**Day 7**

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

#include <stdio.h>

int main() {

int x = 10;

int \*ptr = &x;

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

return 0;

}

2. Write a program to access array elements using pointers.

#include <stdio.h>

int main() {

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

int \*ptr = arr;

for(int i = 0; i < 4; i++) {

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

}

return 0;

}

3. 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;

}

int main() {

int x = 5, y = 10;

swap(&x, &y);

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

return 0;

}

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

#include <stdio.h>

int main() {

int a = 5, b = 7;

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

int sum = \*p1 + \*p2;

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

return 0;

}

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

#include <stdio.h>

int main() {

char str[] = "Hello";

char \*p = str;

int length = 0;

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

length++;

p++;

}

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

return 0;

}

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

#include <stdio.h>

#include <string.h>

int main() {

char str[] = "hello";

int len = strlen(str);

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

printf("%c", \*(str + i));

}

return 0;

}

7. Write a program to count vowels using pointer.

#include <stdio.h>

int main() {

char str[] = "Pointer Example";

char \*p = str;

int count = 0;

while (\*p) {

char ch = \*p;

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

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

count++;

}

p++;

}

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

return 0;

}

8. Write a program to demonstrate pointer to pointer.

#include <stdio.h>

int main() {

int x = 10;

int \*ptr = &x;

int \*\*pptr = &ptr;

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

return 0;

}

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

#include <stdio.h>

#include <stdlib.h>

int main() {

int \*ptr = (int\*)malloc(5 \* sizeof(int));

if (ptr == NULL) {

printf("Memory not allocated.\n");

return 1;

}

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

ptr[i] = i + 1;

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

}

free(ptr);

return 0;

}

10. Write a program to sort an array using pointer Notation

#include <stdio.h>

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

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

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

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

int temp = \*(arr + i);

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

\*(arr + j) = temp;

}

}

}

}

int main() {

int arr[] = {4, 2, 3, 1, 5};

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

sort(arr, n);

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

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

}

return 0;

}