DATE: LAB REPORT NO.: **8** SET: **A**

TITLE OF THE PROGRAM: **Basic Pointer**

OBJECTIVES

- To understand the basic declaration and initialization of pointers.
- To learn the use of & and * operators with pointers.
- To perform arithmetic operations using pointers.
- To find the greatest among two numbers using pointers.

REQUIREMENTS

1. C Compiler (e.g., GCC)
2. Computer System
3. IDE or Text Editor
4. OS compatible with the software

THEORY

A Pointer in C language is a variable that holds the address of another variable of the same data type. Pointers are used to access memory and manipulate the address**. Here is how we can declare pointers:**

```
int* p;
```

Here, we have declared a pointer **p** of int type. **You can also declare pointers in these ways**:

```
int *p1;
int * p2;
```

PROCEDURE (Program Code, Comment, and Output)

1. Basic Declaration of a Pointer and Use of & and * Operators

```c
#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", *p);
    printf("%d\n", *&a);

// Prints the address of 'a'

    printf("%u\n", &a);

    printf("%u\n", p);

   printf("%u\n", &p); // Prints address of 'p'

  return 0;

}
```

Output

Explanation

- *p and *&a both give the value stored in a which is 10.
- &a and p both give the address of a.
- &p gives the address of the pointer p.

2. Program to Find the Arithmetic Calculation Two Numbers Using a Pointer

```c
#include <stdio.h>

  int main()
{
int fno, sno, *ptr, *qtr, sum, subtract, mul, div;
printf("\n\n Pointer : Add two numbers :\n");
printf ("-------------------------------------------------------\n");
printf(" Input the first number : ");
scanf("%d", &fno);
printf(" Input the second number : ");
scanf("%d", &sno);

  ptr = &fno;
  qtr = &sno;
  sum = *ptr + *qtr;
  subtract = *ptr - *qtr;
  mul = *ptr *  *qtr;
 div = *ptr / *qtr;
printf(" The sum of the entered numbers is : %d\n\n", sum);
printf(" The subtraction of the entered numbers is : %d\n\n", subtract);
printf(" The multiplication of the entered numbers is : %d\n\n", mul);
printf(" The division of the entered numbers is : %d\n\n", div);

  return 0;

}
```

Output

```
Pointer : Add two numbers :
---------------------------------
Input the first number : 5
Input the second number : 6
The sum of the entered numbers is : 11
The subtraction of the entered numbers is : -1
The multiplication of the entered numbers is : 30
The division of the entered numbers is : 0
```

Explanation:

- This program demonstrates how to perform arithmetic operations using pointers.
- ptr and qtr are pointers to fno and sno, respectively.
- Arithmetic operations are performed using dereferenced pointers

3. Program to Find the Greatest Among Two Numbers Using a Pointer

```
#include <stdio.h>

 int main()

{

 int fno, sno, *ptr1 = &fno, *ptr2 = &sno;

 printf("\n\n Pointer : Find the maximum number between two numbers :\n");

 printf("-------------------------------------------------------------------------------------\n");

 printf(" Input the first number : ");

  // ptr1 already holds the address of fno so no need to give &

 scanf("%d", ptr1);

 printf(" Input the second number : ");

  // ptr2 already holds the address of sno so no need to give &

  scanf("%d", ptr2);

  if (*ptr1 > *ptr2) {

    printf("\n\n %d is the maximum number.\n\n", *ptr1);

  }

  else {

     printf("\n\n %d is the maximum number.\n\n", *ptr2);

        }

  return 0;

}
```

Output

```
Pointer : Find the maximum number between two numbers :
-------------------------------------------------------------
Input the first number : 5
Input the second number : 6
6 is the maximum number.
```

Explanation:

- This program compares two numbers using pointers.
- ptr1 and ptr2 are pointers to fno and sno, respectively.
- The program compares the values pointed to by ptr1 and ptr2 to determine the greater number.

CONCLUSION

In this lab, we explored the practical applications of pointers in C programming. We learned how to declare and initialize pointers, use & and * operators, perform arithmetic operations using pointers, and find the greatest among two numbers using pointers. This knowledge enhances our ability to manipulate memory addresses directly, which is a fundamental concept in C programming and critical for optimizing performance in various applications.