POINTER

1. Write a C program to find the sum of all the elements of an array using pointers.

#include <stdio.h>

int main() {

int n, sum = 0;

// Input the size of the array

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

scanf("%d", &n);

int arr[n];

// Input array elements

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

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

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

}

// Calculate the sum using pointers

int \*ptr = arr;

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

sum += \*ptr;

ptr++;

}

// Print the sum

printf("Sum of array elements: %d\n", sum);

return 0;

}

1. Write a C program to swap value of two variables using pointer.

#include <stdio.h>

void swap(int \*a, int \*b) {

int temp = \*a;

\*a = \*b;

\*b = temp;

}

int main() {

int num1, num2;

// Input two numbers from the user

printf("Enter the first number: ");

scanf("%d", &num1);

printf("Enter the second number: ");

scanf("%d", &num2);

// Print numbers before swapping

printf("Before swapping: num1 = %d, num2 = %d\n", num1, num2);

// Call function to swap values using pointers

swap(&num1, &num2);

// Print numbers after swapping

printf("After swapping: num1 = %d, num2 = %d\n", num1, num2);

return 0;

}

1. Write a C program to add two numbers using pointers.

#include <stdio.h>

int main() {

int num1, num2, sum;

// Input two numbers from the user

printf("Enter the first number: ");

scanf("%d", &num1);

printf("Enter the second number: ");

scanf("%d", &num2);

// Add numbers using pointers

int \*ptr1 = &num1;

int \*ptr2 = &num2;

sum = \*ptr1 + \*ptr2;

// Print the sum

printf("Sum of %d and %d is: %d\n", num1, num2, sum);

return 0;

}

1. Write a C program to input and print array elements using pointer.

#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 using pointers

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

int \*ptr = arr;

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

scanf("%d", ptr + i);

}

// Print array elements using pointers

printf("Array elements are:\n");

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

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

}

return 0;

}

1. Write a C program to copy one array to another using pointer.

#include <stdio.h>

int main() {

int n;

// Input the size of the array

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

scanf("%d", &n);

int arr1[n], arr2[n];

// Input array elements using pointers

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

int \*ptr1 = arr1;

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

scanf("%d", ptr1 + i);

}

// Copy array elements using pointers

int \*ptr2 = arr2;

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

\*(ptr2 + i) = \*(ptr1 + i);

}

// Print the copied array elements

printf("Copied array elements are:\n");

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

printf("%d ", \*(ptr2 + i));

}

return 0;

}

1. Write a C program to swap two arrays using pointers.

#include <stdio.h>

void swapArrays(int \*arr1, int \*arr2, int n) {

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

int temp = \*(arr1 + i);

\*(arr1 + i) = \*(arr2 + i);

\*(arr2 + i) = temp;

}

}

int main() {

int n;

// Input the size of the arrays

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

scanf("%d", &n);

int arr1[n], arr2[n];

// Input array elements for the first array

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

int \*ptr1 = arr1;

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

scanf("%d", ptr1 + i);

}

// Input array elements for the second array

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

int \*ptr2 = arr2;

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

scanf("%d", ptr2 + i);

}

// Call function to swap arrays using pointers

swapArrays(ptr1, ptr2, n);

// Print the swapped arrays

printf("Swapped arrays are:\n");

printf("First array: ");

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

printf("%d ", \*(ptr1 + i));

}

printf("\nSecond array: ");

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

printf("%d ", \*(ptr2 + i));

}

return 0;

}

1. Write a C program to reverse an array using pointers.

#include <stdio.h>

void reverseArray(int \*arr, int n) {

int \*start = arr;

int \*end = arr + n - 1;

while (start < end) {

// Swap elements using pointers

int temp = \*start;

\*start = \*end;

\*end = temp;

// Move pointers to the next elements

start++;

end--;

}

}

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 using pointers

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

int \*ptr = arr;

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

scanf("%d", ptr + i);

}

// Call function to reverse the array using pointers

reverseArray(ptr, n);

// Print the reversed array

printf("Reversed array is:\n");

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

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

}

return 0;

}

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

#include <stdio.h>

int searchElement(int \*arr, int n, int key) {

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

if (\*(arr + i) == key) {

return i; // Return the index if found

}

}

return -1; // Return -1 if not found

}

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 using pointers

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

int \*ptr = arr;

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

scanf("%d", ptr + i);

}

// Input the element to be searched

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

scanf("%d", &key);

// Call function to search for the element using pointers

int index = searchElement(ptr, n, key);

// Print the result

if (index != -1) {

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

} else {

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

}

return 0;

}

1. Write a C program to add two 2 X 2 matrix using pointers.

#include <stdio.h>

void addMatrices(int mat1[2][2], int mat2[2][2], int result[2][2]) {

for (int i = 0; i < 2; i++) {

for (int j = 0; j < 2; j++) {

\*(\*(result + i) + j) = \*(\*(mat1 + i) + j) + \*(\*(mat2 + i) + j);

}

}

}

int main() {

int mat1[2][2], mat2[2][2], result[2][2];

// Input elements for the first matrix

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

for (int i = 0; i < 2; i++) {

for (int j = 0; j < 2; j++) {

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

}

}

// Input elements for the second matrix

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

for (int i = 0; i < 2; i++) {

for (int j = 0; j < 2; j++) {

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

}

}

// Call function to add matrices using pointers

addMatrices(mat1, mat2, result);

// Print the result matrix

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

for (int i = 0; i < 2; i++) {

for (int j = 0; j < 2; j++) {

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

}

printf("\n");

}

return 0;

}

1. Write a C program to multiply two 2 X 2 matrix using pointers.

#include <stdio.h>

void multiplyMatrices(int mat1[2][2], int mat2[2][2], int result[2][2]) {

for (int i = 0; i < 2; i++) {

for (int j = 0; j < 2; j++) {

\*(\*(result + i) + j) = 0;

for (int k = 0; k < 2; k++) {

\*(\*(result + i) + j) += \*(\*(mat1 + i) + k) \* \*(\*(mat2 + k) + j);

}

}

}

}

int main() {

int mat1[2][2], mat2[2][2], result[2][2];

// Input elements for the first matrix

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

for (int i = 0; i < 2; i++) {

for (int j = 0; j < 2; j++) {

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

}

}

// Input elements for the second matrix

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

for (int i = 0; i < 2; i++) {

for (int j = 0; j < 2; j++) {

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

}

}

// Call function to multiply matrices using pointers

multiplyMatrices(mat1, mat2, result);

// Print the result matrix

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

for (int i = 0; i < 2; i++) {

for (int j = 0; j < 2; j++) {

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

}

printf("\n");

}

return 0;

}

1. Write a C program to find length of string using pointers.

#include <stdio.h>

int findLength(char \*str) {

int length = 0;

while (\*str != '\0') {

length++;

str++;

}

return length;

}

int main() {

char str[100];

// Input a string from the user

printf("Enter a string: ");

gets(str);

// Call function to find the length of the string using pointers

int length = findLength(str);

// Print the length of the string

printf("Length of the string is: %d\n", length);

return 0;

}

1. Write a C program to copy one string to another using pointer.

#include <stdio.h>

void copyString(char \*dest, const char \*src) {

while ((\*dest++ = \*src++))

;

}

int main() {

char source[100], destination[100];

printf("Enter source string: ");

gets(source);

copyString(destination, source);

printf("Copied string: %s\n", destination);

return 0;

}

1. Write a C program to concatenate two strings using pointers.

#include <stdio.h>

void concatenateStrings(char \*str1, const char \*str2) {

while (\*str1)

str1++;

while ((\*str1++ = \*str2++))

;

}

int main() {

char string1[100], string2[100];

printf("Enter first string: ");

gets(string1);

printf("Enter second string: ");

gets(string2);

concatenateStrings(string1, string2);

printf("Concatenated string: %s\n", string1);

return 0;

}

1. Write a C program to compare two strings using pointers.

#include <stdio.h>

int compareStrings(const char \*str1, const char \*str2) {

while (\*str1 && \*str2 && (\*str1 == \*str2)) {

str1++;

str2++;

}

return (\*str1 - \*str2);

}

int main() {

char string1[100], string2[100];

printf("Enter first string: ");

gets(string1);

printf("Enter second string: ");

gets(string2);

int result = compareStrings(string1, string2);

if (result == 0)

printf("Strings are equal.\n");

else if (result > 0)

printf("First string is greater.\n");

else

printf("Second string is greater.\n");

return 0;

}