Chapter 3 :- Input & output in C

- Provided using library functions called stdlib.h
- putchar / printf OR getchar / scanf
- other functions from other libraries are math.h , string.h, stdlib.h.

3.1 : Conversion specifier :

- used with printf & scanf to specify type & properties of parameter
- could be used with long (l) or short (h) specifiers as well.

3.2 : Reading input data :

** Note

Note :- with %s as conversion specifier no need for &

- scanf (" control string ", address);
- symbols or specific whitespace characters can be included as delimiter like
  scanf (" %d - %d - %d ", &date, &month, &year)

" to print % used %% "

Note **
- space between conversion specifier is ignored !

3.3 :- Writing output data :

- printf (" control string ", variable)
- conversion specifiers as control string
- cout be also used just for printing
- to print ' , " , \ → use \' , \" , \\ (use backslash)

3.4 : Formatting Input And output :

Like how data is used in scanf printf using conversion specifiers
we can format data using format specifiers

3.4.1 :- Format for integer input :

%wd

max field width of i/p data

conversion specifier

scanf (%2d %2d, &a , &b) →

(a) when i/p data ≤ w unaltered and printed

(b) when i/p data > w untered and printed

e.g.: scanf (%2d %3 d, &a , &b) →

1 25 → unaltered → output 1 , 25

100 293 → altered → output 10 , 00

3.4.2 :- Format for integer output :

%wd → if o/p ≤ w → unaltered and leading blanks are added imp

→ if o/p > ω → then also printed correctly and not altered !

3.4.3 :- Format for floating point numeric inputs

%ωf   ω → field width including decimal point and data before & after

f → floating point conversion specifier

scanf ("%3f %4f", &x , &y)

(a) when i/p is less than ω , unaltered

i/p → 2.9   3.2 → output 2.9   3.20

i/p

(b) when o/p is greater than ω , altered

i/p → 2.93   3.237 → output 2.9   3.00

3.4.4 :- Format for floating point numeric inputs :

%ω.nf   ω – field width   f → conversion specifier

n – no of decimals after decimal point

printf ("x= %4.1f , y= %7.2f", x, y)

8   5.9

	8	.	0

			5	.	9	0

25.3   1635.92

2	5	.	3

1	6	3	5	.	9	2

15.23 &   65.888721

1	5	.	2

		6	5	.	8	9

3.4.5 :- Format for string input :

%ωs   ω → total number of characters to be stored in string

→ format conversion specifier

if scanf ("%3s", str) with input " Kiran "

output will be 'K', 'I', 'R', '\0'   automatic

3.4.6 :- Format for string output

%ω.ns   ω → total number of characters to be stored in string

s → format conversion specifier

n → only first n characters will be displayed.

Rest (ω–n) leading blanks will be added !

with less width still printed

printf ("%3f", kiranjojare) :-

K	I	R	A	N	J	O	J	A	R	E

printf ("%7f", Pappu) :-

		P	A	P	P	U

leading space

only first three letters used

printf (%.3f, "pappu") :-

P	A	P

No leading space

only 7-3 = 4 leading spaces

printf (%7.3f, suresh kumar) :-

				S	U	r

with leading space

Imp**

2.4.7: Suppression character in scanf():

* → suppression character in c

any conversion specifier with * will be ignored

`scanf("%d %*d ;%d", &a, &b, &c)`

i/p 20 35 29   a = 20   b = garbage / ignored   c = 29

not assigned

instead of 35 garbage val will be assigned