C in One Shot

Part – 3

Loops in One Shot

What and Why?

loops → baar baar → repetition

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

DRY → do not repeat yourself

initialization condition updation/increment

```
for(int i=1;i<=5;i=i+1){
    printf("Hello PW\n");
}
```

Output

- Hello PW
- Hello PW
- Hello PW
- Hello PW
- Hello PW
-

Steps :

1) Condition check

2) Loop ke andar aao

3) Updation

i = i + 1 ⇔ i++

```c
for(int i=1;i<=10;i=i+2){
    printf("Hello World\n");
}
```

H.W. → i = i+3;

- Hello World
- Hello World
- Hello World
- Hello World
- Hello World

Step 1: Check condition

Step 2: Go inside loop

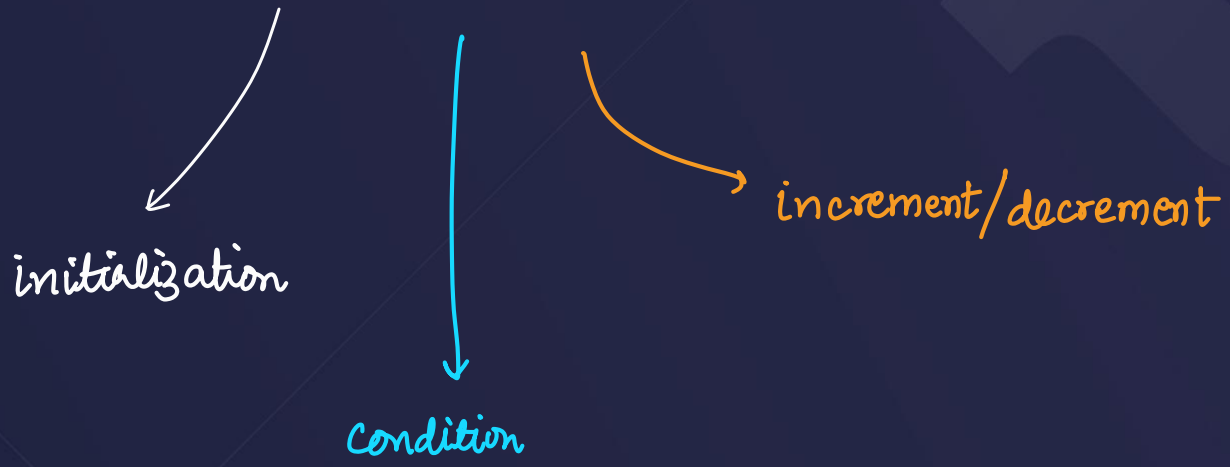
Step 3: Updation

For Loop

```
for(int i = 1; i<10; i++){

    // code

}
```

Ques : Print hello world ‘n’ times. Take ‘n’ as input from user

How for loop works : the various parameters.



Ques : Print numbers from 1 to 100

in different lines

```
for(int i=1;i<=100;i++){
    printf("%d ",i);
}
```

Output:

1
2
3
4
.
.
.
.
.
100

i = ~~1~~ ~~2~~ 3

Output

1 _ 2 _ 3 _
4 _ 5 _ 6

Ques : Print all the even numbers from 1 to 100

```
if( i % 2 == 0){
    printf("%d ", i);
}
```

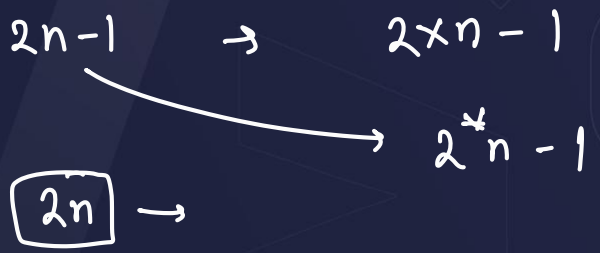
HW : Print all the odd numbers from 1 to 100

Ques : Print the table of 19.

19, 38, 57 190

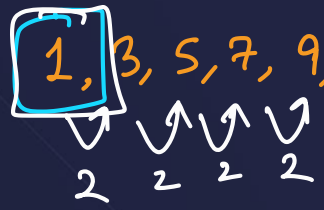
HW : Print the table of 'n'. Here 'n' is a integer which user will input.

```
int n;
scanf("%d",&n);
```



Maths

Ques : Display this AP - 1,3,5,7,9.. upto 'n' terms.



$a_n = 1 + (n-1) \cdot 2$

$= 1 + 2n - 2 = 2n - 1$

$a, a+d, a+2d, a+3d \ldots a+(n-1)d$

- a : first term
- d : common difference
- $a+(n-1)d$: n^{th} term

```
int i = 1 ; i <= 2n-1 ; i = i + 2
```

HW : Display this AP - 4,7,10,13,16.. upto 'n' terms.

4, 7, 10, 13, 16 . . . . .

3

$a = 4$

$d = 3$

$a_n = a + (n-1)d$

$a_n = 4 + (n-1)3$

$= 4 + 3n - 3$

$= \boxed{3n + 1}$

```
for(int i=4; i<=3*n+1; i=i+3)
{
    printf("%d ", i);
}
```

3

Ques : Display this GP – 1,2,4,8,16,32,.. upto 'n' terms.

×2 ×2 ×2 ×2 ×2

GP → $a_n = ar^{n-1}$

$= 1 \cdot 2^{n-1}$

```
int a = 1;
for(int i = 1; i <= n ; i++) {
    printf("%d ", a);
    a = a * 2;
}
```

HW : Display this GP - 3,12,48,.. upto ‘n’ terms.

Ques : Display this AP – 100,97,94,..upto all terms which are positive.

$100, 97, 94, 91, 88 \ldots$

$-3 \quad -3 \quad -3$

$a_n = 100 + (n-1)(-3)$

$= 100 - 3n + 3$

$= \boxed{103 - 3n}$

problem → no. of terms → ??

$103 - 3n > 0$

$\Rightarrow 3n < 103$

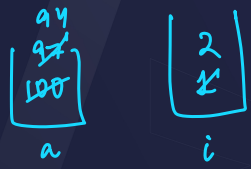
$\Rightarrow n < \frac{103}{3} \rightarrow n < 34.33$

$\boxed{n_{max} = 34}$

```c
// 100 97 94 ....
int a = 100;
for(int i=1;a>0;i++){
    printf("%d_",a);
    a = a - 3;
}
return 0;
```

Output

100 97 94 91 88



HW : Display this GP - 100,50,25,.. upto 'n' terms.

100 50 25 12.5

$\times \frac{1}{2}$ $\times \frac{1}{2}$ $\times \frac{1}{2}$

float

```
float a = 100;
for(int i=1 ; i<=n; i++){

}
```

Loop ke andar jo bhi daalo vo sab repeat hota hai !!

```
for (      ) {
    line 1;
    line 2;
    line3;
}
```

Break;

↳ It is used to terminate

Ques : WAP to check if a number is prime or not.

25 → composite

Prime → n

n & 1

2, 3, 5, 7, 11, 13, 17, 19, 23

Steps: 2 to 24

25 % 2 → X

25 % 3 → X

25 % 4 → X

25 % 5 ⇒ 0

```c
int n;
printf("Enter a number : ");
scanf("%d",&n);
for(int i=2;i<=n-1;i++){
    if(n%i==0){
        printf("the given number is composite\n");
    }
}
return 0;
```

Rounds → 'Iterations'

i → 2 to 24

23 baar loop

n = 25

~~i=2~~

~~i=3~~ ✓ 5

1, 5, 25

12 → 1, 2, 3, 4, 6, 12

Number → n

→ 2 to n-1

if(n%i==0) → composite

break;

0
a

→ if( ) → a = 1;

break;

↳ if (a == 0) prime

else composite

```c
int a = 0;
for(int i=2;i<=n-1;i++){
    if(n%i==0){ // i is a factor of n
        a = 1;
        break;
    }
}
if(a==0) printf("the given number is prime\n");
else printf("the given number is composite\n");
return 0;
```

n

↓

$i \to 2$ to $n-1$

$n=2$

↘ $i \to 2$ to 1

$i=2 \to n-1$

$n=1$

↘ $i = 2$ to 0

Loop✗

Continue;

```
for (int i = 1; i <= 100; i++)
{
    if (i % 2 == 0) // even
    { // i is a factor of n
        continue; → Skip Karo us round Ko
    }
    printf("%d ",i);
}
```

i = 1 2 3 4

Output

1 3

Ques : WAP to print odd numbers from 1 to 100.

```
for (int i = 1; i <= 100; i++)
{
    if ( i % 2 != 0 ){   //odd
        printf(    )
    }
```

Homework : WAP to print all the even numbers from 1 to 100, using continue statement.

While Loop

```
int i = 0;
while(i<10){
    // code
    i++;

}
```

```
for (int i=1 ; i<10 ; i++)
{
    printf("%d", i);
}
```

Infinite Loops :

```
int i = 1;
while (i<10){
   printf("%d\n", i);
    i--;
}
```

true

i = 1 0 -1

1
0
-1
-2
-3

Do- While Loop

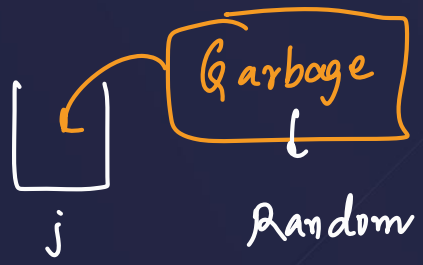
→ Useless
of
Semester

```
do {

    //code

} while ( another == 'y' ) ;
```

do {
≡
} while (condition);

Predict the output

```
main( ) {
        int j ;
        while ( j <= 10 ) {
            printf ( "\n%d", j ) ;
            j = j + 1 ;
        }
}
```



Predict the output

```
main( ) {
        int i = 1 ;
        while ( i <= 10 ) ;
        {
            printf ( "\n%d", i ) ;
            i++ ;
        }
}
```

Output

1
2
3
4
5
6
7
8
9
10

Predict the output

```
main( ) {
        int x = 1 ;
        while ( x == 1 ) {
            x = x - 1 ;
            printf ( "\n%d", x ) ;
        }
}
```

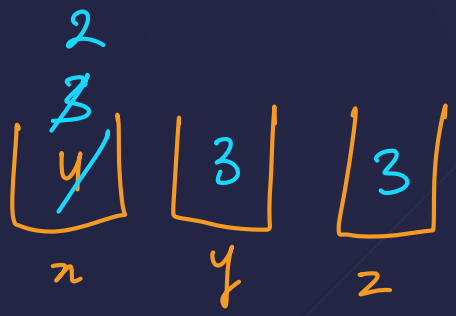
Output

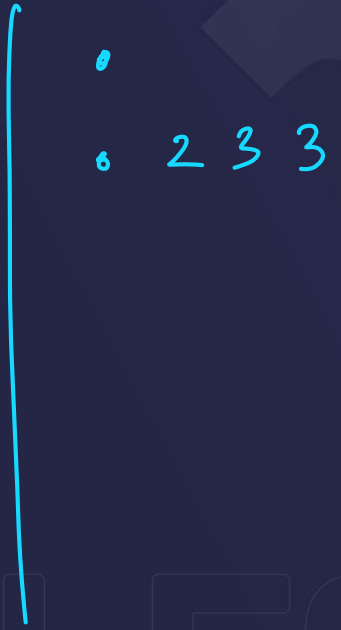
.

. 0

Predict the output

```
main( ) {
        int x = 4, y, z ;
        y = --x ;
        z = x-- ;
        printf ( "\n%d %d %d", x, y, z ) ;
}
```





x = 4
--x → x = 3

$x++ \Rightarrow x = x + 1;$ post

$++x \Rightarrow x = x + 1;$ pre

$x-- \Rightarrow x = x - 1;$

$--x \Rightarrow x = x - 1$

Predict the output

```
main( ) {
        int x = 4, y = 3, z ;
        z = x-- -y ;
        printf ( "\n%d %d %d", x, y, z ) ;
}
```

Output

3 3 1

Predict the output

'a' → 97 'b' 98

```
main( ) {
    while ( 'a' < 'b' )
        printf ( "\nmalyalam is a palindrome" ) ;
}
```

malayalam

Infinite loop

Output

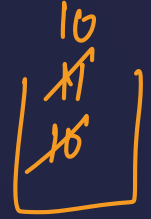
Predict the output

```
main( ) {
    int i = 10 ;
    while ( i = 20 )
        printf ( "\nA computer buff!" ) ;
}
```

Infinite Loop

Predict the output

```
main( ) {
    int i ;
    while ( i = 10 ) {
        printf ( "\n%d", i ) ;
        i = i + 1 ;
    }
}
```



Output

10
10
10
10
10

Infinite Loop

Predict the output

```
main( ) {
    float x = 1.1 ;
    while ( x == 1.1 ) {
        printf ( "\n%f", x ) ;
        x = x – 0.1 ;
    }
}
```

Output

0

1.1

Predict the output

```
main( ) {
    while ( '1' < '2' )
        printf ( "\nIn while loop" ) ;
}
```

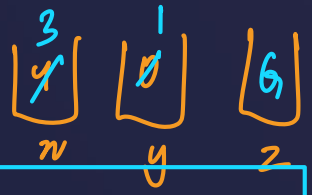
Always true

Terminate X

Infinite Loop

Predict the output

```
main( ) {
    int x = 4, y = 0, z ;
    while ( x >= 0 ) {
        x-- ;
        y++ ;
        if ( x == y )
            continue ;
        else
            printf ( "\n%d %d", x, y ) ;
    }
}
```



Output

-
- 3 1

Homework :

Predict the output

```
main( ) {
        int x = 4, y = 0, z ;
        while ( x >= 0 ) {
                if ( x == y )
                        break ;
                else
                        printf ( "\n%d %d", x, y ) ;
                        x-- ;
                        y++ ;
        }
}
```

Questions using Operators

+ - * /

Ques : WAP to count digits of a given number.

n = 19325

Hint : 1) / operator

2) 19325/10 = 1932

```
int count = 0;
n = n/10;
count++;
```

Condition → `n != 0` or `n > 0`

Ques : WAP to count digits of a given number.

```
int count = 0;
while(n!=0){
    n = n/10;
    count++;
}
```

n = 1234 123 12 1 0

Count = 0 1 2 3 4

Ques : WAP to print sum of digits of a given number.

n = 12345

Sum = 1+2+3+4+5 = 15

Hint

1) % operator & / operator

2) 1+2+3+4+5 = 5+4+3+2+1

```
Sum = 0;
```

```
Sum = Sum + lastDigit;
n = n/10;
```

Ques : WAP to print sum of digits of a given number.

Sum = 0; 5 9 12 14 15

n = 12345 1234 123 12 1 0

```
ld = n%10;   // ld = 5 4 3 2 1
sum = sum + ld;
n = n/10;
```

HW : WAP to print sum of all the even digits of a given number.

n = 12345

Sum = 2 + 4 = 6

Hint : Use the same code, (if condition)

Ques : WAP to print reverse of a given number. & store it.

n = 1234

r = 4321

Algorithm :

n
~~1234~~
~~123~~
~~12~~
~~1~~
0

r
~~0~~
~~1~~ 10
~~13~~ 130
~~132~~ 1320
4321

$r = r + (n \% 10)$

$r = r * 10$

$n = n / 10$

Hints

1) Abhi 2 ques

2) 10 se multiply

Condition → n > 0

or

n != 0

Ques : WAP to print reverse of a given number.

'Dry Run'

n = ~~1234~~ ~~123~~ ~~12~~ ~~1~~ 0

r = ~~0~~ ~~4~~ ~~40~~ ~~43~~ ~~430~~ ~~432~~ ~~4320~~ ~~4321~~ 43210

```
int r = 0;
while(n>0){
    r = r + (n%10);
    r = r*10;
    n = n/10;
}
```

swap

r = r/10;

HW : WAP to print the sum of given number and its reverse.

n = 1234 r = 4321

Sum = 1234 + 4321 = 5555

Ques : Print the sum of this series : 1 - 2 + 3 - 4 + 5 - 6... upto 'n'.

terms

Method - 1

Hint : loop , if-else

if n = 1000 lets say
then loop will run 1000 times

1 + 2 + 3 + 4 ... n terms.

$$\frac{n(n+1)}{2}$$

Ques : Print the sum of this series :
1 - 2 + 3 - 4 + 5 - 6... upto 'n'.

$(1-2)+(3-4)+(5-6)+(7-8)$ $\rightarrow n=8$

$-1 \quad + \quad -1 \quad + \quad -1 \quad + \quad -1$

$\Rightarrow \quad -4$

n is even, if (n%2==0)

Sum = -n/2 ;

Ques : Print the sum of this series : 1 - 2 + 3 - 4 + 5 - 6... upto 'n'.

if n is odd → if (n %2 != 0)

n = 7

(1 - 2) + (3 - 4) + (5 - 6) + 7

-1 + -1 + -1 + 7

⇒ -3 + 7

⇒ $-\frac{n}{2} + n$

Ques : Print the factorial of a given number 'n'.

$5! = 5 \times 4 \times 3 \times 2 \times 1$

$n!$ → n factorial

$10! = 10 \times 9 \times 8 \times 7 \times 6 \times 5 \times 4 \times 3 \times 2 \times 1$

$n! = n \times (n-1) \times (n-2) \times \cdots 3 \times 2 \times 1$

$0! = 1$

```
int product = 1;
for( i=1; i<=n; i++)
    product = product * i;
```

Ques : Print the factorial of a given number ‘n’.

```
int product = 1;
for(int i=1;i<=n;i++){
    product = product*i;
}
```

product = 1 1 2 6 24 120

i = 1 2 3 4 5 6

n = 5

Done in lecture

HW : Print the factorials of first 'n' numbers

n = 5

$1! = 1$

$2! = 2 \times 1 = 2$

$3! = 3 \times 2 \times 1 = 6$

$4! = 4 \times 3 \times 2 \times 1 = 24$

$5! = 5 \times 4 \times 3 \times 2 \times 1 = 120$

$n! = n \times (n-1)!$

$1! = 1$

$2! = 2 \times 1!$

$3! = 3 \times 2!$

$4! = 4 \times 3!$

$5! = 5 \times 4!$

$6! = 6 \times 5!$

Ques : Print the nth fibonacci number.

15th

1, 1 2 3 5 8 13 21 34 55 89 . . .

1 2 3 4 5 6 7 8 9 10 11

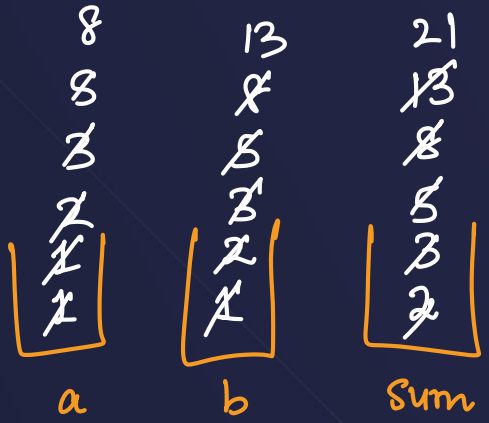
8^{th} term = 6^{th} term + 7^{th} term

Logic : 1) 3 variables a

```
a = 1;
b = 1;
sum = 0;
```

```
loop(){
    sum = a + b;
    a = b;
    b = sum;
}
```

Ques : Print the nth fibonacci number.



Sum = a + b

a = b

b = sum

n = 6 → 21 → 8th term

n → (n+2)th fibonacci

HW : Print first 'n' fibonacci numbers.

Output n = 7

The 1st fibonacci number is 1

The 2nd ___ ___ is 1

2

3

5

8

13

Ques : Two numbers are entered through the keyboard. Write a program to find the value of one number raised to the power of another.

$a, b \rightarrow a^b$

$a = 2, \quad b = 5 \rightarrow 2^5 = 32$

$2^5 = \underbrace{2 \times 2 \times 2 \times 2 \times 2}_{5 \text{ times}}$

```
int power = 1;
for ( ){
  power = power * a;
}
```

Ques : Two numbers are entered through the keyboard. Write a program to find the value of one number raised to the power of another.

```
int power = 1;
for(int i=1;i<=b;i++){
    power = power * a;
}
```

a = 2, b = 5 power = 1 2 4 8 16 32

$a^b = \underbrace{a \times a \times a \times a \times a \cdots}_{b \text{ times}}$

$2^n \rightarrow 2^1, 2^2, 2^3, 2^4, 2^5, 2^6 \cdots 2^n$

Ques : Write a program to print all the ASCII values and their equivalent characters of 26 alphabets using a while loop. (Capital case)

A → 65

B → 66

C → 67

D → 68

:

:

Z → 90

HW : Write a program to print out all Armstrong numbers between 1 and 500. If sum of cubes of each digit of the number is equal to the number itself, then the number is called an Armstrong number. For example, 153 = (1 * 1 * 1) + (5 * 5 * 5) + (3 * 3 * 3)

$1^3 + 5^3 + 3^3 = 153$