/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Write a program to print elements of 1-D array using pointers

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <stdio.h>

int main() {

int arr[] = {10, 20, 30, 40, 50};

int size = sizeof(arr) / sizeof(arr[0]);

int \*ptr;

printf("Elements of the array:\n");

ptr = arr;

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

printf("%d ", \*(ptr + i));

}

printf("\n");

return 0;

}

A black background with white text

Description automatically generated

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Write a program to add two 3x3 matrices using pointers

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <stdio.h>

int main() {

int matrix1[3][3] = {{1, 2, 3}, {4, 5, 6}, {7, 8, 9}};

int matrix2[3][3] = {{9, 8, 7}, {6, 5, 4}, {3, 2, 1}};

int resultMatrix[3][3];

printf("Matrix 1:\n");

for (int i = 0; i < 3; i++) {

for (int j = 0; j < 3; j++) {

printf("%d ", \*(\*(matrix1 + i) + j));

}

printf("\n");

}

printf("\nMatrix 2:\n");

for (int i = 0; i < 3; i++) {

for (int j = 0; j < 3; j++) {

printf("%d ", \*(\*(matrix2 + i) + j));

}

printf("\n");

}

for (int i = 0; i < 3; i++) {

for (int j = 0; j < 3; j++) {

\*(\*(resultMatrix + i) + j) = \*(\*(matrix1 + i) + j) + \*(\*(matrix2 + i) + j);

}

}

printf("\nResultant Matrix (Matrix 1 + Matrix 2):\n");

for (int i = 0; i < 3; i++) {

A black screen with white text

Description automatically generated for (int j = 0; j < 3; j++) {

printf("%d ", \*(\*(resultMatrix + i) + j));

}

printf("\n");

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Write a program to search an element in an array using pointers.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <stdio.h>

int main() {

int arr[] = {10, 20, 30, 40, 50, 60, 70};

int size = sizeof(arr) / sizeof(arr[0]);

int searchElement;

int \*ptr;

int found = 0;

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

scanf("%d", &searchElement);

ptr = arr;

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

if (\*(ptr + i) == searchElement) {

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

found = 1;

break;

}

}

if (!found) {

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

}

return 0;

}

A black and white screen with white text

Description automatically generated