Experiment Details

|  |  |
| --- | --- |
| Department Name | Computer Science and Engineering |
| Class | S.Y.BTech |
| Semester | 1 |
| Subject Name | Data Structures Lab |
| Experiment No. | 1 |
| Experiment Name | Pointers in C |

Version History

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sr. No. | Version Number | Created By | Approved By | Date |
| 1 | v1.0 | Jay Deelip Kamble | Prof. Shivani Kale Ma’am | 17/10/2020 |
|  |  |  |  |  |

AIM:

To study and understand the concept of Pointers

THEORY:

1. A pointer is a variable that stores the address of another variable.
2. Unlike other variables that hold values of a certain type, pointer holds the address of a variable.
3. For example, an integer variable holds (or you can say stores) an integer value, however an integer pointer holds the address of a integer variable.

Example to understand pointers:-

#include <stdio.h>

int main()

{

int num = 10;

printf("Value of variable num is: %d", num);

/\* To print the address of a variable we use %p

\* format specifier and ampersand (&) sign just

\* before the variable name like &num.

\*/

printf("\nAddress of variable num is: %p", &num);

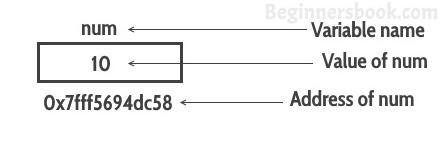
return 0;

}

Output:

Value of variable num is: 10

Address of variable num is: 0x7fff5694dc58



**Operators used with pointers:-**

### “Address of”(&) Operator

We have already seen in the first example that we can display the address of a variable using ampersand sign. I have used &num to access the address of variable num. The & operator is also known as “Address of” Operator.

printf("Address of var is: %p", &num);

## “Value at Address”(\*) Operator

The \* Operator is also known as Value at address operator.

How to declare a pointer?

int \*p1 /\*Pointer to an integer variable\*/

double \*p2 /\*Pointer to a variable of data type double\*/

char \*p3 /\*Pointer to a character variable\*/

float \*p4 /\*pointer to a float variable\*/

The above are the few examples of pointer declarations. If you need a pointer to store the address of integer variable then the data type of the pointer should be int. Same case is with the other data types.

By using \* operator we can access the value of a variable through a pointer.  
For example:

double a = 10;

double \*p;

p = &a;

\*p would give us the value of the variable a. The following statement would display 10 as output.

printf("%d", \*p);

Similarly if we assign a value to \*pointer like this:

\*p = 200;

It would change the value of variable a. The statement above will change the value of a from 10 to 200.

PRE TEST:

**What type of code does the compiler generates?**

1. Assembly code
2. **Object code**
3. Executable code
4. None of these

**Which of the following is a logical OR operator?**

1. &
2. &&
3. ||
4. None of these

**Which of the following is not a valid variable name declaration?**

1. int\_a3;
2. int a\_3;
3. **int 3\_a;**
4. int\_3a;

**Each and every keyword in C language is\_\_\_\_\_\_\_**

1. **Lowercase Letter**
2. CamelCase Letter
3. UpperCase Letter
4. Non of these

**Which of the following is not a valid variable syntax in C?**

1. float rate;
2. int number;
3. int variable\_count;
4. **int $main;**

PROCEDURE:

1. Execute the code line by line
2. Understand and interpret the code.

POST TEST:

**Before using a pointer variable,**

1. It should be initialized
2. It should be declared
3. **It should be both declared and initialized**
4. None of these

**In C, how can we create a pointer variable to an integer ?**

1. **int \*p;**
2. int p\*;
3. int +p;
4. int $p;

**A pointer variable can be**

1)Changed with a function

2)Passed to a function

**3)Returned by a function**

4)Can be assigned an integer value

**Let x be an array. Which of the following is not possible?**

1. x\*2
2. ++x
3. x++
4. **All of the above**

**In the pointer concept, “&” is called as\_\_\_\_\_\_**

1. Logical operator
2. Conditional operator
3. **Address operator**
4. None of these

REFERENCES:

https://compsciedu.com/mcq-questions/C-Programming/Pointers-and-Arrays-in-C/28

https://www.geeksforgeeks.org/pointers-in-c-and-c-set-1-introduction-arithmetic-and-array/