* **Loops** :- Loops are used to for executing a set of statements again and again by writting once.

1.) for ( ) 2.) while ( ) 3.) Do-while ( )

Syntax :- for (initialization ; Condition ; Updation)

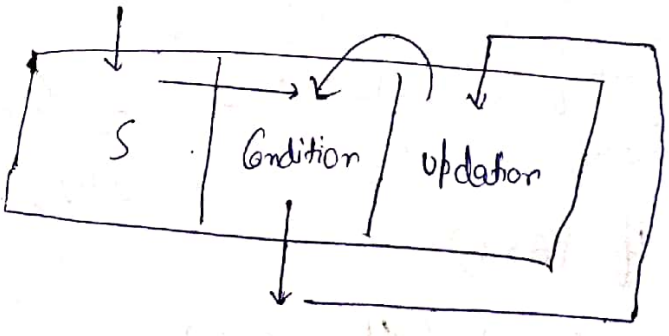
{

Loop body

}

==> Again & Again

==> only once




```
int main ()
{
    int C, N;
    Printf ("Enter value of N: \n");
    Scanf ("%d", &N);
    for (C=1 ; C<=N ; C=C+1)        C++ / ++C / C+=1
    {
        Printf ("%d. BCE \n", C);
    }
    Printf ("India : %d", C)  // 4, if n=3
}
```

Output

Enter N: 1 & 3

1. BCE
2. BCE
3. BCE

1, 2 (+1), 3 (+1)

India : 4

```
for (c=2 ; c<=8 ; c=c+2)
  {
   Printf (" %d", c);
  }
```

2
4
6
8

```
for (c=50 ; c>=20 ; c=c-10)
                  or
                  c-=10
   Printf ("%d \n", c);
```

50
40
30
20

// Billing :-

```
int main ()
{
  int N, Price, Sum=0;
   int c;
  Printf ("Enter No. of Products : ");
  Scanf ("%d", &N);  // 3
  for (c=1 ; c<=N, c++)
     {
      Printf ("Enter price of %d Product:", c);
       Scanf ("%d", & Price);
       Sum = Sum + price;
     }
   Printf (" total bill= %d", Sum); // 1300
}
```

Enter No. of Products : 3 → (N)
Enter Price of 1 Product : 500
——— 2 ——— : 200
——— 3 ——— : 600
total Bill = 1300

Home Work

Loops [Short Qus, 1-4]

Q-① :-

```
int main()
{
    int i, sum = 0;
    for (i = 2; i <= 4; i++)
    {
        if (i % 2 == 0)
            Printf ("%d", i);
            sum += i;
    }
    Printf ("\n sum = %d", sum);
}
```

⇒  Print → 2, 4

Sum = 2 + 3 + 4 = 9

Q-② :-

```
int main()
{
    int i;
    for (i = 4; i >= 2; i--)
        Printf ("%d", i+1);
        Printf ("\n hello");
}
```

⇒ 5, 4, 3
Hello

[∵ when, i = 4, Print → i+1 → 5
i = 3, Print → i+1 → 4
i = 2, Print → i+1 → 3]

Q-③

```
int main ()
{ int I, k;
  for (I=10 ; I<=30 ; I+10)
    {
      k = I/5 +2 ,
        Printf ("%d", k);
    }
}
```

⇒ Print → 10/5 + 2 = 4
20/5 + 2 = 6
30/5 + 2 = 8

Q-④

```
int main ()
{
  int n=5 , i , t;
    for (i=1 ; i<=3 ; i++)
      [
        t = n* i ;
          Printf ("%d", t);
      }
}
```

⇒ 5*1 = 5
5*2 = 10
5*3 = 15

Program 1 :- Program for factorial of N.

```
int main ()
{
    int N, m, factorial = 1;
    Printf ("Enter value of N : \n");
    Scanf ("%d", &N);
    for (m = 1; m <= N; m++)
        {
            factorial = factorial * m;
        }
    Printf (" factorial of %d is %d : ", N, factorial);
}
```

Program - 2 :- Program to print table of N.

```
int main ()
{
    int N, m;
    Printf (" Enter value of N : \n");
    Scanf ("%d", &N);
    for (m = 1; m <= 10; m++)
      {
        [illegible]
        [illegible]
      }
        Printf ("%d x %d = %d \n", N, m, N*m);
```

Program - 3 :- Program to print Square & Cube of each value upto N

```
int main ()
{
    int N, square, cube, m;
    Printf ("Enter the value of N: \n");
    Scanf ("%d", &N);
    for (m = 1; m <= N; m++)
    {
        square = N * N    m * m
        Cube   = N * N * N    m * m * m
    Printf ("Square of %d is: %d \n", m, square \n");
    Printf ("Cube of %d is: %d \n", m, cube \n");
}
```

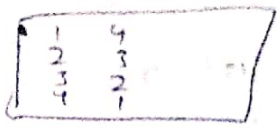
Program 4 :- Already done in Class Work

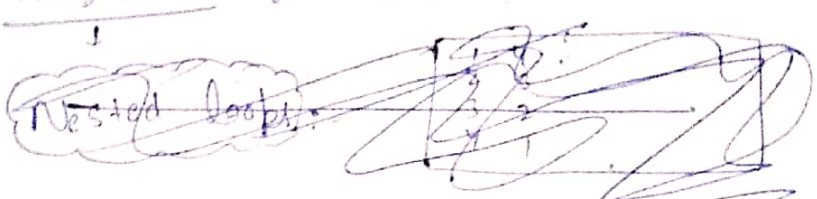
Program-5 :- Program to add all Even and odd values Separately from 1 to N.

```
int main ( )
{
    int N, m, Sum 1 = 0, Sum 2 = 0;
    Printf (" Enter the Value of N : \n");
    Scanf (" %d", & N);
    for ( m = 1 ; m <= N ; m++)
    {
        if ( m % 2 == 0)  // filter
        {
            Sum 1 = Sum 1 + m;
        }
        else
        {
            Sum 2 = Sum 2 + m ;
        }
    }
    Printf (" Sum of evens = %d \n", Sum 1);
    Printf (" Sum of odd = %d \n", Sum 2);
}
```

Imp Ques

Program-6 :- Program to show the following output





We will do this by Nested loops (loops with in loops)

⇒

```
int main ()
{
    int R, C;
    for (R=1 ; R<=3 ; R++)
    {
        for (C=1 ; C<=4 ; C++)
        {
            Printf
```

```
int main ()
{
    int i, j ;
    for (i = 1, j = 4 ; i <= 4 , j >= 1 ; i++ , j--)
    {
        Printf ("%d %d \n", i j) ;
    }
```

Multiple for loop

(Q4)

```
for (i = 1 ; i <= 4 ; i++)
    { Printf ("%d %d \n", i, 5-i) ;
    }
```

★ Nested loops :- Loop within Loop

```
int main()
{
    int R, C;
    for(R=1; R<=3; R++) // outer loop (Sem-3)
    {
        printf("\n");
    }
    {
        for(C=1; C<=4; C++) // inner loop (4 exam)
        {
            printf("%d_", c);
        }
        printf("\n"); // Result
    }
```

Output :- 1 2 3 4 ↓ 3 times

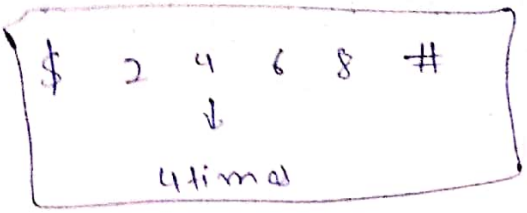
⇒ if we want to add Symbol on starting & ending like

```
$ 1 2 3 4 *
$ 1 2 3 4 *
$ 1 2 3 4 *
```

The print these b/w the two loops.

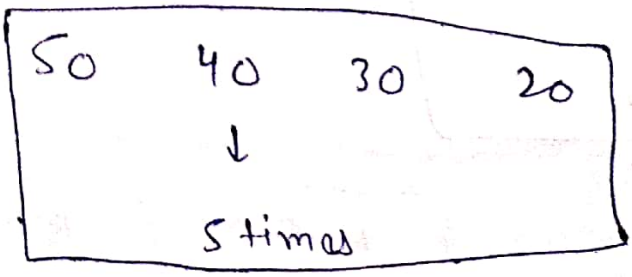
```
for(R=1; R<=3; R++)
    { printf("$");
        for(C=1; C<=4; C++)
            { printf("%d_", c);
            }
        printf("* \n");
    }
```

Ex :-




```
for (R=1; R<=4; R++)
  { Printf ("$_");
    for(C=2; C<=8 ; C=C+2)
      { Printf ("%d_", C);
      }
    Printf ("# \n");
  }
```

Ex :-

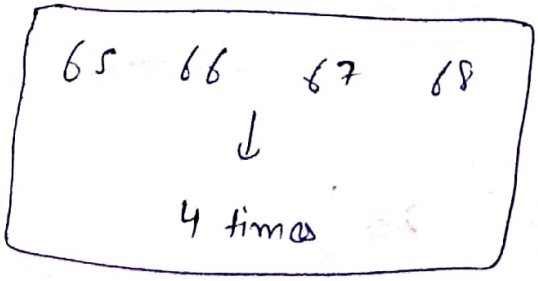



```
for (R=1 ; R<=5 ; R++)
  {
    for (C=50 ; C>=20 ; C=C-10)
      {
        Printf ("%d_", C)
      }
  }
```

⇒ Syntax :-

```
for (outer loop)
{
    ____
    for (inner loop)
    {
        ____
    }
}
____
}
```

Ex :-

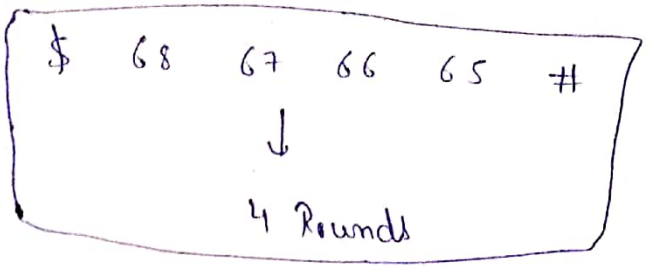


⇒

```
int main ()
{
    int R, C;
    for (R = 1; R <= 4; R++)
    {
        for (C = 65; C <= 68; C++)
        {
            Printf ("%d_", c)
        }
    }
```

Home Work

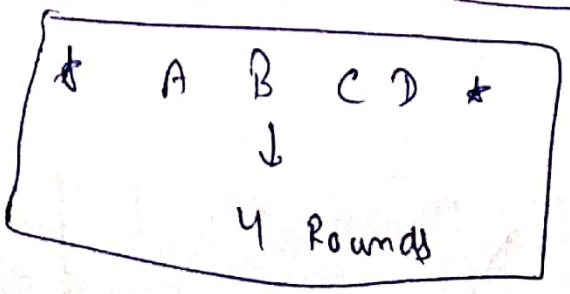
(1)



⇒

```
for (R=1; R<=4; R++)
  {
    Printf ("$ _ ");
    for (C=68; C>=65; C--)
      {
        Printf ("%d _ ", C);
      }
    Printf ("_ * \n");
  }
```

(2)



→ Do this by ASCII Values

'A' = 65
'B' = 66
'C' = 67
'D' = 68

⇒

```
for (R=1; R<=4; R++)
  {
    Printf ("* _ ");
    Printf ("A B C D _ ");
    Printf ("* \n");
  }
```

(3)

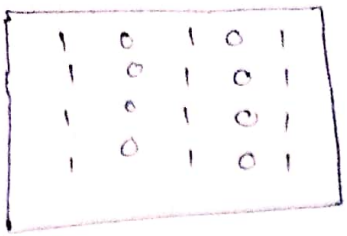
1	1	1	1	1
0	0	0	0	0
1	1	1	1	1
0	0	0	0	0

⇒
```
int main()
{
    int R, C=1;
    for (R=1; R<=2; R++)
    {
        Printf("%d - %d - %d - %d \n", C);
        Printf("%d - %d - %d - %d \n", C-1);
    }
```

⇒
```
int main()
{
    int R, C;
    for (R=1; R<=2; R++)
    {
        for (C=1; C<=5
```

⇒
```
int main()
{
    int R, C=1
    for (R=1; R<=2; R++)
    {
        Printf("%d - %d - %d - %d \n", C, C, C, C);
        Printf("%d - %d - %d - %d", C-1, C-1, C-1, C-1);
    }
```

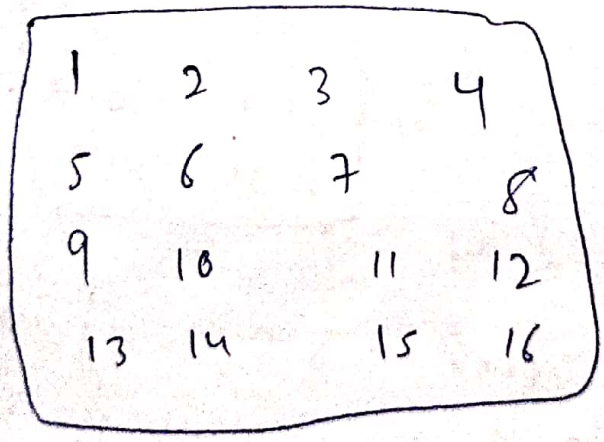
(4)




```
int main()
{
    int R, C=1;
    for (R=1; R<=5; R++)
    {
        Printf ("%d %d %d %d %d\n",
                C, C-1, C, C-1, C);
    }
```

---

(5)



⇒

```
int main()
{
    int R, C
    for (R=
```

(2)

```
* A B C D *
↓
4 Rows
```

→

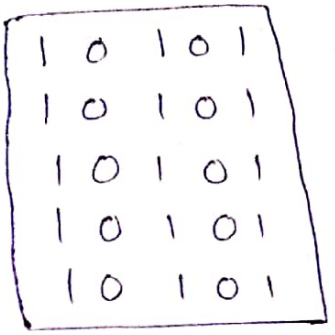
```
int main ()
{
    int R, C;
    for (R=1; R<=4; R++)
    { Printf ("* _");
        for (C=65; C<=68; C++)
        {
            Printf ("%c", C);
        }
        Printf ("* \n");
    }
```

(3)

```
| | | | |
0 0 0 0 0
| | | | |
0 0 0 0 0
| | | | |
```

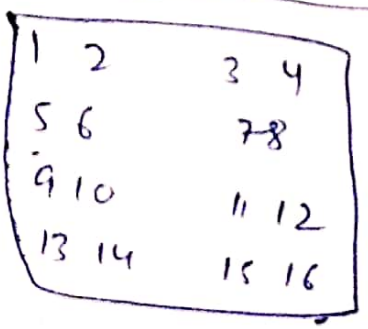
```
for (R=1; R<=5; R++)
{
    for (C=1; C<=5; C++)
    {
        Printf ("%d", R%2);
    }
    Printf ("\n");
}
```

(4)



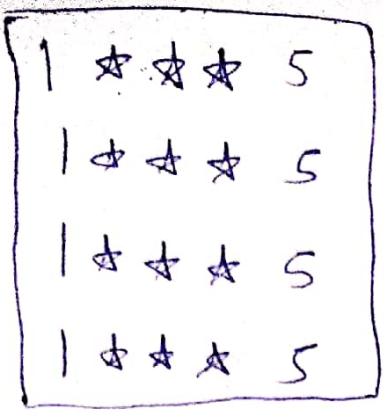

```
for (R = 1; R <= 5; R++)
  {
    for (C = 1 : C <= 5 ; C++)
      { Printf ("%d_", C%2);
      }
    Printf ("\n");
  }
```

(5)




```
                                int, k = 1;
for (R = 1 ; R <= 4 ; R++)
  {
    for (C = 1 ; C <= 4 ; C++)
      {
        printf ("%d", k);
          k++;
      }
    printf ("\n");
  }
```

(6)




```
for (R = 1; R <= 4 ; R++)
  {  for (C = 1; C <= 5; C++)
       {  if (C > 1 && C < 5)
            printf ("* _ ");
          else
            printf ("%d _ ", C);
       }
     printf ("\n");
  }
```

Make Important Questions :-

```
1
1 2
1 2 3
1 2 3 4
```

+1

```
for (R=1 ; R<=4 ; R++)
  {
   for (C=1; C<=R ; C++)
     {
       printf ("%d_", c);
     }
    printf ("\n");
  }
```

① outer loop

top

① ② ③

bottom

② inner loop

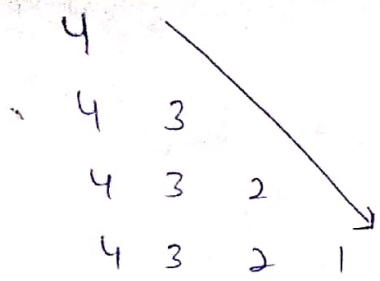
left to right L → R

for (① ; ③ ; ②)

```
1 2 3 4
1 2 3
1 2
1
```

```
for (R=4 ; R>=1 ; R--)
  {
    for (C=1 ; C<=R ; C++)
      {
        printf ("%d_", c);
      }
     printf ("\n");
  }
```

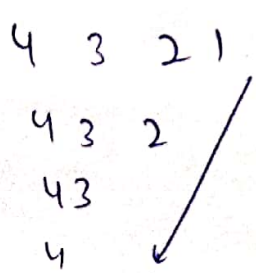
③




```
for (R=4 ; R>=1; R--)
  {  for (C=4 ; C>=R; C--)
       {  printf ("%d _", C);
       }
     printf ("\n");
  }
```

---

④




```
for (R=1; R<=4 ; R++)
{  for (C=4 ; C>=R; C--)
     {  printf ("%d_", C);
     }
   printf ("\n");
}
```

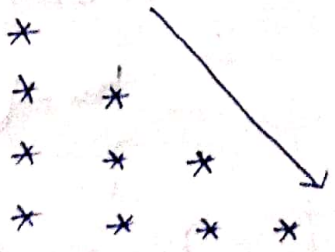
(5) Imp

```
1
2 1
3 2 1
4 3 2 1
```

```
for (R = 1 ; R <= 4 ; R++)
  {
    for (C = R ; C >= 1 ; C--)
      {
        printf ("%d_", C);
      }
    printf ("\n");
  }
```

---

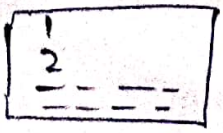
(6) Imp



```
for (R = 1; R <= 4 ; R++)
  {
    for (C = 1; C <= R ; C++)
      {
        printf ("*_");
      }
    printf ("\n");
  }
```

Break :- When the program control executes keyword "break" with in the body of loop then it leaves the current loop and goes to the statement below the body of loop (if any).

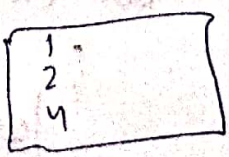
```
int main()
{
    int i;
    for (i=1; i<=4; i++)
    {
        if (i==3)
            break;
        printf("%d\n", i);
    }
    printf("\n=====");
}
```



Continue :- When the program control executes keyword "continue" within the body of loop (for() loop) then it leaves the statements unexecuted which are placed below it and control goes to updation point of loop (for())

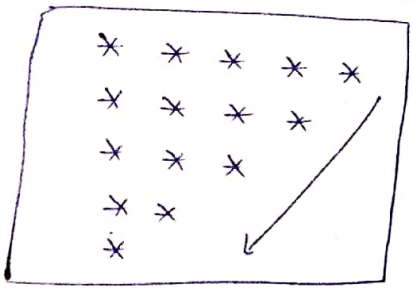
Eg :-

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



Home Work

(1)



```
for (R = 5 ; R >= 1 ; R--)
{
   for ( C = R ; C >= 1 ; C--)
      { printf ("* _ ");
      }
   printf ("\n");
}
```

(2)

```
1
2  3
4  5  6
7  8  9  10
```

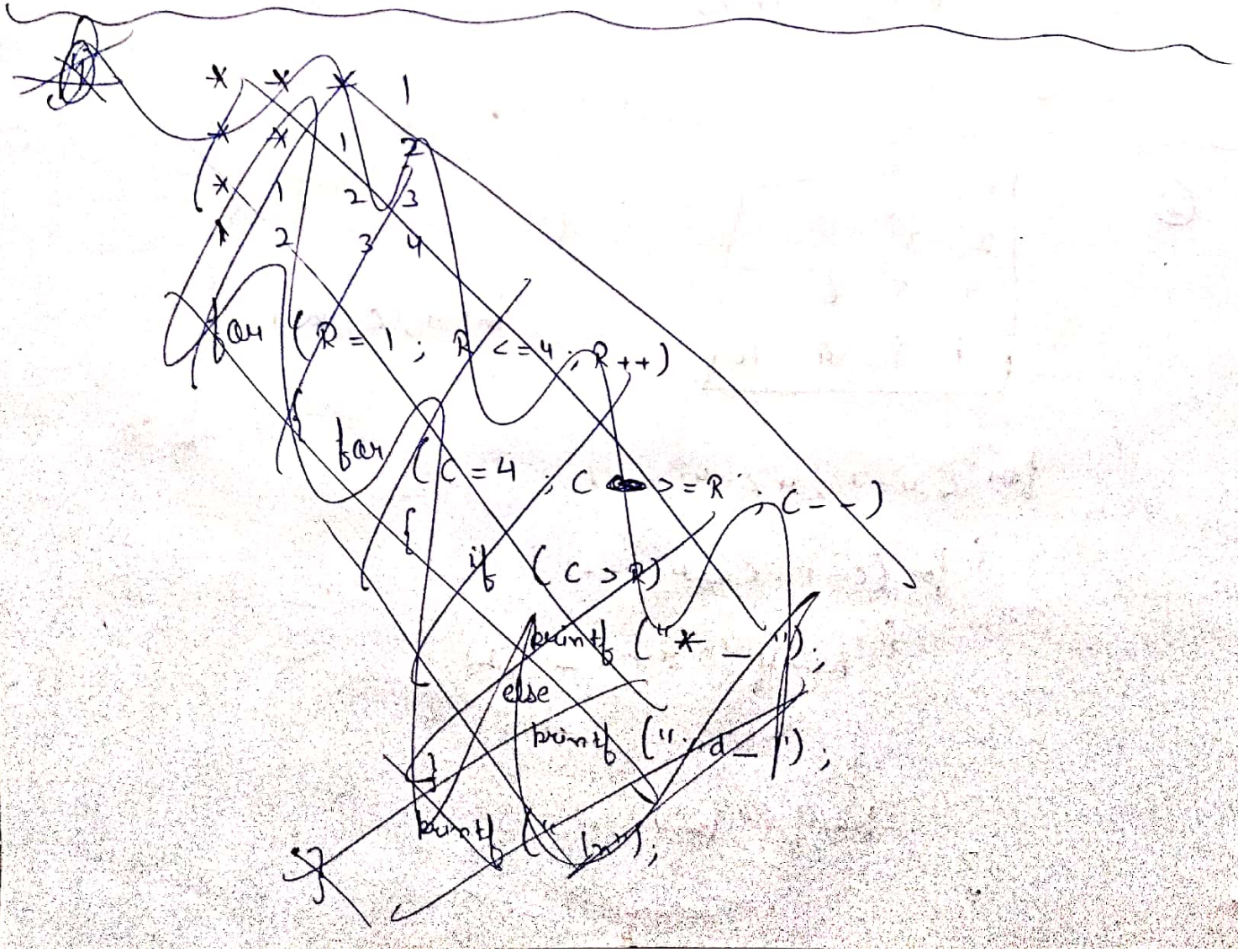
```
int R, C, k=1;

for (R=1 ; R <= 4; R++)
{ for (C=1 ; C<=R ; C++)
   { printf ("%d _ ", k)
     k++;
   }
  printf ("\n");
}
```

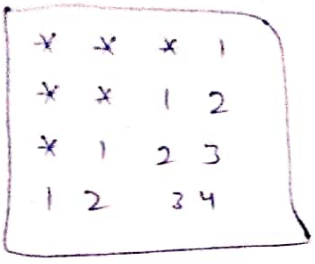
(3)

```
A B C D E
A B C D
A B C
A B
A
```

```
for (R = 69; R >= 65; R--)
  {
    for (C = 65; C <= R; C++)
      {
        printf ("%c ", C);
      }
    printf ("\n");
  }
```

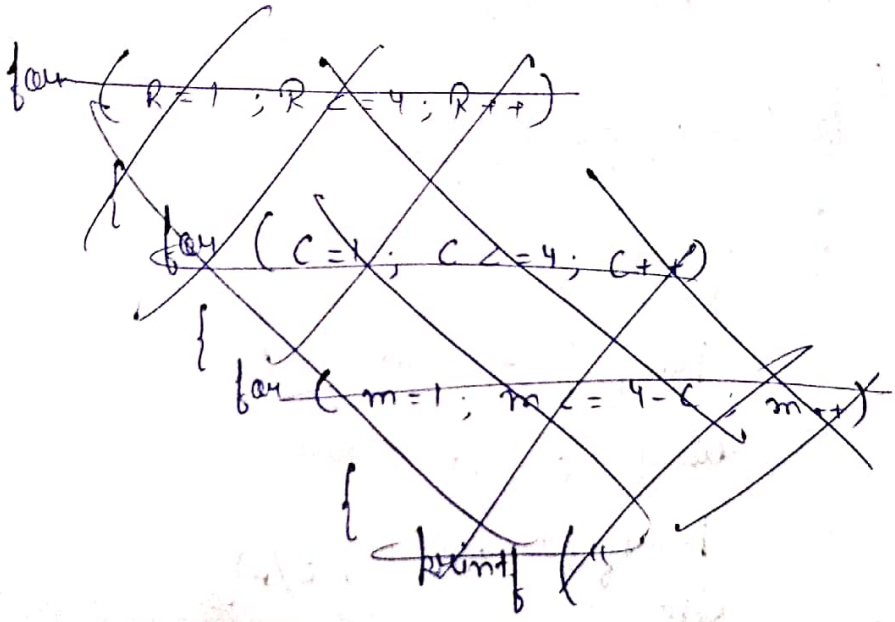


(4)




```
int main ()
{
    int r, c;
    for (r=1; r<=4; r++)
    { for (c=4; c>=1; c--)
      {
        if (c>r)
          printf ("* _ ");
        else if (c==r)
          printf ("%d_", c/r);
        else
          break;
      }
      if (r==2)
        printf ("%d_", r);
      else if (r==3)
        printf ("%d_%d_", r-1, r);
      else if (r==4)
        printf ("%d_%d_%d_", r-2, r-1, r);
      printf ("\n");
    }
```

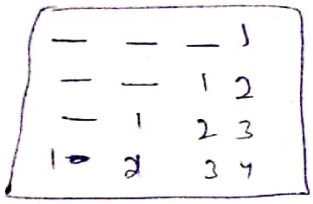
OR



By Sir :-

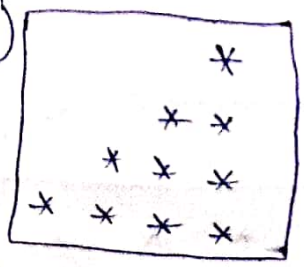
```
for (R=1 ; R<=4 ; R++)
{
    for (S=3 ; S>=R ; S--)
        { printf ("*_");
        }
    for (C=1 ; C<=R ; C++)
        { printf ("%d_", C);
        }
    printf ("\n");
}
```

(5)




```
for (R = 1; R <= 4; R++)
{
    for (S = 3; S >= R; S--)
    {
        printf ("  ");
    }
    for (C = 1; C <= R; C++)
    {
        printf ("%d ", C);
    }
    printf ("\n");
}
```

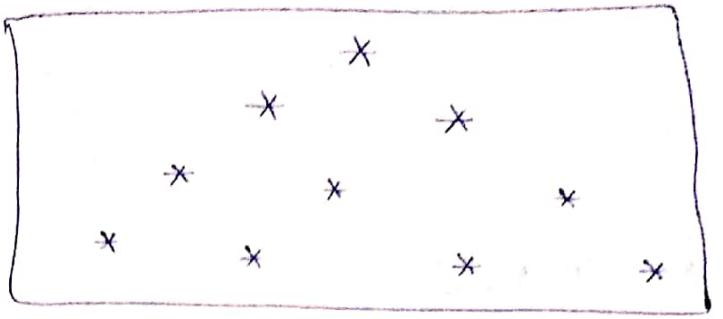
(6)




```
for (R = 1; R <= 4; R++)
{
    for (S = 3; S >= R; S--)
    {
        printf ("  ");
    }
    for (C = 1; C <= 3; C++)
    {
        printf ("* ");
    }
    printf ("\n");
}
```

Imp

(7)

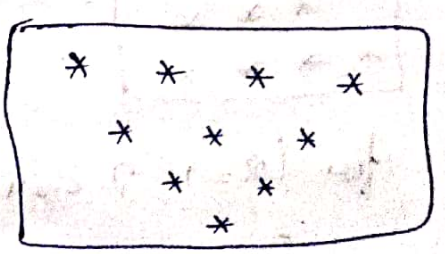


```
for (r=1 ; r<=4 ; r++)
  { for (s=3 ; s>=r ; s--)
    { printf ("__");
    }
   for (c=1 ; c<=r ; c++)
   { printf ("*___");
   }
  printf ("\n");
}
```

---

(8)

```
for (r=1 ; r<=4 ; r++)
  { for (s=1 ; s<=r ; s++)
    { printf ("__");
    }
   for (c=r ; c<=4 ; c++)
   { printf ("*___");
   }
  printf ("\n");
}
```



* while ( ) Loop :- Like for ( ) loop it is also used for executing a set of statements again & again by writting once.

Syntax :-

```
initialization
while (condition)
  {
    ======
    ======
    ======
    Updation
  }
```

// Program for factorial of N

```
=> int main
   {
     int N, m, factorial = 1;
     Printf (" Enter the value of N : \n");
     Scanf (" %d", &N);  //0
     m = 1;  // initialization
     while (m <= N)
       {
         factorial = factorial * m;
         m++;  // Updation
       }
     printf (" factorial = %d", factorial);
   }
```

Enter

11 Program for Reverse ; check Palindrome , Sum, Count of digits

⇒

```
int main ()
{ int N, rev = 0, r, photo ;
  int Sum = 0, Count = 0 ;
Printf (" Enter the value of N: \n");
Scanf (" %d", & N) ;
  Photo = N;
While (N! = 0)
  {
    r = N % 10 ;
    rev = rev x 10 + r ;
    N = N/10 ;
    Sum = Sum + r ;
    Count ++ ;
  }
printf (" \n Reverse = %d ", rev) ;
    if (rev == Photo)
        printf (" Palindrome") ;
    else
        printf (" Not palindrome") ;
printf (" Sum of all digits = %d \n", Sum) ;
printf (" No. of Counts = %d \n", Count) ;
}
```

// check Prime number

```
int main ()
{
    int N, d, Jasus = 1;
    Printf ("Enter value of N: \n");
    Scanf ("%d", &N);
    for (d = 2; d < N; d++)
    {
        if (N%d == 0)
        {
            Jasus = 0;
            break;
        }
    }
    if (Jasus == 1)
        printf ("Prime");
    else
        printf ("Not prime");
}
```

Home Work

Q- Program to print and Count Palindrome numbers 11 to n.

⇒

```
int main()
{
    int N, rev = 0, m, r, Prince, Count = 0;
    Printf ("Enter the value of n: \n");
    Scanf ("%d", &N);
    for (m = 11; m <= n; m++)
    {
        rev = 0;
        Prince = m;
        While (m != 0) ( Prince != 0)
        {
            r = m % 10;      r = Prince % 10;
            rev = rev * 10 + r;   rev = rev * 10 + r;
            m = m / 10;      Prince = Prince / 10;
        }
        if (rev == m)
        {
            printf ("Palindrome : %d", Prince);
        } Count ++;
    }
    printf("Count of Palindrome : %d \n");
}
```

Q- Write a program for following output

15 4
four five one

```
int main()
{
    int N, Rev = 0, Prince, r;
    Printf("Enter the value of N : \n");
    Scanf("%d", &N);   // 154
    Prince = N;   // 154
    while (N != 0)
    {
        r = N % 10;
        Rev = Rev * 10 + r;
        N = N/10;
    }
    switch(r)
    {
        case 0: printf("Zero");
        break;
        case 1:
        printf("one");
        break;
        case 2:
        printf("two");
        break;
        case 3:
        printf("three");
        break;
        case 4:
        printf("four");
        break;
        case 5:
        printf("five");
        break;
        case 6:
        printf("Six");
        break;
        case 7:
        printf("Seven");
        break;
        case 8:
        printf("Eight");
        break;
        case 9:
        printf("nine");
        break;
        case 10:
        printf("Ten");
        break;
    }
}
```

Q- Write a program To print table of n using while loop.

⇒
```
int main()
{
    int n, m;
    Printf ("Enter the value of n: \n");
    Scanf ("%d", &N);
    m = 1;
    while (m <= 10)
    {
        printf ("%d * %d = %d, n, m, n*
        m++;
    }
}
```

Q- Write a program for reverse & factorial of each digit of digit of Number.

⇒
```
int main()
{
    int N, rev = 0, r, factorial = 1, Prince; int m;
    Printf ("Enter the value of N: \n");
    Scanf ("%d", &N);
    Prince = N;
```

```
while (N != 0)
{
    r = N % 10;
    rev = rev * 10 + r;
    N = N/10;
```

```
    n = 1; factorial = 1;
    while (n <= r)
    {
        factorial = factorial * n;
        n++;
    }
    printf (" factorial of %d : %d \n", r, factorial);
}
printf ("Reverse : %d ", rev);
}
```

Q- Write a Program to Count no. of odd & even digits a number Contain.

```
int main ()
{ int N, r; int Count1 = 0, Count2 = 0;
    Printf (" Enter the value of N: \n");
    Scanf (" %d", &N);
while(N!=0){
    r = N % 10;
    N = N/10;
    if (r % 2 == 0)
        Count1 ++;
    else
        Count2 ++;
}
printf (" Count of even: %d", Count1);
printf (" Count of odd : %d \n", Count2);
```

11 Program to write Fibbonacci Series

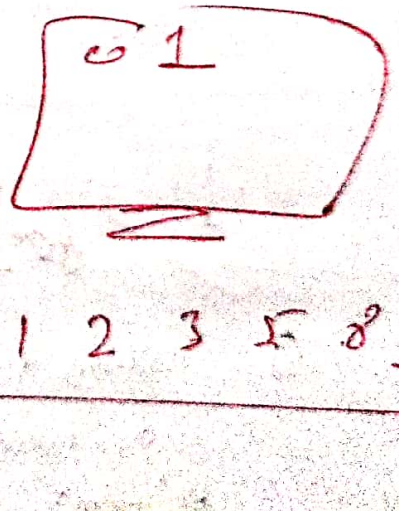
0 1 1 2 3 . . . N

a + b → c

```
int main ()
{
    int N, a = 0, b = 1, c = 0; x = 1;
    printf ("Enter the value of N : \n");
    Scanf ("%d", &N); //3.
    while (x <= N)
    {
        a = b;
        b = c;
        x++;
        printf ("%d ", c);
        c = a + b;
    }
}
```

A	B	C	N
0	1	0	
1	0	1	<= 3
0	1	1	

a	b	c
0	1	0
1	0	1
0	1	1



Do - while ( ) :-

Like for ( ) and while ( ) loops, It is also used for executing a set of statements again and again by writting once.

But unlike other loops, it firsts executes the [redacted] statements within loop body than checks the Condition. Also called exit Controlled loop.

* It always executes atleast once.

Syntax :-

```
initialization
do {
    ========
    ========
    ========
    Updation
} while (condition);
```

```
int main ()
{  int i=1;
   do{
      printf ("%d \n",i);
      i++;
   } while (i<=3);
}
```

1
2
3

Diff. b/w While & Do-while

while ( )

```
int main ()
{  int i = 4;
   while (i<=3)
      {
         printf ("%d \n", i);
         i++;
      }
}
```

No output

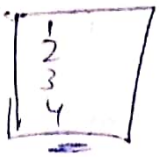
Do while ( )

```
int main ()
{
   int i=4;
   do {
      printf ("%d \n", i);
      i++;
   }
   while (i<=3)
}
```

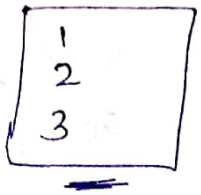
4

Some Technicals :-

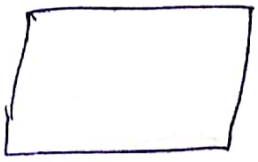
```
for (i=1 ; i<=3 ; i++)
    printf ("%d\n", i);
    printf ("%d", i); 114
```



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

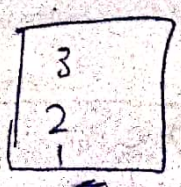


```
int i=3;
for ( ; ; )
{
    printf ("%d\n", i);
    i--;
    if (i==0)
        break;
}
```

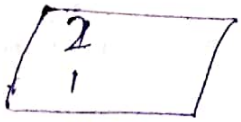


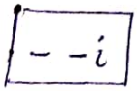
RAAZ KI BAAT :- `for ( ; if (i<=2); )`

```
int i=3;
for ( ; i ; )
    printf ("%d\n", i);
    i--;
}
```



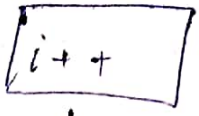
```
for (i=3; --i; )
    printf ("%d\n", i);
```





↑

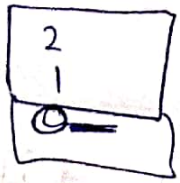
prefix decrement operator



↓

post fix decrement operator

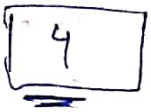
```
for (i=3; i--; )
{
    printf ("%d\n", i);
}
```



```
int main ()
{ int x = 2, y;
y = --x; printf ("y"); ①
y = x--;
printf ("x & y");
  ①   ②
```

- Gift :-

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



Home Work

Q- Write a program to print a table of n using Do while loop.

A-
```
int main()
{ int n, m;
  Printf("Enter the value of n: \n");
  Scanf("%d", &n);
  m = 1;
  do
  { printf("%d * %d = %d \n", n, m, n*m);
    m++;
  } while(m <= 10)
}
```

Q- Write a Program for palindrome number using do while

A -
```
int main()
{ int N, n, m, Prince, rev = 0;
  Printf("Enter the value of N: \n");
  Scanf("%d", &N);
  Prince = N;
  Do
  { n = N % 10;
    rev = rev * 10 + n;
    N = N / 10;
  } while(n != 0);
  printf("reverse = %d \n", rev);
  if (rev == prince)
  printf("Palindrome");
  else
  printf("Not");
}
```

Q - Write a program for armstrong number.

A -

```
int main()
{
    int n, m, sum = 0, r, prince;
    Printf ("Enter the value of n: \n");
    Scanf ("%d", &n);
    Prince = n;
    while (n != 0)
    {
        r = n % 10;
        n = n / 10;
        sum = sum + (r * r * r);
    }
    if (sum == prince)
        Printf ("yes it is Armstrong number \n");
    else
        printf ("it is not Armstrong number");
}
```