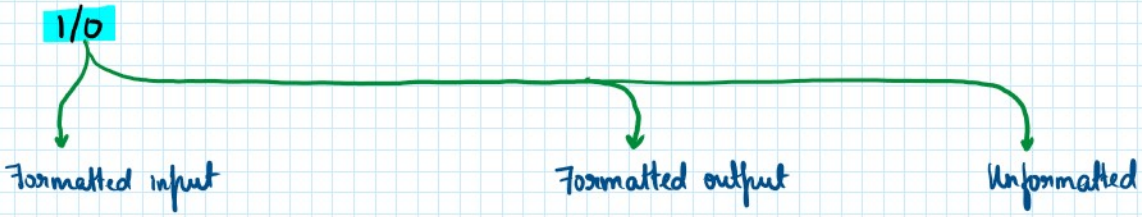
2. Input/Output Functions

07 February 2024 10:13



printf()

what it's this — format string

```
int printf(const char* format, arg1, arg2...)
```

- Predefined function declared in stdio.h
- Variadic (?) function : takes variable no. of arguments

NOTE: To print ASCII value of a character

```
char a = 'A';
printf("ASCII value: %d", a);
```

Format string

%[flags][field_width][.precision]conversion_character

- flags: optional argument that specifies output justification
- field_width: optional minimum width of o/p field
- .precision: optional arg that specifies maximum no. of characters to print

NOTE: Fractional values

- %f → 6 decimal places
- %e → exponential form
  - $10^{-4} <$, $>$ specified para... ?
- %g → strips trailing zero

Field width

%5d

"5" here specifies minimum no. of spaces that should be occupied. Note that int values are right justified by default.

Flags

1. "-" : left justifies
2. "+" : Displays the sign of a number
3. "0" : Displays zeroes instead of spaces for any minimum spaces not filled
4. "#" : Displays prefixes for octal, hexadecimal...

Precision

%.3f

This ensures the float value has 3 decimal places. If it has more than 3, then it is rounded off.

NOTE: Predefined macros

INT_MAX : max signed int value
INT_MIN : min " " "
UINT_MAX : max unsigned " " (%u)
CHAR_MAX : max char value (signed?)
CHAR_MIN
FLT_MAX
FLT_MIN
USHRT_MAX

and so on

DBL_MAX
DBL_MIN
LONG_MAX
LLONG_MAX
LONG_MIN
LLONG_MIN
SHRT_MAX
SHRT_MIN

scanf()

```
int scanf(const char* format, ...)
```

```
int a, b;
scanf("%d%d", &a, &b)
printf("%d\n%d", a, b)
```

`&a, &b` → references variable address

output:

```
2
2
3
```

3 → scanf → printf