1.Wap to change the value of a variable using a pointer

```
#include <stdio.h>
int main()
{
   int a;
   a = 10;
   int *p = &a;     // declaring and initializing the pointer
   //prints the value of 'a'
   printf("%d\n", a);
   *p=34;  //change the value stored in variable a
   //prints the value of 'a'
   printf("%d\n", a);
   return 0;
}
```

Output:

```
10
34
```

2. Increment (++) and Decrement(--) operation

.

```
#include <stdio.h>
int main()
{
        // Initializing integer variable
        int a = 34;
        // Declaring pointer variable
        int* ptr_a;
        // Initializing pointer variable
        ptr_a = &a;
        // Value of a before increment
        printf("Increment:\n");
        printf("Before increment a = %d\n", *ptr_a);
        // Unary increment operation
        (*ptr_a)++; //Increases the value of a by 1
        // Value of a after increment
        printf("After increment a = %d", *ptr_a);
        // Value before decrement
        printf("\n\nDecrement:\n");
        printf("Before decrement a = %d\n", *ptr_a);
        // unary decrement operation
        (*ptr_a)--; //Decreases the value of a by 1
        // Value after decrement
```

Compiled by:
Er.Gaurab Mishra (Head of Department of Computer Science) KMC +2 Bagbazar

```
        printf("After decrement a=%d", *ptr_a);
        return 0;
}
```

Output:

```
Increment:
Before increment a = 34
After increment a = 35

Decrement:
Before decrement a = 35
After decrement a=34
--------------------------------
```

3.Addition of an integer to a pointer variable.

```
//Important thing  Rule:
//new_address=current_address + n* sizeof(datatype)
//Where n is the number by which the pointer gets increased

#include <stdio.h>
int main()
{
   int a = 10;
   int *b = &a;     // declaring and initializing the pointer
   printf("Memory address before addition of an integer to pointer=%u",b);
   b=b+4;//addition of an integer to a pointer variable.
   printf("\n After adding 4 to the memory address of a");
   printf("\n Memory address after addition of an integer to pointer=%u",b);
   return 0;
}
```

Output:

```
Memory address before addition of an integer to pointer=6684180
 After adding 4 to the memory address of a
 Memory address after addition of an integer to pointer=6684196
```

4.Subtraction of an integer to a pointer variable

```
//Important thing  Rule:
//new_address=current_address -  n* sizeof(datatype)
//Where n is the number by which the pointer gets decreased

#include <stdio.h>
int main()
{
```

```
    int a = 10;
    int *b = &a;     // declaring and initializing the pointer
    printf("Memory address before subtraction of an integer to pointer=%u",b);
    b=b-4;//subtraction of an integer to a pointer variable.
    printf("\n After adding 4 to the memory address of a");
    printf("\n Memory address after subtraction of an integer to pointer=%u",b);
    return 0;
}
```

Output:

```
Memory address before subtraction of an integer to pointer=6684180
 After subtracting 4 to the memory address of a
 Memory address after subtraction of an integer to pointer=6684164
```

Pointer Assignments

5. Wap to assign the value of one pointer to another pointer.

```
#include <stdio.h>
int main()
{
            int a;
             a = 10;
             int *ptr1,*ptr2;
           ptr1 = &a;
           ptr2=ptr1;
           printf("values at ptr1 and ptr2 %d %d\n", *ptr1,*ptr2);
           printf("Address pointed to by ptr1 and ptr2: %u %u",ptr1,ptr2);
          return 0;
}
```

Output:

```
values at ptr1 and ptr2 10 10
Address pointed to by ptr1 and ptr2: 6684172 6684172
------------------------------
```