

## Basics of Pointers, their uses in C Programming while dynamic memory allocation.

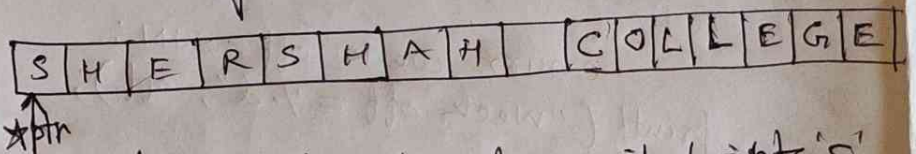
Introduction to Pointer— In C Programming, a pointer is a special type of variable that stores memory address of another variables. Pointer allows direct access to a specific location in memory. Pointer is declared by using the asteric (\*) symbol before the variable name.

Pointer is a fundamental concept used for dynamic memory allocation, accessing array elements and building complex data structures like - list, stack, queue and tree etc.

### Accessing and printing value :-

\* Symbol before pointer name is used to print and access the value stored at memory

Pointer Arithmetic :- Basic increment and decrement can be used with pointer variable to navigate through memory locations within an array such as:-



In above diagram ptr is pointing to S, it prints 'S'. If we increase it by 1 then it will print 'H' and similarly 'E' and 'R' and so on. But if we decrease it then previous character will be printed.

In this way we can traverse all elements in both order from first to last and from last to first.

Example:-  

```
int num=10;  
int *ptr; //declaring a pointer variable  
ptr=&num; // Assigning the memory address of num to ptr.
```

```
printf("value stored in num = %d\n", num);  
printf("value accessed through ptr = %d\n", *ptr);
```

NULL POINTER — A pointer with a value of NULL indicates that it does not point to any valid memory location.

Pointer to a Pointer — A pointer that stores the address of another pointer is called multi-level pointer. It is declared as:-  

```
int **ptr;
```

It can hold address of a normal pointer variable.

Let us see the following example which uses the pointer for string elements accessing: —

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

```
int main() { char name[10] = "SASARAM";
```

```
    char *ptr;
```

```
    ptr = name;
```

```
    // ptr now pointing to first character of name i.e., 'S'.
```

```
    // Now if we want to print it with pointer then we can
```

```
    // use for loop or while loop both according to our ease.
```

```
    while (*ptr != '\0')
```

```
    { printf(" %c", *ptr);
```

```
      ptr++;
```

```
    }
```

```
    // The above while loop prints "SASARAM"
```

```
    // Since we know that every string is terminated
```

```
    // automatically by '\0' by the compiler.
```

```
    return 0;
```

```
    } // close of main()
```

*You can compile & run this program without writing commented lines. These comments are only for understanding purpose.*

You can also accept user input through scanf function to read the string by keyboard and store it into variable name[] array.

The above statement

```
ptr = name;
```

is similar to 

```
ptr = &name[0];
```

Both statements have the same meaning.

Since, we know that name of any array represents only address of its first element that means &name[0] and name are same here.