CS & IT [redacted]

ENGINEERING

C PROGRAMMING

Data Types and I/O Functions

Lecture No.- 03

By- Satya sir

Recap of Previous Lecture

- Printf ( ) function => Process characters to o/p console, returns the count of characters Processed.

- I/o Functions
  - Forms of input, output
  - For Text input | output
    - Console i/o functions (i/p ← keyboard, o/p → Monitor)
    - File i/o functions (i/o ↔ Files | memory)
  - Console i/o functions
    - Formatted i/o : Scanf(), Printf()
    - Unformatted i/o

Topics to be Covered

- Printf ( ) syntax

- Printing different data types with format Specifiers

- Scanf ( )

Topic : I/O Functions - 2

Associativity of ( ) is Left to Right only

Associativity of ,(Comma) is Left to Right

Ex:1

```
int a, b;

a = 5, 4, 3;

b = 2, -1, 0;

Printf("a= %d, b= %d", a, b);
```

Operator

Ignored

a) 5 2

b) 4 -1

c) 3 0

d) Error

Ex:2

```
int a, b;

a=(5, 4, 3);

b=(2,-1,0);

Printf("%d %d", a,b);
```

seperator

a = ~~5~~ ~~4~~ 3

b = ~~2~~ ~~-1~~ 0

a) 5 2

b) 4 -1

c) ✓ 3 0

d) Error

NOTE: Without ( ), Left most Value is assigned to LHS but with ( ), right most Value is assigned.

Topic : I/O Functions - 2

→ Pre-defined function in library.

Printf ( ) Syntax :

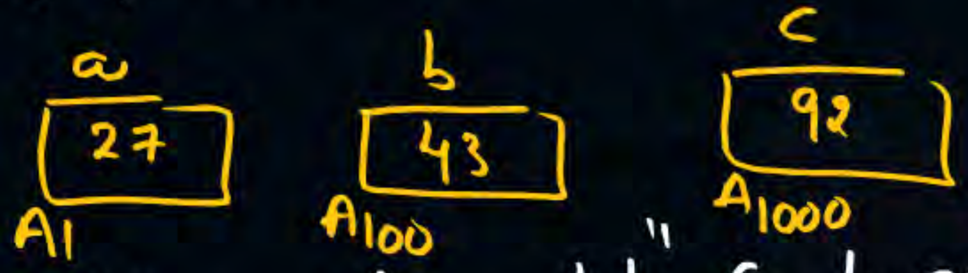
① Printf(" Message or string"); //Not accessing any data from memory.

All in lower-case

Ex: Printf(" C longauge is gud longauge") // o/p: C longauge is gud longauge

② printf(" message format specifier(s)", variable Names); //Accessing any data from Memory

Ex: int a=27, b=43, c=92;



Let 2Bytes Per Integer

printf(" a value = %d b value = %d, c value = %d", c, b, a);

Seperator

format Specifier

d == decimal Integer

o/p:

a value = 92

b value = 43

c value = 27

Topic : I/O Functions - 2

Data	Format Specifier
Decimal Integer	%d, %i
Octal Integer	%O (letter O)
HexaDecimal Integer	%x (or) %X
Character	%c
Strings	%s
Unsigned Integer	%u
float (or) double	%f (or) %e
long double	%lf (or) %f
long int	%ld

Topic : I/O Functions - 2

Printing Decimal Integers (By default all integers)

```
int P= 107;                              o/p

Printf(" P= %d", P);                     P=107
Printf(" P= %i", P);                     P=107
Printf(" P= %o", P);                     P=154
Printf(" P= %x", P);                     P= 6b
Printf(" P= %X", P);                     P= 6B
Printf(" P= %f", P);                     P= 107.000000
Printf(" P= %c", P);                     P= k (Lower Case alphabet, ASCII character)
Printf(" P=%7d", P);                     P=     107
                                           _ _ _ _ _ _ _
```

$$8 \,\underline{|\, 107}$$
$$8 \,\underline{|\, 13} \rightarrow 4$$
$$1 \rightarrow 5$$
$$(154)_8$$

$$16 \,\underline{|\, 107}$$
$$6 - 11$$
$$= 6b$$

% Sign number Specifier

- Sign → +ve: Right Justified/ aligned
- Sign → -ve: Left aligned
- number → minimum No.of characters to be reserved for displaying o/p value

```
Printf("P= %-7d", P);
          P= 107
             _ _ _ _ _ _ _
Printf("P= %07d", P);
          P= 0000107
             _ _ _ _ _ _ _
```

Topic : I/O Functions - 2

Printing Octal Integers (Zero Prefix) [Range : 0 to 7]

```
int P=0107;   ( int P=0183; //Error, octal
                                does not have 8 )

Printf("P=%o",P);      // P=107
Printf("P=%d",P);      // P=71
Printf("P=%f",P);      // P=71.000000
Printf("P=%c",P);      // P=G (71 Equivalent ASCII char)
Printf("P=%x",P);      // P=47
Printf("P=%7o",P);  //P= ____107
Printf("P=%7d",P);  //P= _____71
Printf("P=%-7o",P); //P=107____
```

$$(107)_8 = \begin{array}{ccc} 2 & 1 & 0 \\ \hline \boxed{1} & \boxed{0} & \boxed{7} \end{array}$$

$$7\times 8^0 + 0\times 8^1 + 1\times 8^2$$

$$= 7+0+64 = (71)_{10}$$

$$(71)_{10} = (\quad\quad)_{16}$$

$$16 \underline{\big| 71}$$

$$4 - 7$$

Topic : I/O Functions - 2

Printing Hexa Decimal Integers (Prefix: Zero x)

```
int P= 0x3A;
Printf("P= %x", P);   // P=3a
Printf(" P= %X", P);  // P=3A;
Printf(" P= %d", P);  // P=58
Printf(" P= %o", P);  // P=72
Printf ("P= %c",P);   // P= 58 Equivalent ASCII char (: Colon)
Printf ("P=%f",P);    // P=58.000 000
Printf(" P= %7x",P);  // P=     3a
                      //   -------
Printf(" P= %-7d",P); // P=58
                      //   -------
```

$$(3A)_{16} = \begin{array}{|c|c|} \multicolumn{1}{c}{1} & \multicolumn{1}{c}{0} \\ \hline 3 & A \\ \hline \end{array}$$

$$A \times 16^0 + 3 \times 16^1$$

$$= A + 48$$

$$= (58)_{10}$$

$$(58)_{10} = (\quad)_8$$

$$8 \underline{\big|\, 58}$$
$$\quad 7 - 2$$

Topic : I/O Functions - 2

Printing characters

```
Char ch= 'y';

Printf(" ch= %c", ch); // ch=y
Printf(" ch= %d", ch); // ch=121 (ASCII value)
Printf(" ch= %f", ch); // ch= 121.000 000
Printf(" ch= %o", ch); // ch= 171 (Decimal's Equivalent
                                           octal)
Printf(" ch= %x", ch); // ch= 79
Printf(" ch= %7c", ch); // ch= _ _ _ _ _ _ y
Printf(" ch= %-7c", ch); // ch= y _ _ _ _ _ _
```

```
8 | 121
8 | 15 - 1
     1 - 7
```

```
16 | 121
      7 - 9
```

Printing Strings

```
Char Str[10] = "GATE EXAM";

Printf(" String is %s", Str);
  // String is GATE EXAM
Printf(" String is %12s", Str);
  // String is _ _ _ G A T E _ E X A M
Printf(" String is %-12s", Str);
  // String is G A T E _ E X A M _ _ _
Printf(" String is %12.3s", Str);
// String is _ _ _ _ _ _ _ _ _ G A T
```

Topic : I/O Functions - 2

float Precision = 6 digit
double " = 13 digit (15 digit)
Long double " = 18-digit

Printing float values

```
float P= 17.215;

Printf(" P= %f", P);     // P= 17.215000

Printf(" P= %d", P);     // P= 17

Printf(" P= %c", P);     // P= Decimal Equivalent
                         //    ASCII char

Printf(" P= %o", P);     // P= 21

Printf(" P= %x", P);     // P= 11

Printf(" P= %7f", P);    // P= _ 1 7 . 2 1 5

Printf(" P= %7.2f", P);  // P= _ _ 1 7 . 2 1
```

Two digit Precision

-

```
Printf ("GATE", "EXAM", "2024");    o/p: GATE

Printf("GATE" "EXAM" "2024");    o/p: GATEEXAM2024
```

Terminator

-

```
int i=1, j=2, k=3;

Printf(" %d %d ", i, j, k);    o/p: 1 2

Printf(" %d, %d, %d, %d", i, j, k);    o/p: 1, 2, 3, Garbage value
```

scanf( ) : formatted input function

Prototype : Unsigned int scanf ($\underbrace{\text{Character Pointer}}_{\text{string as argument}}$);

Syntax: scanf("format specifier(s)", &variable1, &variable2 ....)

& = address of

```
int a, b;
char i, j;
float x, y;
scanf("%d %d %c %c %f %f", &a, &b, &i, &j, &x, &y);
```

⇒ Run time input (or) Dynamic input.

- scanf( ) accepts input values at run time and returns No. of inputs (format specifiers) accepted.

Ex:

```
int a, b;
float i, j;

b = scanf("%d %f %f", &a, &i, &j);
Printf("%d", b);
```

O/p: 3

i/p: 456
23.176
113.19024

2 mins Summary

- Printf( )
- Printing Numbers
- Printing Characters
- Printing float values
- Scanf ( )

THANK - YOU