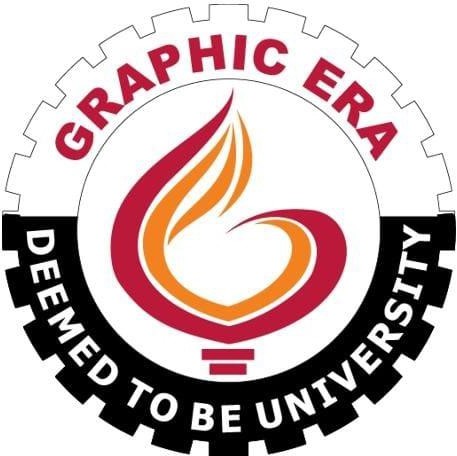
LAB FILE

C PROGRAMMING FILE GEU

BATCH

2023-2026

BCA Al AND DS

SUBMITTED BY:

TanishaTowm STUDENT ID: 231602087

SUBMITTED TO:

Mr. Rishi Kumar AE!. fr,gf,, CSIT GEU

INDEX:

|  |  |
| --- | --- |
| SERIAL NUMBER | TOPIC NAME |
| 1 | Write a program in C to read n number of values in an array and display them in reverse order. |
| 2 | Write a program in C to find the sum of all elements of the array |
| 3 | Write a program in C to copy the elements of one array into another array. |
| 4 | Write a program in C to count the total number of duplicate elements in an array. |
| 5 | Write a program in C to find the maximum and minimum elements in an array. |
| 6 | Write a C program to sort the elements of an array in descending order |
| 7 | Write a program in C to separate odd and even integers into separate arrays. |

|  |  |
| --- | --- |
| 8 | Write a program in C to merge two arrays of the same size sorted in descending/ascending order. |
| 9 | Write a program in C to merge two arrays of the same size sorted in descending order |
| 10 | WAP using Switch case: Consider two matrices of the size m and n. Implement matrix operation and display. Show these things in program 1) Read matrix elements and display 2) Matrix Multiplication and display 3) addition of matrix and display 4)Subtraction of Matrix and display 5)Transpose of Matrix and display. |

Write a program in C to read n number of values in an array and display them in reverse order:

CODE:

#include <stdio.h> int main() {

int n;

printf("Enter the number of values: "); scanf("%d", &n);

int arr[n];

printf("Enter %d values:\n" , n); for (int i = 0; i < n; i++) {

scanf("%d", &arr[i]);

}

printf("Values in reverse order:\n"); for (int i = n - 1; i >= 0; i--) {

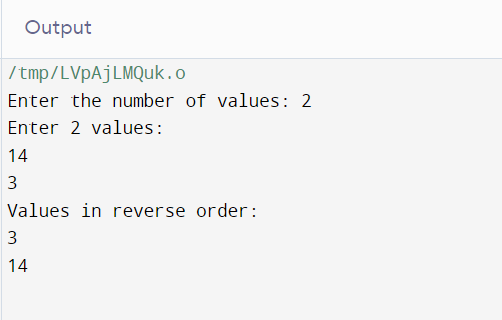
printf("%d\n", arr[i]);

}

return 0;

}

OUTPUT:



Write a program in C to find the sum of all elements of the array Code:

#include<stdio.h> int main()

{

int arr[100], size, i, sum = 0; printf("Enter array size\n"); scanf("%d",&size);

printf("Enter array elements\n"); for(i = 0; i < size; i++)

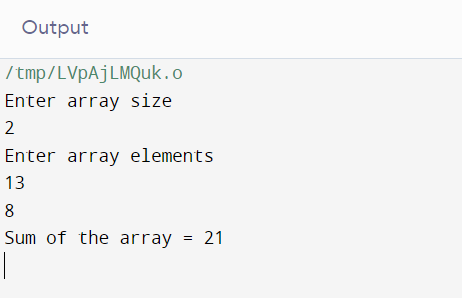
scanf("%d",&arr[i]); for(i = 0; i < size; i++)

sum = sum + arr[i];

printf("Sum of the array = %d\n",sum); return 0;

}

OUTPUT:



Write a program in C to copy the elements of one array into another array.

Code:

#include <stdio.h> int main() {

int n;

printf("Enter the number of elements in the array: "); scanf("%d", &n);

int sourceArray[n], destinationArray[n];

printf("Enter %d elements for the source array:\n", n); for (int i = 0; i < n; i++) {

scanf("%d", &sourceArray[i]);

}

for (int i = 0; i < n; i++) { destinationArray[i] = sourceArray[i];

}

printf("Elements in the destination array:\n"); for (int i = 0; i < n; i++) {

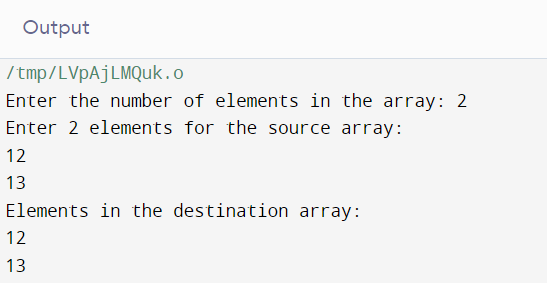
printf("%d\n", destinationArray[i]);

}

return 0;

}

# OUTPUT:



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

Code:

#include <stdio.h> int main() {

int n;

printf("Enter the number of elements in the array: "); scanf("%d", &n);

int arr[n];

printf("Enter %d elements:\n", n); for (int i = 0; i < n; i++) {

scanf("%d", &arr[i]);

}

int count = 0;

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

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

if (arr[i] == arr[j] && arr[i] != -1) { count++;

arr[j] = -1;

}

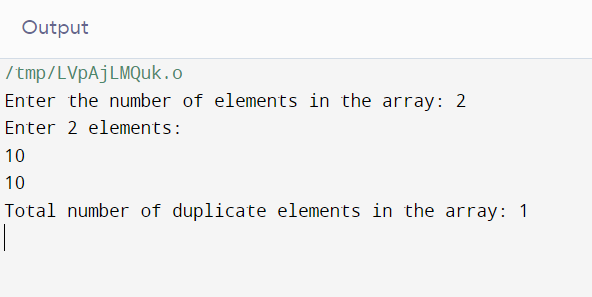
}

}

printf("Total number of duplicate elements in the array: %d\n", count); return 0;

}

# OUTPUT:



Write a program in C to find the maximum and minimum elements in an array.

Code:

#include <stdio.h> int main() {

int n;

printf("Enter the number of elements in the array: "); scanf("%d", &n);

if (n <= 0) {

printf("Invalid input. The array must have at least one element.\n"); return 1;

}

int arr[n];

printf("Enter %d elements:\n", n); for (int i = 0; i < n; i++) {

scanf("%d", &arr[i]);

}

int max = arr[0]; int min = arr[0];

for (int i = 1; i < n; i++) { if (arr[i] > max) {

max = arr[i];

}

if (arr[i] < min) { min = arr[i];

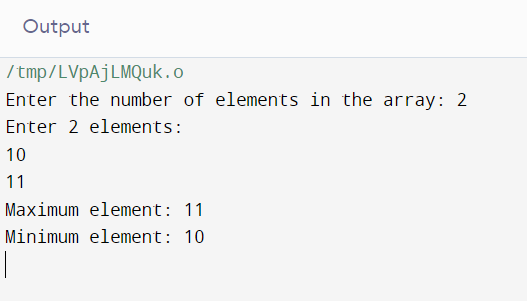
}

}

printf("Maximum element: %d\n", max); printf("Minimum element: %d\n", min); return 0;

}

# OUTPUT:



Write a C program to sort the elements of an array in descending order Code:

#include <stdio.h>

void bubbleSort(int arr[], int n) { int temp;

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

for (int j = 0; j < n - 1 - i; j++) { if (arr[j] < arr[j + 1]) {

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

}

}

}

}

int main() { int n;

printf("Enter the number of elements in the array: "); scanf("%d", &n);

if (n <= 0) {

printf("Invalid input. The array must have at least one element.\n");

return 1;

}

int arr[n];

printf("Enter %d elements:\n", n); for (int i = 0; i < n; i++) {

scanf("%d", &arr[i]);

}

bubbleSort(arr, n);

printf("Sorted array in descending order: "); for (int i = 0; i < n; i++) {

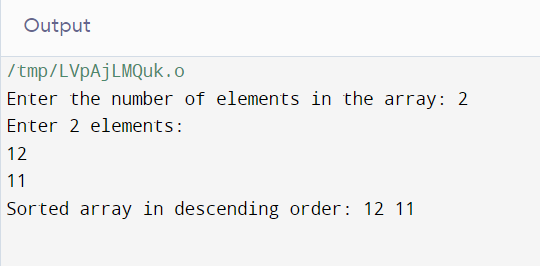
printf("%d ", arr[i]);

}

printf("\n"); return 0;

}

# OUTPUT:



Write a program in C to separate odd and even integers into separate arrays.

Code:

#include <stdio.h> int main() {

int n;

printf("Enter the number of elements in the array: "); scanf("%d", &n);

if (n <= 0) {

printf("Invalid input. The array must have at least one element.\n"); return 1;

}

int arr[n];

printf("Enter %d elements:\n", n); for (int i = 0; i < n; i++) {

scanf("%d", &arr[i]);

}

int evenArray[n], oddArray[n]; int evenCount = 0, oddCount = 0; for (int i = 0; i < n; i++) {

if (arr[i] % 2 == 0) {

evenArray[evenCount] = arr[i]; evenCount++;

} else {

oddArray[oddCount] = arr[i]; oddCount++;

}

}

printf("Even integers: ");

for (int i = 0; i < evenCount; i++) { printf("%d ", evenArray[i]);

}

printf("\n"); printf("Odd integers: ");

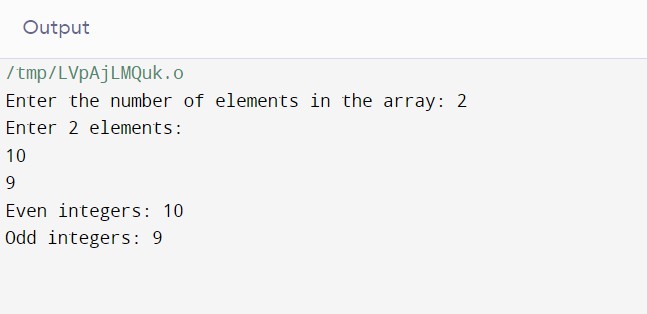
for (int i = 0; i < oddCount; i++) { printf("%d ", oddArray[i]);

}

printf("\n"); return 0;

}

# OUTPUT:



Write a program in C to merge two arrays of the same size sorted in descending/ascending order.

Code:

#include <stdio.h> int main() {

int n;

printf("Enter the number of elements in each array: "); scanf("%d", &n);

if (n <= 0) {

printf("Invalid input. Both arrays must have at least one element.\n");

return 1;

}

int arr1[n], arr2[n], mergedArray[2 \* n];

printf("Enter %d elements for the first array (ascending order):\n", n); for (int i = 0; i < n; i++) {

scanf("%d", &arr1[i]);

}

printf("Enter %d elements for the second array (ascending order):\n", n);

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

scanf("%d", &arr2[i]);

}

int i = 0, j = 0, k = 0; while (i < n && j < n) {

if (arr1[i] < arr2[j]) {

mergedArray[k] = arr1[i]; i++;

} else {

mergedArray[k] = arr2[j]; j++;

}

k++;

}

while (i < n) {

mergedArray[k] = arr1[i]; i++;

k++;

}

while (j < n) {

mergedArray[k] = arr2[j]; j++;

k++;

}

printf("Merged array (ascending order): "); for (int i = 0; i < 2 \* n; i++) {

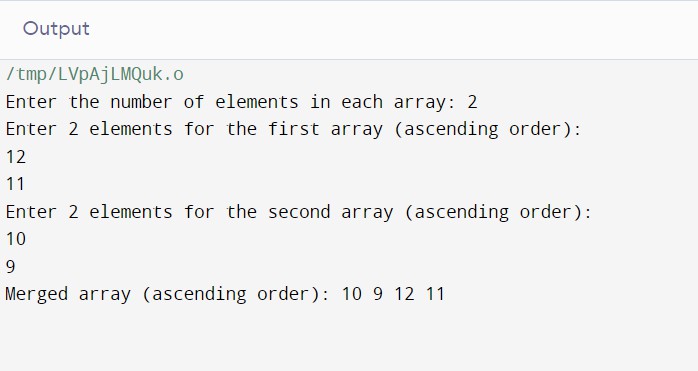
printf("%d ", mergedArray[i]);

}

printf("\n"); return 0;

}

# OUTPUT:



Write a program in C to merge two arrays of the same size sorted in descending order

Code:

#include <stdio.h> int main() {

int n;

printf("Enter the number of elements in each array: "); scanf("%d", &n);

if (n <= 0) {

printf("Invalid input. Both arrays must have at least one element.\n");

return 1;

}

int arr1[n], arr2[n], mergedArray[2 \* n];

printf("Enter %d elements for the first array (descending order):\n", n);

for (int i = 0; i < n; i++) { scanf("%d", &arr1[i]);

}

printf("Enter %d elements for the second array (descending order):\n", n);

for (int i = 0; i < n; i++) { scanf("%d", &arr2[i]);

}

int i = 0, j = 0, k = 0; while (i < n && j < n) {

if (arr1[i] > arr2[j]) {

mergedArray[k] = arr1[i]; i++;

} else {

mergedArray[k] = arr2[j]; j++;

}

k++;

}

while (i < n) {

mergedArray[k] = arr1[i]; i++;

k++;

}

while (j < n) {

mergedArray[k] = arr2[j]; j++;

k++;

}

printf("Merged array (descending order): "); for (int i = 0; i < 2 \* n; i++) {

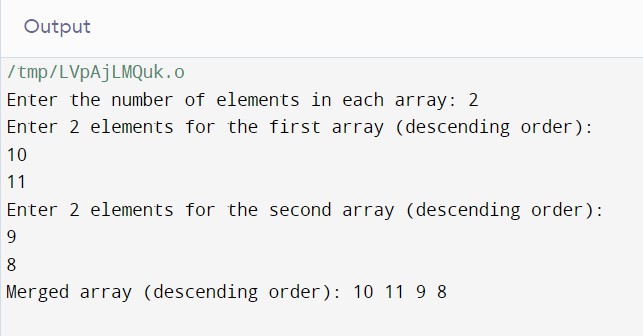
printf("%d ", mergedArray[i]);

}

printf("\n"); return 0;

}

# OUTPUT:



WAP using Switch case: Consider two matrices of the size m and n.

Implement matrix operation and display. Show these things in program

1) Read matrix elements and display 2) Matrix Multiplication and display 3) addition of matrix and display 4)Subtraction of Matrix and display 5)Transpose of Matrix and display.

Code:

#include <stdio.h>

void readMatrix(int matrix[][10], int m, int n) { printf("Enter the elements of the matrix:\n"); for (int i = 0; i < m; i++) {

for (int j = 0; j < n; j++) { scanf("%d", &matrix[i][j]);

}

}

}

void displayMatrix(int matrix[][10], int m, int n) { printf("Matrix:\n");

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

printf("%d\t", matrix[i][j]);

}

printf("\n");

}

}

void multiplyMatrices(int m1[][10], int m, int n, int m2[][10], int p, int q, int result[][10]) {

if (n != p) {

printf("Matrix multiplication is not possible.\n"); return;

}

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

result[i][j] = 0;

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

result[i][j] += m1[i][k] \* m2[k][j];

}

}

}

}

void addMatrices(int m1[][10], int m, int n, int m2[][10], int result[][10]) { for (int i = 0; i < m; i++) {

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

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

}

}

}

void subtractMatrices(int m1[][10], int m, int n, int m2[][10], int result[][10]) { for (int i = 0; i < m; i++) {

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

result[i][j] = m1[i][j] - m2[i][j];

}

}

}

void transposeMatrix(int matrix[][10], int m, int n, int result[][10]) { for (int i = 0; i < m; i++) {

for (int j = 0; j < n; j++) { result[j][i] = matrix[i][j];

}

}

}

int main() {

int m1[10][10], m2[10][10], result[10][10];

int m, n, p, q; int choice;

printf("Enter the number of rows and columns for the first matrix: "); scanf("%d %d", &m, &n);

readMatrix(m1, m, n);

printf("Enter the number of rows and columns for the second matrix: "); scanf("%d %d", &p, &q);

readMatrix(m2, p, q);

printf("Choose an operation:\n"); printf("1. Display Matrices\n");

printf("2. Multiply Matrices\n"); printf("3. Add Matrices\n");

printf("4. Subtract Matrices\n");

printf("5. Transpose Matrix\n"); printf("Enter your choice: "); scanf("%d", &choice);

switch (choice) { case 1:

displayMatrix(m1, m, n); displayMatrix(m2, p, q); break;

case 2:

if (n != p) {

printf("Matrix multiplication is not possible.\n");

} else {

multiplyMatrices(m1, m, n, m2, p, q, result); displayMatrix(result, m, q);

}

break; case 3:

if (m != p || n != q) {

printf("Matrix addition is not possible.\n");

} else {

addMatrices(m1, m, n, m2, result); displayMatrix(result, m, n);

}

break; case 4:

if (m != p || n != q) {

printf("Matrix subtraction is not possible.\n");

} else {

subtractMatrices(m1, m, n, m2, result); displayMatrix(result, m, n);

}

break; case 5:

transposeMatrix(m1, m, n, result); displayMatrix(result, n, m); transposeMatrix(m2, p, q, result); displayMatrix(result, q, p);

break; default:

printf("Invalid choice.\n");

}

return 0;

}

OUTPUT:

