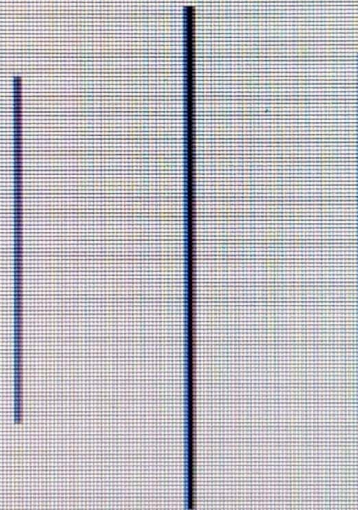


**INSTITUTE OF ENGINEERING
ADVANCED COLLEGE OF ENGINEERING
& MANAGEMENT
KUPONDOLE, LATIPUR
(AFFILIATED TO TRIBHUVAN UNIVERSITY)**



LAB REPORT

LAB NO.: 8

SUBJECT: C PROGRAMMING

SUBMITTED BY:

**NAME: ASHWANI KUMAR CHAUDHARY
ROLL NO.: 019
DATE: 2078/**

SUBMITTED TO:

**DEPARTMENT OF
COMPUTER &
ELECTRONICS**

Pointers in C-programming

Objective

To understand programming with pointer & function call by reference

Theory

A pointer in C-language is a variable which hold the address of another variable of some data type. Pointers are used to access memory & manipulate the content of that address.

Declaration of Pointer variables

Syntax.

Data-Type * Variable_Names.

Initialisation of pointer variables.

Pointer initialization is the process of assigning address of a variable to a pointer variable. Pointer variable can only contains address of a variable of the same data type.

Example:

```
void main()  
{  
    int a = 6;  
    int *p;  
    p = &a;  
}
```

Dereference operator

The dereference operator or indirection operator, sometimes denoted by '*' is unary operator found in C-language. It operates on the pointer variable, return value equivalent to the value at the pointer address. It is used in these ways.

- Declaration
- Dereference operator.

Bad pointer

A bad pointer is a pointer which contains a random address just like an uninitialized int variable which starts out with a random int value.

Null pointer.

A null pointer is a pointer which points nothing. It is used to pass a null pointer to function argument when we don't want to pass any valid memory address.

Function call by reference.

To pass a value by reference, argument pointers are passed to function just like any other values so, accordingly we need to declare the function parameter as pointer type.

Example,

```
void swap (int *a, int *b).
```

```
{  
    int c;  
    *c = *a;  
    *a = *b;  
    *b = c;  
    return 0;  
}
```