1. Write a program to print the address of a variable using poicnter.

#include <stdio.h>

Viod main() {

int num = 10;

int \*ptr = &num;

printf("Value: %d\n", num);

printf("Address: %p\n", ptr);

}

1. Write a program to access array elements using pointer.

#include <stdio.h>

void main() {

int arr[5] = {10, 20, 30, 40, 50};

int \*ptr = arr;

for (int i = 0; i < 5; i++) {

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

}

}

1. Write a program to swap two numbers using pointers.

#include <stdio.h>

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

int temp = \*a;

\*a = \*b;

\*b = temp;

}

Void main() {

int x = 5, y = 10;

printf("Before swap: x = %d, y = %d\n", x, y);

swap(&x, &y);

printf("After swap: x = %d, y = %d\n", x, y);

}

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

#include <stdio.h>

void main() {

int a = 7, b = 8;

int \*p1 = &a, \*p2 = &b;

int sum = \*p1 + \*p2;

printf("Sum = %d\n", sum);

}

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

#include <stdio.h>

void main() {

char str[] = "hello";

char \*ptr = str;

int len = 0;

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

len++;

ptr++;

}

printf("Length = %d\n", len);

}

1. Write a program to reverse a string using pointers.

#include <stdio.h>

#include <string.h>

void main() {

char str[] = "hello";

char \*start = str;

char \*end = str + strlen(str) - 1;

char temp;

while (start < end) {

temp = \*start;

\*start = \*end;

\*end = temp;

start++;

end--;

}

printf("Reversed string: %s\n", str);

}

1. Write a program to count vowels using pointer.

#include <stdio.h>

Void main() {

char str[] = "hello world";

char \*ptr = str;

int count = 0;

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

char ch = \*ptr;

if (ch == 'a' || ch == 'e' || ch == 'i' || ch == 'o' || ch == 'u' ||

ch == 'A' || ch == 'E' || ch == 'I' || ch == 'O' || ch == 'U') {

count++;

}

ptr++;

}

printf("Vowel count: %d\n", count);

}

1. Write a program to demonstrate pointer to pointer.

#include <stdio.h>

Void main() {

int num = 50;

int \*ptr = &num;

int \*\*pptr = &ptr;

printf("Value = %d\n", \*\*pptr);

printf("Address = %p\n", \*pptr);

}

1. Write a program to allocate memory using malloc() and free it.

#include <stdio.h>

void main() {

int \*ptr;

int n = 5;

ptr = (int \*)malloc(n \* sizeof(int));

if (ptr == NULL) {

printf("Memory not allocated\n");

return 1;

}

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

ptr[i] = i + 1;

}

printf("Allocated array: ");

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

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

}

free(ptr);

}

1. Write a program to sort an array using pointer notation.

#include <stdio.h>

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

int temp;

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

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

if (\*(arr + j) > \*(arr + j + 1)) {

temp = \*(arr + j);

\*(arr + j) = \*(arr + j + 1);

\*(arr + j + 1) = temp;

}

}

}

}

int main() {

int arr[5] = {30, 10, 50, 20, 40};

int n = 5;

sort(arr, n);

printf("Sorted array: ");

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

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

}

}