Assignment 5: Array

Submitted By: U19CS012(D-12)

1. WAP to add corresponding elements of two 1-Dimensional array and store in third array, also

Calculate the average of the 2 array.

Code:

*#include* <stdio.h>

int max(int a, int b)

{

*if* (a >= b)

*return* a;

*else*

*return* b;

}

void main()

{

    int n, m;

    printf("\nEnter the Size of Two Input Arrays(less than 20)(size1 and size2) : ");

    scanf("%d %d", &n, &m);

    int arr1[21] = {0};

    int arr2[21] = {0};

    int arr3[21] = {0};

    printf("\nEnter the Input for First Array : ");

*for* (int i = 0; i < n; i++)

    {

        scanf("%d", &arr1[i]);

    }

    printf("\nEnter the Input for Second Array : ");

*for* (int j = 0; j < n; j++)

    {

        scanf("%d", &arr2[j]);

    }

    int sum = 0;

*for* (int k = 0; k < max(n, m); ++k)

    {

        arr3[k] = arr1[k] + arr2[k];

        sum += arr3[k];

    }

    int avg = sum / (max(n, m));

    printf("\nThe Final Array is : ");

*for* (int i = 0; i < max(n, m); ++i)

    {

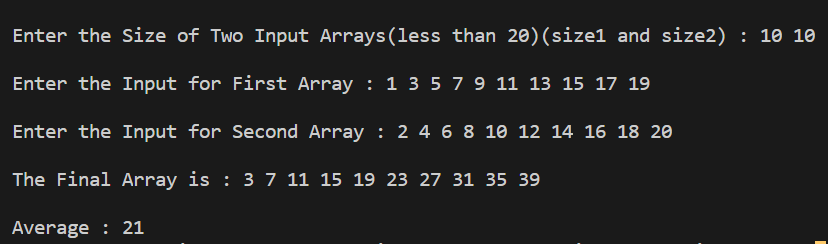
        printf("%d ", arr3[i]);

    }

    printf("\n\nAverage : %d\n", avg);

}

Output:



2. WAP to count total no of odd and even numbers from the 1-D array.

Code:

*#include* <stdio.h>

void main()

{

    int n, odd = 0, even = 0;

    printf("\nEnter the Size of Input Array(less than 20): ");

    scanf("%d", &n);

    int arr[21] = {0};

    printf("\nEnter the Values of Array : ");

*for* (int i = 0; i < n; i++)

    {

        scanf("%d", &arr[i]);

*if* (arr[i] & 1)

            odd += 1;

*else*

            even += 1;

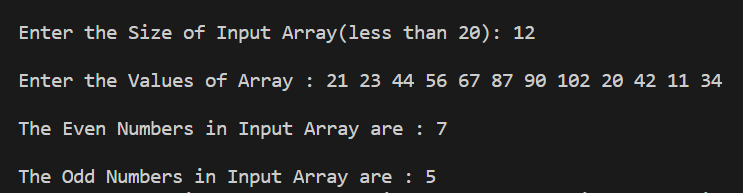
    }

    printf("\nThe Even Numbers in Input Array are : %d\n", even);

    printf("\nThe Odd Numbers in Input Array are : %d\n", odd);

}

Output:



3. WAP to sort an array in descending order.

Code:

*#include* <stdio.h>

void main()

{

    int n, odd = 0, even = 0;

    printf("\nEnter the Size of Input Array(less than 20): ");

    scanf("%d", &n);

    int arr[21] = {0};

    printf("\nEnter the Values of Array : ");

*for* (int i = 0; i < n; i++)

    {

        scanf("%d", &arr[i]);

    }

    int temp = 0;

*for* (int i = 0; i < n; ++i)

    {

*for* (int j = i + 1; j < n; ++j)

        {

*if* (arr[i] < arr[j])

            {

                temp = arr[i];

                arr[i] = arr[j];

                arr[j] = temp;

            }

        }

    }

    printf("\nArray after Sorting :");

*for* (int i = 0; i < n; i++)

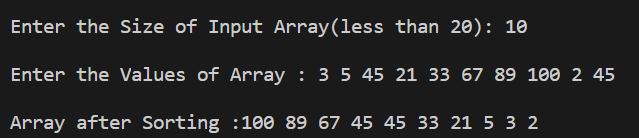
    {

        printf("%d ", arr[i]);

    }

}

Output:



4. WAP to exchange the smallest and largest values in 1-D array.

Code:

*#include* <stdio.h>

int max(int a, int b)

{

*if* (a >= b)

*return* a;

*else*

*return* b;

}

int min(int a, int b)

{

*if* (a <= b)

*return* a;

*else*

*return* b;

}

void main()

{

    int n, odd = 0, even = 0;

    printf("\nEnter the Size of Input Array(less than 20): ");

    scanf("%d", &n);

    int arr[21] = {0};

    int max1 = 0, min1 = 100000;

    printf("\nEnter the Values of Array(less than 100000) : ");

*for* (int i = 0; i < n; i++)

    {

        scanf("%d", &arr[i]);

        max1 = max(arr[i], max1);

        min1 = min(arr[i], min1);

    }

    printf("\nThe Maximum Value in Array is : %d\n", max1);

    printf("\nThe Minimum Value in Array is : %d\n", min1);

    int min\_index = -1, max\_index = -1;

*for* (int j = 0; j < n; j++)

    {

*if* (arr[j] == max1)

            max\_index = j;

*else*

        {

*if* (arr[j] == min1)

                min\_index = j;

        }

    }

    int temp = arr[max\_index];

    arr[max\_index] = arr[min\_index];

    arr[min\_index] = temp;

    printf("\nArray after Swapping : ");

*for* (int k = 0; k < n; ++k)

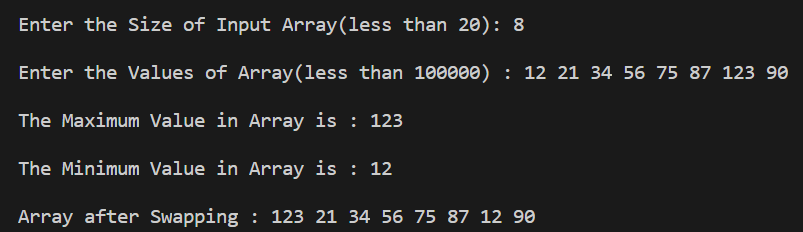
    {

        printf("%d ", arr[k]);

    }

}

Output:



5. WAP to delete an element of an array given by the user.

Code:

*#include* <stdio.h>

void main()

{

    int n;

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

    scanf("%d", &n);

    int arr[100], pos, i;

    printf("\nEnter %d elements of the Array : ", n);

*for* (i = 0; i < n; i++)

        scanf("%d", &arr[i]);

    int del\_element, flag = 1;

    printf("\nEnter the Element to be deleted : ");

    scanf("%d", &del\_element);

*for* (int i = 0; i < n; i++)

    {

*if* (arr[i] == del\_element)

        {

            pos = i;

            flag = 0;

        }

    }

*if* (flag)

    {

        printf("\nElement to be Deleted Not Found.\n");

        printf("\nFinal Array After element Deletion:");

*for* (i = 0; i < n; i++)

            printf("%d ", arr[i]);

    }

*else*

    {

*for* (i = pos; i < n - 1; i++)

            arr[i] = arr[i + 1];

        printf("\nFinal Array After element Deletion:");

*for* (i = 0; i < n - 1; i++)

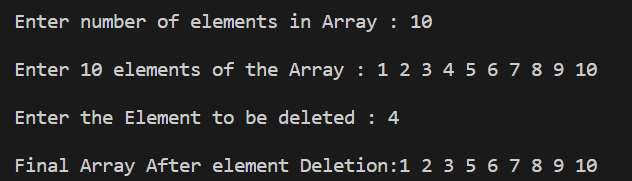
            printf("%d ", arr[i]);

    }

    printf("\n");

}

Output:



6. WAP to insert an element in an array specified by the user.

Code:

*#include* <stdio.h>

void main()

{

    int n;

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

    scanf("%d", &n);

    int arr[100], pos, i;

    printf("\nEnter %d elements of the Array : ", n);

*for* (i = 0; i < n; i++)

        scanf("%d", &arr[i]);

    int ins\_element;

    printf("\nEnter the Element to be Inserted : ");

    scanf("%d", &ins\_element);

    printf("\nEnter the Position of Element : ");

    scanf("%d", &pos);

*for* (i = n; i >= pos; i--)

    {

        arr[i] = arr[i - 1];

    }

    arr[pos - 1] = ins\_element;

    printf("\nFinal Array After element Deletion:");

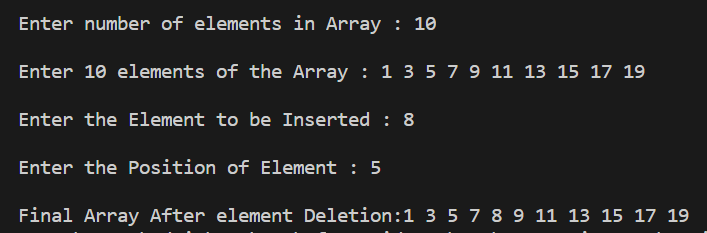
*for* (i = 0; i < n + 1; i++)

        printf("%d ", arr[i]);

    printf("\n");

}

Output:



7. Given an array arr[] of size N. The task is to find the sum of arr[i] % arr[j] for all valid pairs.

Answer can be large. So, output answer modulo 1000000007

Input: arr[] = {1, 2, 3}

Output: 5

(1 % 1) + (1 % 2) + (1 % 3) + (2 % 1) + (2 % 2)

+ (2 % 3) + (3 % 1) + (3 % 2) + (3 % 3) = 5

Code:

*#include* <stdio.h>

void main()

{

    int n;

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

    scanf("%d", &n);

    int arr[100],i;

    printf("\nEnter %d elements of the Array : ", n);

*for* (i = 0; i < n; i++)

        scanf("%d", &arr[i]);

    int sum = 0;

*for*(int j=0;j<n;j++)

    {

*for*(int k=0;k<n;k++)

        {

            sum += arr[j]%arr[k];

        }

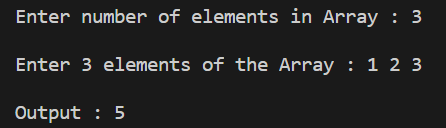
    }

    printf("\nOutput : %d",sum%1000000007);

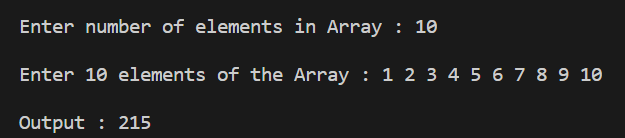
}

Output:

SAMPLE CASE:



TEST CASE:



8. WAP to perform matrix multiplication of 3\*3 matrixes.

Code:

*#include* <stdio.h>

void main()

{

    int a[3][3], b[3][3], c[3][3];

    printf("\nEnter the Elements for First Matrix : \n");

    int i, j, k;

*for* (i = 0; i < 3; ++i)

    {

*for* (j = 0; j < 3; ++j)

        {

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

            scanf("%d", &a[i][j]);

        }

    }

    printf("\nEnter the Elements for Second Matrix : \n");

*for* (i = 0; i < 3; ++i)

    {

*for* (j = 0; j < 3; ++j)

        {

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

            scanf("%d", &b[i][j]);

        }

    }

*for* (i = 0; i < 3; ++i)

    {

*for* (j = 0; j < 3; ++j)

        {

*for* (k = 0; k < 3; ++k)

            {

                c[i][j] = a[i][k] \* b[k][j];

            }

        }

    }

    printf("\nMatrix A : \n");

*for* (i = 0; i < 3; ++i)

    {

*for* (j = 0; j < 3; ++j)

        {

            printf("%d ", a[i][j]);

        }

        printf("\n");

    }

    printf("\nMatrix B : \n");

*for* (i = 0; i < 3; ++i)

    {

*for* (j = 0; j < 3; ++j)

        {

            printf(" %d ", b[i][j]);

        }

        printf("\n");

    }

    printf("\nFinal Resultant Matrix (After Multiplication) : \n");

*for* (i = 0; i < 3; ++i)

    {

*for* (j = 0; j < 3; ++j)

        {

            printf(" %d ", c[i][j]);

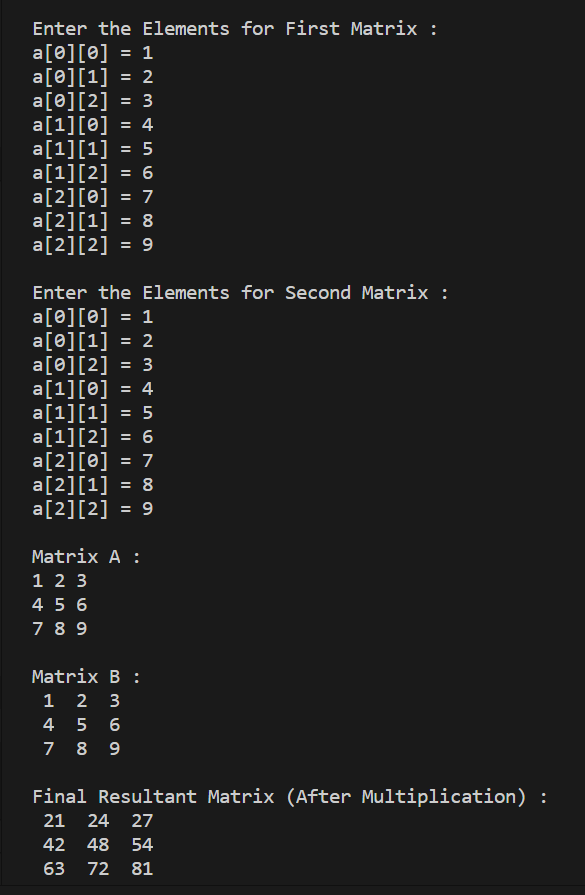
        }

        printf("\n");

    }

}

Output:



9. Given an array of integers of size n, find out if the numbers in the array appear in a

Palindromic order. A palindrome is a sequence that reads the same when you flip it. For

Example, 121 is a palindrome, 3 is a palindrome, and 234432 is also a palindrome

Code:

*#include* <stdio.h>

int Ispalindrome(int n)

{

    int k = n;

*// Single Digit Number*

*if* (n / 10 == 0)

*return* 1;

*// Find Reverse Number*

    int rev\_num = 0;

*while* (k > 0)

    {

        rev\_num = rev\_num \* 10 + k % 10;

        k = k / 10;

    }

*if* (rev\_num == n)

*return* 1;

*else*

*return* 0;

}

void main()

{

    int n;

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

    scanf("%d", &n);

    int arr[100], i;

    printf("\nEnter %d elements of the Array : ", n);

    int palin\_count = 0;

*for* (i = 0; i < n; i++)

    {

        scanf("%d", &arr[i]);

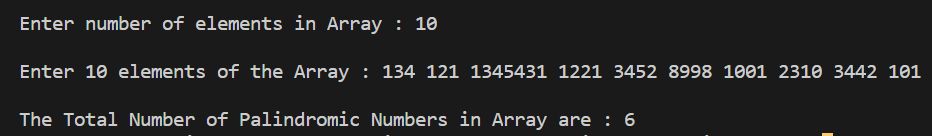
        palin\_count += Ispalindrome(arr[i]);

    }

    printf("\nThe Total Number of Palindromic Numbers in Array are : %d", palin\_count);

}

Output:



10. Given two sorted arrays of sizes m and n, write a program that merges the two into another

Array of size m + n such that this new array also remains sorted.

Code:

*#include* <stdio.h>

void mergeArrays(int arr1[], int arr2[], int n1, int n2, int arr3[])

{

    int i = 0, j = 0, k = 0;

*while* (i < n1 && j < n2) *//Array Traversal*

    {

*//The Smaller of two gets Stored in arr3*

*if* (arr1[i] < arr2[j])

            arr3[k++] = arr1[i++];

*else*

            arr3[k++] = arr2[j++];

    }

*while* (i < n1) *// Store remaining elements of first array*

        arr3[k++] = arr1[i++];

*while* (j < n2) *// Store remaining elements of second array*

        arr3[k++] = arr2[j++];

}

void main()

{

    int n, m;

    printf("\nEnter number of elements in Array 1 : ");

    scanf("%d", &n);

    int arr1[100];

    printf("\nEnter %d elements of the Array 1 : ", n);

*for* (int i = 0; i < n; i++)

        scanf("%d", &arr1[i]);

    printf("\nEnter number of elements in Array 2 : ");

    scanf("%d", &m);

    int arr2[100];

    printf("\nEnter %d elements of the Array 2 : ", m);

*for* (int i = 0; i < m; i++)

        scanf("%d", &arr2[i]);

    int arr3[n + m];

    mergeArrays(arr1, arr2, n, m, arr3);

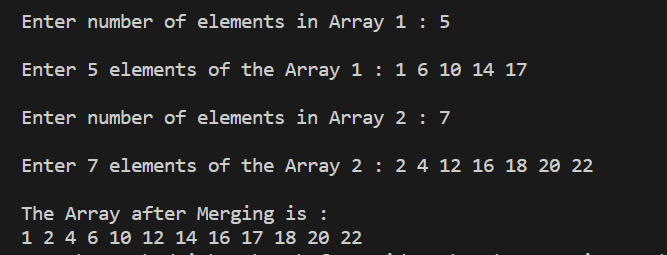
    printf("\nThe Array after Merging is : \n");

*for* (int j = 0; j < n + m; j++)

        printf("%d ", arr3[j]);

}

Output:



11. WAP to subtract 2-D Matrices.

Code:

*#include* <stdio.h>

void main()

{

    int a[3][3], b[3][3], c[3][3];

    printf("\nEnter the Elements for First Matrix : \n");

    int i, j, k;

*for* (i = 0; i < 3; ++i)

    {

*for* (j = 0; j < 3; ++j)

        {

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

            scanf("%d", &a[i][j]);

        }

    }

    printf("\nEnter the Elements for Second Matrix : \n");

*for* (i = 0; i < 3; ++i)

    {

*for* (j = 0; j < 3; ++j)

        {

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

            scanf("%d", &b[i][j]);

        }

    }

*for* (i = 0; i < 3; ++i)

    {

*for* (j = 0; j < 3; ++j)

        {

                c[i][j] = a[i][j] - b[i][j];

        }

    }

    printf("\nMatrix A : \n");

*for* (i = 0; i < 3; ++i)

    {

*for* (j = 0; j < 3; ++j)

        {

            printf("%d ", a[i][j]);

        }

        printf("\n");

    }

    printf("\nMatrix B : \n");

*for* (i = 0; i < 3; ++i)

    {

*for* (j = 0; j < 3; ++j)

        {

            printf(" %d ", b[i][j]);

        }

        printf("\n");

    }

    printf("\nFinal Resultant Matrix (After Subtraction) : \n");

*for* (i = 0; i < 3; ++i)

    {

*for* (j = 0; j < 3; ++j)

        {

            printf(" %d ", c[i][j]);

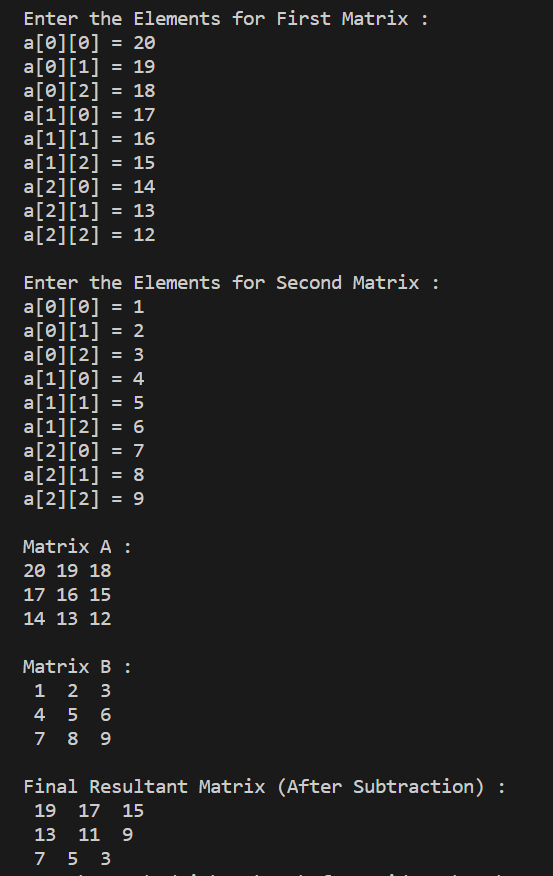
        }

        printf("\n");

    }

}

Output:



Submitted By:

Bhagya Rana

U19CS012 (D-12)(CSE,SVNIT)