

Chapter - 18 : Console input/output

getchar() (I/P) → stdio.h

→ C = getchar(); → one character

putchar() (O/P) → stdio.h

→ putchar(char variable);

putchar(c);

getch()

stdio.h X
portable

conio.h

↓
console input
output

scanf

scanf("formatting info", arg1, arg2, ..., argn);

scanf(" " , ⓧarg1, ⓧarg2, ..., ⓧargn);



Figure - 18.2

int \rightarrow %d

char \rightarrow %c

unsigned \rightarrow %u

octal \rightarrow %o

hexadecimal \rightarrow %x

double \rightarrow %lf

long double \rightarrow %Lf

format
specifier

specify the width of the input/output.

" " precision of the real number.

→ `scanf("%3d %3d %3d", &a, &b, &c);`
 1 2 3 a b

123 456 789

→ 123456789

1234 5678 9

ignored

→ printf("%-3d", q);

<u>a</u>	<u>b</u>	<u>c</u>
1	2	3
123	456	789

123 456 789

123 4 567

night
justified

[illegible]

scanf("%3d %5f %c", &a, &b, &c);

10 256.875 T
↓
C

<u>a</u>	<u>b</u>	<u>c</u>
10	256.87	T
✓	X	X
	<u>256.8</u>	<u>7</u>

printf("format specifiers", <variable names>);

`printf("%d", printf("%d", a));`

Diagram illustrating the execution of the code:

- The inner `printf("%d", a);` is annotated with a '4' above it, indicating it returns the value 4.
- The outer `printf("%d", ...);` is annotated with a '3' below it, indicating it returns the value 3.
- An arrow points from the return value of the inner function to the output `100 3`, where `100` is underlined.

`a = 100`

(100)

(100)

The default ^{return} value of a function in C is an integer.