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

IPO

input-Enter a value as input.

process-To print the address of a variable using pointer.

Output-output the variable

program

```
#include <stdio.h>
int main()
 {
    int num = 42;
    int *ptr;
    ptr = &num;
    printf("Value of num: %d\n", num);
    printf("Address of num: %p\n", (void*)&num);
    printf("Address stored in pointer: %p\n", (void*)ptr);
    return 0;
}
```

Output

```
Value of num: 42
Address of num: 0x7ffea4c7b5ac
Address stored in pointer: 0x7ffea4c7b5ac
```

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

IPO

input-Enter a value as input.

process-To access array elements using pointers.

Output-output the variable

program

```
#include <stdio.h>
int main()
{
    int arr[5] = {10, 20, 30, 40, 50};
    int *ptr;
    int i;
    ptr = arr; // Pointer points to the first element of the array
    printf("Array elements using pointers:\n");
    for (i = 0; i < 5; i++)
 {
        printf("Element %d: %d\n", i, *(ptr + i));
```

```
    }

    return 0;
}
```

Output

```
Array elements using pointers:
Element 0: 10
Element 1: 20
Element 2: 30
Element 3: 40
Element 4: 50
```

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

IPO

input-Enter a value as input.

process-To swap two numbers using pointers.

Output-output the variable

program

```
#include <stdio.h>

int main() {
    int a, b, temp;
    int *p1, *p2;

    printf("Enter two numbers: ");
    scanf("%d %d", &a, &b);

    p1 = &a;
    p2 = &b;

    temp = *p1;
    *p1 = *p2;
    *p2 = temp;

    printf("After swapping:\n");
    printf("a = %d\n", a);
    printf("b = %d\n", b);

    return 0;
}
```

Output
Enter two numbers: 5 10
After swapping:
a = 10
b = 5

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

IPO
input-Enter a value as input.
process-To add two numbers using pointers.
Output-output the variable

program

```
#include <stdio.h>

int main() {
    int num1, num2, sum;
    int *p1, *p2;

    printf("Enter two numbers: ");
    scanf("%d %d", &num1, &num2);

    p1 = &num1;
 p2 = &num2;
    sum = *p1 + *p2;
    printf("Sum = %d\n", sum);

    return 0;
}
```

Output
Enter two numbers: 7 13
Sum = 20

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

IPO
input-Enter a value as input.
process-To find the length of a string using pointers.
Output-output the variable

Program

```
#include <stdio.h>

int main() {
    char str[100];
    char *ptr;
    int length = 0;

    printf("Enter a string: ");
    scanf("%s", str);
    ptr = str;    while (*ptr != '\0') {         length++;
        ptr++;
    }

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

    return 0;
}
```

Output

```
Enter a string: hello
Length of the string: 5
```

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

IPO
input-Enter a value as input.
process-To reverse a string using pointers.
Output-output the variable

Program

```
#include <stdio.h>
#include <string.h>

int main() {
    char str[100], temp;
    char *start, *end;
```

```
    printf("Enter a string: ");
    scanf("%s", str);  // Reads a word (no spaces)

    start = str;
    end = str + strlen(str) - 1;

    while (start < end) {
        // Swap characters
        temp = *start;
        *start = *end;
        *end = temp;

        start++;
        end--;
    }

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

    return 0;
}
```

Output
Enter a string: hello
Reversed string: olleh

7. Write a program to count vowels using pointer.

IPO
input-Enter a value as input.
process-To count vowels using pointer.
Output-output the variable

Program

```
#include <stdio.h>

int main() {
    char str[100];
    char *ptr;
    int count = 0;
```

```
    printf("Enter a string: ");
    scanf("%s", str);

    ptr = str;
    while (*ptr != '\0') {
            if (*ptr == 'a' || *ptr == 'e' || *ptr == 'i' || *ptr == 'o' || *ptr == 'u' ||
          *ptr == 'A' || *ptr == 'E' || *ptr == 'I' || *ptr == 'O' || *ptr == 'U') {
          count++;
       }
       ptr++;
    }

    printf("Number of vowels: %d\n", count);

    return 0;
}
```

Output
Enter a string: education
Number of vowels: 5

8. Write a program to demonstrate pointer to pointer.

IPO
input-Enter a value as input.
process-To demonstrate pointer to pointer.
Output-output the variable

Program

```
#include <stdio.h>

int main() {
    int num = 42;
    int *ptr;
    int **pptr;
    ptr = &num;
    pptr = &ptr;
```

```
    printf("Value of num: %d\n", num);
    printf("Value of num using *ptr: %d\n", *ptr);
    printf("Value of num using **pptr: %d\n", **pptr);

    printf("Address of num: %p\n", (void*)&num);
    printf("Address stored in ptr: %p\n", (void*)ptr);
    printf("Address stored in pptr: %p\n", (void*)pptr);

    return 0;
}
```

Output
Value of num: 42
Value of num using *ptr: 42
Value of num using **pptr: 42
Address of num: 0x7ffee4b5a9dc
Address stored in ptr: 0x7ffee4b5a9dc
Address stored in pptr: 0x7ffee4b5a9d0

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

IPO
 input-Enter a value as input.
process-To demonstrate pointer to pointer.
Output-output the variable

Program

```
#include <stdio.h>
#include <stdlib.h>
int main() {
    int n, i;
    int *ptr;

    printf("Enter number of elements: ");
    scanf("%d", &n);
    ptr = (int*)malloc(n * sizeof(int));
    if (ptr == NULL) {
        printf("Memory allocation failed!\n");
        return 1;
    }
    printf("Enter %d integers:\n", n);
    for (i = 0; i < n; i++) {
```

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

    // Output values
    printf("You entered: ");
    for (i = 0; i < n; i++) {
        printf("%d ", *(ptr + i));
    }
    free(ptr);

    return 0;
}
```

Output
Enter number of elements: 5
Enter 5 integers:
10 20 30 40 50
You entered: 10 20 30 40 50

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

IPO
input-Enter a value as input.
process-To sort an array using pointer notation.
Output-output the variable

Program

```
#include <stdio.h>

int main() {
    int arr[100], n, i, j, temp;
    int *ptr;

    printf("Enter number of elements: ");
    scanf("%d", &n);

    ptr = arr;    printf("Enter %d integers:\n", n);
    for (i = 0; i < n; i++) {
        scanf("%d", (ptr + i)); n
```

```
    }

    // Sorting using pointer notation
    for (i = 0; i < n - 1; i++) {
        for (j = i + 1; j < n; j++) {
            if (*(ptr + i) > *(ptr + j)) {
                temp = *(ptr + i);
                *(ptr + i) = *(ptr + j);
                *(ptr + j) = temp;
            }
        }
    }
    printf("Sorted array: ");
    for (i = 0; i < n; i++) {
        printf("%d ", *(ptr + i*
```

Output

```
Enter number of elements: 5
Enter 5 integers:
50 10 30 20 40
Sorted array: 10 20 30 40 50
```