Assignment 1 Code Given Below Submitted by

Sahil Mangla

Batch- 2C24

Roll no: 1024030359



1. #include <stdio.h>

int main() {

int arr[100], size, i, j, k;

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

scanf("%d", &size);

printf("Enter %d elements:\n", size);

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

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

}

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

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

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

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

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

}

size--;

} else {

j++;

}

}

}

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

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

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

}

return 0;

}

2. #include <stdio.h>

int main() {

int A[10][10], B[10][10], C[10][10];

int i, j, k, r1, c1, r2, c2;

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

scanf("%d%d", &r1, &c1);

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

scanf("%d%d", &r2, &c2);

if (c1 != r2) {

printf("Matrix multiplication not possible. Columns of A must equal rows of B.\n");

return 1;

}

printf("Enter elements of matrix A:\n");

for (i = 0; i < r1; i++) {

for (j = 0; j < c1; j++) {

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

}

}

printf("Enter elements of matrix B:\n");

for (i = 0; i < r2; i++) {

for (j = 0; j < c2; j++) {

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

}

}

for (i = 0; i < r1; i++) {

for (j = 0; j < c2; j++) {

C[i][j] = 0;

for (k = 0; k < c1; k++) {

C[i][j] += A[i][k] \* B[k][j];

}

}

}

printf("Resultant matrix after multiplication:\n");

for (i = 0; i < r1; i++) {

for (j = 0; j < c2; j++) {

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

}

printf("\n");

}

return 0;

}

3.#include <stdio.h>

#define MAX 100

int arr[MAX];

int size = 0;

void create() {

printf("Enter number of elements: ");

scanf("%d", &size);

if (size > MAX) {

printf("Size exceeds maximum limit!\n");

size = 0;

return;

}

printf("Enter elements:\n");

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

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

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

}

}

void display() {

if (size == 0) {

printf("Array is empty.\n");

return;

}

printf("Array elements: ");

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

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

}

printf("\n");

}

void insert() {

int pos, elem;

printf("Enter position to insert (0-based index): ");

scanf("%d", &pos);

printf("Enter element to insert: ");

scanf("%d", &elem);

if (pos < 0 || pos > size || size == MAX) {

printf("Invalid position or array is full.\n");

return;

}

for (int i = size; i > pos; i--) {

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

}

arr[pos] = elem;

size++;

printf("Element inserted.\n");

}

void delete\_element() {

int elem, pos = -1;

printf("Enter element to delete: ");

scanf("%d", &elem);

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

if (arr[i] == elem) {

pos = i;

break;

}

}

if (pos == -1) {

printf("Element not found.\n");

return;

}

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

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

}

size--;

printf("Element deleted.\n");

}

void linear\_search() {

int elem, found = 0;

printf("Enter element to search: ");

scanf("%d", &elem);

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

if (arr[i] == elem) {

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

found = 1;

break;

}

}

if (!found) {

printf("Element not found.\n");

}

}

int main() {

int choice;

do {

printf("\nMENU\n");

printf("1. CREATE\n");

printf("2. DISPLAY\n");

printf("3. INSERT\n");

printf("4. DELETE\n");

printf("5. LINEAR SEARCH\n");

printf("6. EXIT\n");

printf("Enter your choice: ");

scanf("%d", &choice);

switch (choice) {

case 1:

create();

break;

case 2:

display();

break;

case 3:

insert();

break;

case 4:

delete\_element();

break;

case 5:

linear\_search();

break;

case 6:

printf("Exiting the program.\n");

break;

default:

printf("Invalid choice. Please enter a number between 1 and 6.\n");

}

} while (choice != 6);

return 0;

}

4.#include <stdio.h>

int main() {

int arr[100], size, i, temp;

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

scanf("%d", &size);

printf("Enter %d elements:\n", size);

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

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

}

for (i = 0; i < size / 2; i++) {

temp = arr[i];

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

arr[size - i - 1] = temp;

}

printf("Reversed array:\n");

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

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

}

return 0;

}

5.#include <stdio.h>

int main() {

int A[10][10];

int rows, cols, i, j;

int rowSum, colSum;

printf("Enter number of rows and columns: ");

scanf("%d %d", &rows, &cols);

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

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

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

printf("Enter A[%d][%d]: ", i, j);

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

}

}

printf("\nMatrix:\n");

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

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

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

}

printf("\n");

}

printf("\nSum of each row:\n");

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

rowSum = 0;

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

rowSum += A[i][j];

}

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

}

printf("\nSum of each column:\n");

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

colSum = 0;

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

colSum += A[i][j];

}

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

}

return 0;

}

6.#include <stdio.h>

int main() {

int A[10][10], transpose[10][10];

int rows, cols, i, j;

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

scanf("%d %d", &rows, &cols);

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

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

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

printf("Enter element A[%d][%d]: ", i, j);

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

}

}

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

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

transpose[j][i] = A[i][j];

}

}

printf("\nOriginal Matrix:\n");

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

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

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

}

printf("\n");

}

printf("\nTransposed Matrix:\n");

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

for (j = 0; j < rows; ++j) {

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

}

printf("\n");

}

return 0;

}