

    for(int i=0;i<n;i++)

    {for(int j=0;j<n;j++)

    {

        printf("enter %d,%d element",i,j);

        scanf("%d",&arr[i][j]);

    }}

}

void print\_matrix(int arr[][3],int n){

    for(int i=0;i<n;i++)

    {for(int j=0;j<n;j++)

    {

        printf("%d\t",arr[i][j]);

    }

    printf("\n");}

}

void lower\_tri\_matrix(int arr[][3]){

    int n=3;

    for(int i=0;i<n;i++){

        for(int j=0;j<i;j++){

            arr[i][j]=0;

        }

    }

}

int main(){

     int n=3;

    printf("enter matrix \n");

    int arr1[3][3];

    in\_matrix(arr1,n);

    printf("matrix is\n");

    print\_matrix(arr1,n);

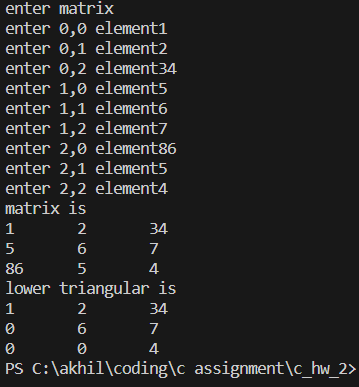
  printf("lower triangular is\n");

lower\_tri\_matrix(arr1);

 print\_matrix(arr1,n);

    return 0;

}



1. Write a program to calculate the determinant of a 3 x 3 matrix.

#include <stdio.h>

int calculateDeterminant(int mat[3][3]) {

    return mat[0][0] \* (mat[1][1] \* mat[2][2] - mat[2][1] \* mat[1][2]) -

           mat[0][1] \* (mat[1][0] \* mat[2][2] - mat[2][0] \* mat[1][2]) +

           mat[0][2] \* (mat[1][0] \* mat[2][1] - mat[2][0] \* mat[1][1]);

}

int main() {

    int mat[3][3];

    printf("Enter the elements of the 3x3 matrix:\n");

    for (int i = 0; i < 3; i++) {

        for (int j = 0; j < 3; j++) {

            printf("Element at position (%d, %d): ", i + 1, j + 1);

            scanf("%d", &mat[i][j]);

        }

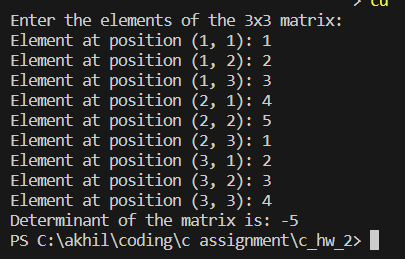
    }

    int determinant = calculateDeterminant(mat);

    printf("Determinant of the matrix is: %d\n", determinant);

    return 0;

}



1. Write a program to accept two matrices and check whether they are equal(iff they can be compared).

#include <stdio.h>

void in\_matrix(int arr[][3],int n){

    for(int i=0;i<n;i++)

    {for(int j=0;j<n;j++)

    {

        printf("enter %d,%d element",i,j);

        scanf("%d",&arr[i][j]);

    }}

}

void print\_matrix(int arr[][3],int n){

    for(int i=0;i<n;i++)

    {for(int j=0;j<n;j++)

    {

        printf("%d\t",arr[i][j]);

    }

    printf("\n");}

}

int MatricesEqual(int mat1[][3], int mat2[][3], int rows, int cols) {

    if (rows != 3 || cols != 3) {

        return 0;

    }

    for (int i = 0; i < rows; i++) {

        for (int j = 0; j < cols; j++) {

            if (mat1[i][j] != mat2[i][j]) {

                return 0;

            }

        }

    }

    return 1; }

int main() {

    int mat1[3][3];

    int mat2[3][3] ;

printf("enter matrix 1\n");

in\_matrix(mat1,3);

printf("enter matrix 2\n");

in\_matrix(mat2,3);

printf("matrix 1 is\n");

print\_matrix(mat1,3);

printf("matrix 2 is\n");

print\_matrix(mat2,3);

    if (MatricesEqual(mat1, mat2, 3, 3)) {

        printf("The matrices are equal.\n");

    } else {

        printf("The matrices are not equal.\n");

    }

    return 0;

}

