1. Write a program to store marks for n number of student in an array and print their marks.

#include <stdio.h>

int main() {

int n;

// Input the number of students

printf("Enter the number of students: ");

scanf("%d", &n);

int marks[n];

// Input marks for each student

printf("Enter marks for %d students:\n", n);

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

printf("Enter marks for student %d: ", i + 1);

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

}

// Print marks for each student

printf("Marks for %d students are:\n", n);

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

printf("Student %d: %d\n", i + 1, marks[i]);

}

return 0;

}

1. Write a program which stores the marks of subject Mathematics and English of n number of students in an array and then prints their individual total marks.

#include <stdio.h>

int main() {

int n;

// Input the number of students

printf("Enter the number of students: ");

scanf("%d", &n);

int mathMarks[n], englishMarks[n], totalMarks[n];

// Input marks for Mathematics and English for each student

printf("Enter marks for Mathematics and English for %d students:\n", n);

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

printf("Enter Mathematics marks for student %d: ", i + 1);

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

printf("Enter English marks for student %d: ", i + 1);

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

// Calculate total marks for each student

totalMarks[i] = mathMarks[i] + englishMarks[i];

}

// Print total marks for each student

printf("Total marks for %d students are:\n", n);

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

printf("Student %d: %d\n", i + 1, totalMarks[i]);

}

return 0;

}

1. Write a program to insert an element in an array in a particular position.

#include <stdio.h>

int main() {

int n, position, element;

// Input the size of the array

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

scanf("%d", &n);

int arr[n+1]; // Increased size of the array

// Input array elements

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

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

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

}

// Input the position and element to insert

printf("Enter the position to insert the element: ");

scanf("%d", &position);

printf("Enter the element to insert: ");

scanf("%d", &element);

// Shift elements to create space for the new element

for (int i = n; i >= position; i--) {

arr[i] = arr[i - 1];

}

// Insert the element at the specified position

arr[position - 1] = element;

// Print the array after insertion

printf("Array after insertion:\n");

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

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

}

return 0;

}

1. Write a program to delete an element from a particular position of an array.

#include <stdio.h>

int main() {

int n, position;

// Input the size of the array

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

scanf("%d", &n);

int arr[n];

// Input array elements

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

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

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

}

// Input the position to delete the element

printf("Enter the position to delete the element: ");

scanf("%d", &position);

// Shift elements to fill the gap left by the deleted element

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

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

}

// Print the array after deletion

printf("Array after deletion:\n");

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

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

}

return 0;

}

1. Write a program to convert a decimal number taken as input from user to corresponding binary number and store the result in an array.

#include <stdio.h>

int main() {

int decimalNum, temp, i = 0;

int binaryArr[32]; // Assuming 32-bit integer

// Input a decimal number

printf("Enter a decimal number: ");

scanf("%d", &decimalNum);

// Convert decimal to binary and store in an array

temp = decimalNum;

while (temp > 0) {

binaryArr[i] = temp % 2;

temp /= 2;

i++;

}

// Print the binary representation

printf("Binary representation: ");

for (int j = i - 1; j >= 0; j--) {

printf("%d", binaryArr[j]);

}

return 0;

}

1. Write a program to input a binary number in an array and convert into corresponding decimal number.

#include <stdio.h>

int main() {

int binaryArr[32], decimalNum = 0, base = 1, i, n;

// Input the size of the binary array

printf("Enter the size of the binary array: ");

scanf("%d", &n);

// Input binary digits

printf("Enter %d binary digits (0 or 1):\n", n);

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

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

// Validate input (binary digits should be 0 or 1)

if (binaryArr[i] != 0 && binaryArr[i] != 1) {

printf("Invalid input! Binary digits should be 0 or 1.\n");

return 1; // Exit with an error code

}

}

// Convert binary to decimal

for (i = n - 1; i >= 0; i--) {

decimalNum += binaryArr[i] \* base;

base \*= 2;

}

// Print the decimal representation

printf("Decimal representation: %d\n", decimalNum);

return 0;

}

1. Write a program to find the smallest and the largest elements in an array.

#include <stdio.h>

int main() {

int n;

// Input the size of the array

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

scanf("%d", &n);

int arr[n];

// Input array elements

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

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

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

}

// Find the smallest and largest elements

int smallest = arr[0], largest = arr[0];

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

if (arr[i] < smallest) {

smallest = arr[i];

}

if (arr[i] > largest) {

largest = arr[i];

}

}

// Print the smallest and largest elements

printf("Smallest element: %d\n", smallest);

printf("Largest element: %d\n", largest);

return 0;

}

1. Write a program for deleting duplicate elements in an array.

#include <stdio.h>

int main() {

int n;

// Input the size of the array

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

scanf("%d", &n);

int arr[n];

// Input array elements

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

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

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

}

// Delete duplicate elements

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

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

if (arr[i] == arr[j]) {

// Shift elements to fill the gap left by the duplicate element

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

arr[k] = arr[k + 1];

}

n--; // Decrease the size of the array

} else {

j++;

}

}

}

// Print the array after deleting duplicates

printf("Array after deleting duplicates:\n");

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

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

}

return 0;

}

1. Write a program to search a particular element in an array.

#include <stdio.h>

int main() {

int n, key;

// Input the size of the array

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

scanf("%d", &n);

int arr[n];

// Input array elements

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

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

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

}

// Input the element to be searched

printf("Enter the element to be searched: ");

scanf("%d", &key);

// Search for the element

int found = 0; // 0 indicates not found, 1 indicates found

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

if (arr[i] == key) {

found = 1;

printf("Element %d found at index %d\n", key, i);

break; // Exit the loop once the element is found

}

}

if (!found) {

printf("Element %d not found in the array\n", key);

}

return 0;

}

1. Write a program to sort n elements (ascending order).

#include <stdio.h>

int main() {

int n;

// Input the size of the array

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

scanf("%d", &n);

int arr[n];

// Input array elements

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

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

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

}

// Sort array in ascending order

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

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

if (arr[j] > arr[j + 1]) {

// Swap elements if they are in the wrong order

int temp = arr[j];

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

arr[j + 1] = temp;

}

}

}

// Print the sorted array

printf("Array after sorting in ascending order:\n");

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

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

}

return 0;

}

1. Write a program to find second highest number from the array without using sorting.

#include <stdio.h>

int main() {

int n;

// Input the size of the array

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

scanf("%d", &n);

int arr[n];

// Input array elements

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

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

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

}

// Find the second-highest number

int highest = arr[0], secondHighest = arr[0];

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

if (arr[i] > highest) {

secondHighest = highest;

highest = arr[i];

} else if (arr[i] > secondHighest && arr[i] != highest) {

secondHighest = arr[i];

}

}

// Print the second-highest number

printf("Second-highest number: %d\n", secondHighest);

return 0;

}

12. Write a program to perform addition and subtraction between two matrices.

#include <stdio.h>

int main() {

int m, n;

// Input the size of the matrices

printf("Enter the number of rows and columns for the matrices (m n): ");

scanf("%d %d", &m, &n);

int mat1[m][n], mat2[m][n], sumMat[m][n], diffMat[m][n];

// Input elements for the first matrix

printf("Enter elements for the first matrix:\n");

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

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

scanf("%d", &mat1[i][j]);

}

}

// Input elements for the second matrix

printf("Enter elements for the second matrix:\n");

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

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

scanf("%d", &mat2[i][j]);

}

}

// Perform addition and subtraction

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

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

// Addition

sumMat[i][j] = mat1[i][j] + mat2[i][j];

// Subtraction

diffMat[i][j] = mat1[i][j] - mat2[i][j];

}

}

// Print the sum matrix

printf("Sum of the matrices:\n");

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

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

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

}

printf("\n");

}

// Print the difference matrix

printf("Difference of the matrices:\n");

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

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

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

}

printf("\n");

}

return 0;

}

1. Write a program to transpose a matrix.

#include <stdio.h>

int main() {

int m, n;

// Input the size of the matrix

printf("Enter the number of rows and columns for the matrix (m n): ");

scanf("%d %d", &m, &n);

int mat[m][n], transposedMat[n][m];

// Input elements for the matrix

printf("Enter elements for the matrix:\n");

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

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

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

}

}

// Transpose the matrix

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

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

transposedMat[i][j] = mat[j][i];

}

}

// Print the transposed matrix

printf("Transposed matrix:\n");

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

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

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

}

printf("\n");

}

return 0;

}

1. Write a program to add the elements of each row and each column of a matrix.

#include <stdio.h>

int main() {

int m, n;

// Input the size of the matrix

printf("Enter the number of rows and columns for the matrix (m n): ");

scanf("%d %d", &m, &n);

int mat[m][n];

// Input elements for the matrix

printf("Enter elements for the matrix:\n");

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

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

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

}

}

// Calculate the sum of elements in each row and print

printf("Sum of elements in each row:\n");

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

int rowSum = 0;

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

rowSum += mat[i][j];

}

printf("Row %d: %d\n", i + 1, rowSum);

}

// Calculate the sum of elements in each column and print

printf("Sum of elements in each column:\n");

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

int colSum = 0;

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

colSum += mat[i][j];

}

printf("Column %d: %d\n", j + 1, colSum);

}

return 0;

}

1. Write a program to perform multiplication of two matrices.

#include <stdio.h>

int main() {

int m1, n1, m2, n2;

// Input the size of the first matrix

printf("Enter the number of rows and columns for the first matrix (m1 n1): ");

scanf("%d %d", &m1, &n1);

// Input the size of the second matrix

printf("Enter the number of rows and columns for the second matrix (m2 n2): ");

scanf("%d %d", &m2, &n2);

if (n1 != m2) {

printf("Multiplication not possible. Number of columns in the first matrix must be equal to the number of rows in the second matrix.\n");

return 1; // Exit with an error code

}

int mat1[m1][n1], mat2[m2][n2], productMat[m1][n2];

// Input elements for the first matrix

printf("Enter elements for the first matrix:\n");

for (int i = 0; i < m1; i++) {

for (int j = 0; j < n1; j++) {

scanf("%d", &mat1[i][j]);

}

}

// Input elements for the second matrix

printf("Enter elements for the second matrix:\n");

for (int i = 0; i < m2; i++) {

for (int j = 0; j < n2; j++) {

scanf("%d", &mat2[i][j]);

}

}

// Perform matrix multiplication

for (int i = 0; i < m1; i++) {

for (int j = 0; j < n2; j++) {

productMat[i][j] = 0;

for (int k = 0; k < n1; k++) {

productMat[i][j] += mat1[i][k] \* mat2[k][j];

}

}

}

// Print the product matrix

printf("Product of the matrices:\n");

for (int i = 0; i < m1; i++) {

for (int j = 0; j < n2; j++) {

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

}

printf("\n");

}

return 0;

}

1. Write a program to check whether a matrix is identity matrix or not.

#include <stdio.h>

int main() {

int n;

// Input the size of the square matrix

printf("Enter the size of the square matrix: ");

scanf("%d", &n);

int mat[n][n];

// Input elements for the matrix

printf("Enter elements for the matrix:\n");

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

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

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

}

}

// Check if the matrix is an identity matrix

int isIdentity = 1; // Assume it is an identity matrix

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)) {

isIdentity = 0; // It is not an identity matrix

break;

}

}

if (!isIdentity) {

break;

}

}

if (isIdentity) {

printf("The matrix is an identity matrix.\n");

} else {

printf("The matrix is not an identity matrix.\n");

}

return 0;

}

1. Write a program to check whether a matrix is sparse matrix or not

#include <stdio.h>

int main() {

int m, n;

// Input the size of the matrix

printf("Enter the number of rows and columns for the matrix (m n): ");

scanf("%d %d", &m, &n);

int mat[m][n];

// Input elements for the matrix

printf("Enter elements for the matrix:\n");

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

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

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

}

}

// Count the number of zeroes in the matrix

int zeroCount = 0;

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

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

if (mat[i][j] == 0) {

zeroCount++;

}

}

}

// Check if the matrix is sparse

if (zeroCount > (m \* n) / 2) {

printf("The matrix is a sparse matrix.\n");

} else {

printf("The matrix is not a sparse matrix.\n");

}

return 0;

}