

# C in One Shot

## Part - 1

according to him "programming is automation"

# Flow - 1

- 1) Basic Printing, \n wali cheez, printing numbers also with + and -.
- 2) Variables, printing variables, int, float, and +,-,\*,/ of integers.
- 3) Variables naming rules.
- 4) Comments
- 5) Taking Input ; means line finished
- 6) Modulus Operator
- 7) Float to int, int to float escape sequence (\n)
- 8) Hierarchy
- 9) Char and ASCII

# Basic program in C

```
#include<stdio.h>

int main(){
    printf("hello world");
    return 0;
}
```

[1-1]

# How to move in next line?

Example :

```
printf("Hello PW");
```

```
printf("Hello CW");
```

Output will be :

[1-2]

Hello PWHello CW

# Use of escape sequence '\n'

Example :

```
printf("Hello PW");
```

```
printf("\n");
```

```
printf("Hello CW");
```

[1-3]

Output will be :

Hello PW

Hello CW

# Use of escape sequence '\n'

Predict the output :

```
main(){  
    printf("nn\n\nnnn\n");  
    printf("nn/n/nnn/n");
```

```
}
```

```
# include <stdio.h>  
int main(){  
    return 0;  
}
```

Output

- nn
- [1-10]
- nn
- nn/n/nnn/n

# Variables and their Declaration

Let us focus on int data type as of now.

## 1) Variables as containers :

```
# include <stdio.h>
int main()
{
    printf("Hello");
    int n;
    n = 3;
    n = 5;
    printf("%d", n)
```



Variables      Literals

x  
n, y

'\n'  
'%d'

printf("x");  
→ text

printf("%d", n);

# Printing Variables in C & Updation of Variables

```
int x = 5;           ↗ int n;  
                     n=5;
```

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

```
x = 7;
```

```
printf("%d", x); [1-19]
```

```
x = x + 6;
```

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

```
x = x - 20; → n = 13 - 20
```

```
printf("%d", x); n = -7
```

-7  
n

Output

5 7 13 -7

# Arithmetic operations on int data type

```
int x = 5;
```

'+' '-'

```
int y = 2;
```

%d → int

```
printf("%d", x+y);
```

$$5/2 = 2$$

```
printf("%d", x-y);
```

```
printf("%d", x*y);
```

```
printf("%d", x/y);
```

└

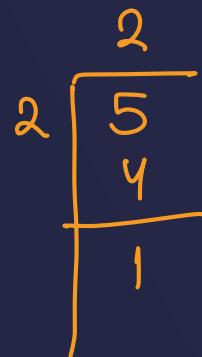
```
int x = 4;
```



```
int y = 2;
```



```
int z = x+y;
```



$5/2 \rightarrow 2.5$  ]  
int  $x = 5/2$ ; ]  $\times$

$5/2 \rightarrow 2$

# Increment – Decrement operators

```
int x = 5;  
x++;  
printf("%d", x);  
x--;  
printf("%d", x);  
++x;  
printf("%d", x);  
--x;  
printf("%d", x);
```

# Float data type

→ Real Numbers

```
float x = 3.1;
```

`%d` → int

`%f` → float

# Arithmetic operations on float data type

```
float x = 5;
```

```
float y = 2;
```

```
printf("%f", x+y); → 7
```

```
printf("%f", x-y); 3
```

```
printf("%f", x*y); 10
```

```
printf("%f", x/y); 2.5
```

```
float n;  
n = 5;
```

$5/2 \rightarrow \text{integer} \rightarrow 2$

$\text{float } z = 5/2 ;$   
 $\rightarrow \text{float } z = 2 ;$

$5.0/2 \rightarrow 2.5$

$5/2.0 \rightarrow 2.5$

$$V = \frac{4}{3}\pi r^3 \quad [1-33]$$

$$V = 4 * 3.14 * r * r * r / 3$$

```
int r=5
```

```
float v = 4 * 3.14 * 5 * 5 * 5 / 3
```

H.W./C.W. WAP to display area of circle with given radius. [1-34]

# Example : Calculating percentage of 5 subjects

```
float x1 = 90; // x1 can be physics
float x2 = 91; // x2 can be chemistry {BODMAS}
float x3 = 92; // x3 can be maths
float x4 = 93; // x4 can be english
float x5 = 94; // ohh wait comments ke baare me to bataya hi nahi xD
float percent = (x1 + x2 + x3 + x4 + x5)/5;
printf("%f",percent);
// change the marks and run each time
```

H-W. Print percentage of 4 Subjects whose marks are out of 40.  
[haven't done this hw]

# Example : Calculating Area of a Circle H.W.

```
float radius = 5;           float r = 5;  
float pi = 3.1415;          int n, int y int z  
float area = pi*radius*radius;  
printf("%f",area);          int a
```

# Variable Naming rules

- 1) Variables can start from an alphabet or underscore \_ .
- 2) Special characters except \_ are not allowed.
- 3) Some particular keywords are not allowed.
- 4) Commas or blanks are not allowed.

*auto  
Auto double int break extern enum unsigned while  
case sizeof for const static long continue float  
else signed do short switch char volatile default  
goto struct if union return void register typedef*

*float macbook ; [32 key words]*

*int Mac ;*

# Variable Naming rules – Examples

Q. Which of the following are invalid variable names and why?

BASICSSALARY

\_basic

basic-hra

#MEAN

group.

422

population in 2006

over time

mindovermatter

FLOAT

hELLO

queue.

team' svictory

Plot#3

2015\_DDay

float != FLOAT

# Example : Calculating Simple Interest

```
float p,r,t,si;  
p = 100;  
r = 10;           [1-37]  
t = 2;  
si = (p*r*t)/100;  
printf("%f",si);
```

$$SI = \frac{PRT}{100}$$

$$\frac{152 \times 10 \times 2}{100} = 30.4$$

# Taking input // Let us take a simple example from User.

```
int x;  
  
printf("Enter a number\n");  
  
scanf("%d",&x); // user will give 'x' a value.  
  
int y = x*x;  
  
printf("square of number that you gave is %d",y);
```

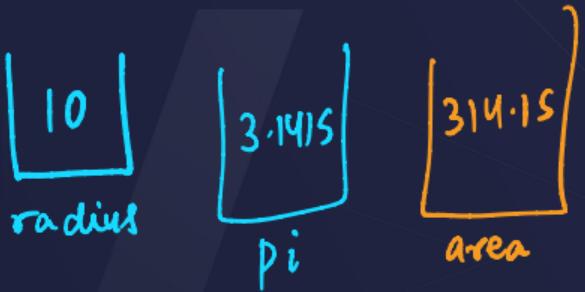
```
#include<stdio.h>
int main(){
    float radius;
    scanf("%f",&radius);
    float pi = 3.1415;
    float area = pi * radius * radius;
    printf("The area of circle is : %f",area);
    return 0;
}
```

Run

10

The area of circle is : 314.15

Dry Run:



Output

10

The area of circle is : 314.15

```
#include<stdio.h>
int main(){
    float principal,rate,time,si; // 4 dabbe ci
    printf("Enter Principal : ");
    scanf("%f",&principal);
    printf("Enter Rate : ");
    scanf("%f",&rate);
    printf("Enter Time : ");
    scanf("%f",&time);
    si = (principal*rate*time)/100;
    printf("Your simple interest is : %f",si);
    return 0;
}
```

### Output :

Enter Principal : 100

Enter Rate : 20

Enter Time : 3

Your simple interest is : 60

Scanf ("%d", &x)  
[why use & ]

100      20      3      60  
principal    rate    time    Si

```
int x = 5;
```

```
float y = 6.1;
```



$\&x \rightarrow$  address of x

```
int x;
```

```
scanf("%d", &x);
```



# Taking input // SUM of 2 given numbers

```
int x;  
printf("Enter first number\n");  
scanf("%d",&x); // user will give 'x' a value.  
  
int y;  
printf("Enter second number\n");  
scanf("%d",&y); // user will give 'y' a value.  
  
int sum = x+y;  
printf("sum of the numbers that you gave is %d",sum);
```

3  
x      6  
y

9  
Sum

Homework :

Output  
Enter 1<sup>st</sup> No.  
En — —

# Taking Input

Predict the output :

```
main(){  
    int p,q;  
  
    printf("Enter values of p and q");  
    scanf(" %d %d ",&p,&q);  
    printf ("p = %d q =%d", p, q);  
}
```

30  
P      40  
Q

Output

Enter Values of P & Q

30 40

P = 30 Q = 40

## 'Dry Run'

**Example :** Take two integers input, **a** and **b** :  $a > b$ , and find the remainder when **a** is divided by **b**. & print the remainder.

main() {

[1-45]

```
int a, b;
printf("Enter 1st No.");
scanf("%d", &a);
printf("Enter 2nd No.");
scanf("%d", &b);
int q = a / b;
int r = a - (b * q);
```

return 0;

$$\begin{array}{r} 7 \sqrt{39} \\ 35 \\ \hline 4 \end{array}$$

$$7 \times 5 + 4 = 39$$

int a = 39;

int b = 7;

int q = a/b; //  $q = 5$

Divisor  $\times$  Quotient + Rem = Dividend

$$\Rightarrow \text{Remainder} = \text{Dividend} - \text{Divisor} \times \text{Quotient}$$

$$\Rightarrow r = a - (b \times q)$$

## Operators:

+ , - , / , \* , %

$$\frac{2}{6} = 0$$

Modulo Operator  $\rightarrow$  %

a = 2;

r = a % b

$$38 \% 8 = 6$$

b = 6;

$$2 \% 2 = 2$$

( $a \% b = a$ )

[ $n > 2$ ]

[ $a < b$ ]

$$8 \overline{)38}$$

$$\begin{array}{r} 32 \\ \hline 6 \end{array}$$

## float to int and int to float :

Ques : Take integer as input and print half of the number.

$\uparrow$  positive

Ques : Take float input and print the fractional part of the real number.

$$\{n\} = n - [n]$$

1, 2, 3, 4, 5, 6

$$'3.14' \rightarrow 0.14$$

G.I.F.

$[x]$

$$x = 7.1$$

$$[x] = 7$$

$$'5.7' = 5 + 0.7$$



0.7

```
float x; →
```

```
scanf("%f", &n);
```

```
int y;
```

```
y = n;
```

```
float z = n-y;
```

float      int

5.7  
n

5  
y

0.7  
z

Output

0.7

Homework: Code it Yourself

# Hierarchy of operators

BODMAS → B, O, D/M, A/S

```
int i = 2 * 3 / 4 + 4 / 4 + 8 - 2 + 5 / 8 ;
```

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

PROG

$$(2^3)/4 = 6/4 = 1$$

$$(2^3)/4 = 6/4 = 1$$

$$2^*(3/4) = 2^* 0 = 0$$

MATHS

$$(2 \times 3)/4 = 6/4 = 1.5$$

$$2 \times (3/4) = 2 \times 0.75 = 1.5$$

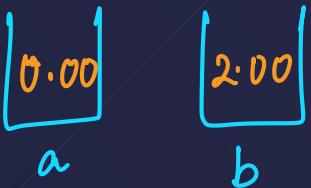
$$i = 1 + 1 + 8 - 2 + 0 = 8$$

8

# Try This!

Predict the output :

```
int main(){
    int i = 2, j = 3, k, l;
    float a, b;
    k = i / j * j;  $(2/3)^* 3 = 0^* 3 = 0$ 
    l = j / i * i;  $= (3/2)^* 2 = 1^* 2 = 2$ 
    a = i / j * j;  $(2/3)^* 3 = 0^* 3 = 0$ 
    b = j / i * i;  $= 2$ 
    printf("%d %d %f %f", k, l, a, b);
}
```



# \* Try This!

Predict the output :

```
int main(){  
    float a = 5, b = 2;  
    int c;  
    c = a % b;  
    printf("%d", c);  
}
```

$$\begin{aligned}a &= 5.000000 \\b &= 2.000000\end{aligned}$$

L  
C

LX

int → +, -, \*, /, %  
float → +, -, \*, /

Output  
0, Error

# char data type

char ch = 'a'; → %c  
variable



char n = 'a';

'a', 'b', 'c' . . .

'A', 'B', 'C' , - . . 'Z'

int , float  
integer → real number/decimal  
%d ↓  
format specifier  
%f

char ch = '#';



# ASCII values → Important topic

char ch = 'a';

$$a = 97$$

$$A \rightarrow 65$$

$$B \rightarrow 66$$

$$C \rightarrow 67$$

$$D \rightarrow 68$$

$$E \rightarrow 69$$

$$F \rightarrow 70$$

$$G \rightarrow 71$$

$$z = 90$$

$$b = 98$$

$$c = 99$$

$$d = 100$$

$$z = 122$$

$$\# \rightarrow 35$$

$$! \rightarrow 33$$

$$+ \rightarrow$$

$$=$$

$$-$$

$$-$$

$$@ \rightarrow 64$$

$$* \rightarrow 42$$

$$/ \rightarrow 36$$

$$\$ \rightarrow 36$$

If - Else

# MCQ Time !

## MCQ 1

Character → single digit

'a', 'b' '#'

char ch = 'a##';

Which of the following is NOT a character constant

- (1) 'Thank You' ~~wRONG~~
- (2) 'Enter values of P, N, R'
- (3) ~~123.56E-03~~
- (4) All the above

## MCQ 2

In  $(b = 6.6 / a + 2 * n;)$  which operation will be performed first?

- (1)  $6.6 / a$
- (2)  $a + 2$
- (3)  $2 * n$
- (4) Depends upon compiler

## MCQ 3

Which of the following statements is false

- (1) Each new C instruction has to be written on a separate line
- (2) Usually all C statements are entered in small case letters
- (3) Blank spaces may be inserted between two words in a C statement
- (4) Blank spaces cannot be inserted within a variable name

```
int n = 4;  
n = n + 7;
```

## MCQ 4

If a is an integer variable,  $a = 5 / 2$ ; will return a value

- (1) 2.5
- (2) 3
- (3) 2
- (4) 0

int a = 5/2;

## MCQ 5

BODMAS

The expression,  $a = 7 / 22 * ( 3.14 + 2 ) * 3 / 5$ ; evaluates to

- (1) 8.28
- (2) 6.28
- (3) 3.14
- (4) 0

$$7/22 * 5.14 * 3/5$$

$$0 * 5.14 * 3/5$$

$$0 * 3/5$$

$$0/5$$

$$0$$

## MCQ 6

*int*  
The expression,  $\uparrow a = 30 * 1000 + 2768$ ; evaluates to

- (1) 32768
- (2) -32768
- (3) 113040
- (4) 0

$$30000 + 2768$$

$$a = 32768$$

Short a = 32768;

- 32768

Data Types

$$2^{16} = 2^{10} \cdot 2^6$$

$$= 1024 \rightarrow 64$$

int v/s short  $\rightarrow$  Chhota dabba

int  $x = 3;$

float  $y = 3.14;$

char  $ch = 'A';$

short  $z =$  int  $\rightarrow$  Short  $\rightarrow$  2 bytes  $\rightarrow$  16 bits

long  $x = 32768;$  8 Bits  $\rightarrow$  1 byte

long long  $x =$

$n$  bits  $\rightarrow 2^n$  numbers

16 bits  $\rightarrow 2^{16}$  numbers store

b  $\curvearrowright$  Kb  $\curvearrowright$  Mb  $\curvearrowright$  Gb  $\curvearrowright$  Tb  
 1000 1000 1000 1000  
 $2^{16} = 65536$

int data type  $\rightarrow$  4 bytes  $\rightarrow$  32 bits

$\rightarrow 2^{32}$  numbers  
 $\downarrow$

long  $\rightarrow$  8 bytes  $\rightarrow$  64 bits  
 $2^{64}$  numbers

short  $-2^{15}$  to  $2^{15} - 1$

int  $-2^{31}$  to  $2^{31} - 1$

long long  $-2^{63}$  to  $2^{63} - 1$

Char  $\rightarrow$  1 byte  $\rightarrow$  8 bits  $\rightarrow 2^8$  numbers  $\rightarrow$  256 numbers  
 $\downarrow$

## MCQ 7

BODMAS → % /, ^ +, -

The expression  $x = 4 + 2 \% - 8$  evaluates to

(1) -6

$$\Rightarrow x = 4 + 2 \div 8$$

(2) ✓ 6

$$2 \% 8 = 2$$

(3) 4

(4) None of the above

5% 2

$$\begin{array}{r} 2 \\ \overline{)5} \\ \underline{4} \\ 1 \end{array}$$

$$\sqrt[8]{2}$$

$$2 \% - 8 = 2 \% 8 = 2$$

## MCQ 8

What will be the value of d if d is a float after the operation  
 $d = 2 / 7.0$ ?

- (1) 0
- (2) 0.2857
- (3) Cannot be determined
- (4) None of the above**

$$\begin{array}{r} \text{float } d = 2/7.0; \\[1ex] 7 \overline{)2.0} \\[-1ex] 14 \\[-1ex] \hline 60 \\[-1ex] 56 \\[-1ex] \hline 40 \\[-1ex] 35 \\[-1ex] \hline 50 \\[-1ex] 49 \\[-1ex] \hline \end{array}$$