Nesting of loops:

```
loop {
    loop {
    }
}
```

Conditionals ?

Loops ?

when do something based on some condition

Conditi.

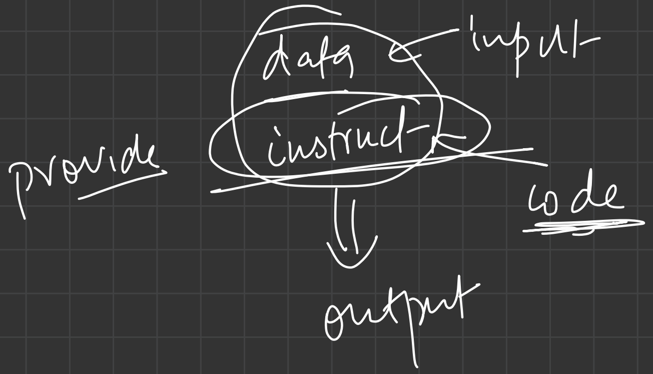
```
if ( condition ) {
    (
    )
}
else {

}
```

input an integer (age) {0 - 100}

{ tell whether the person is an adult ? }

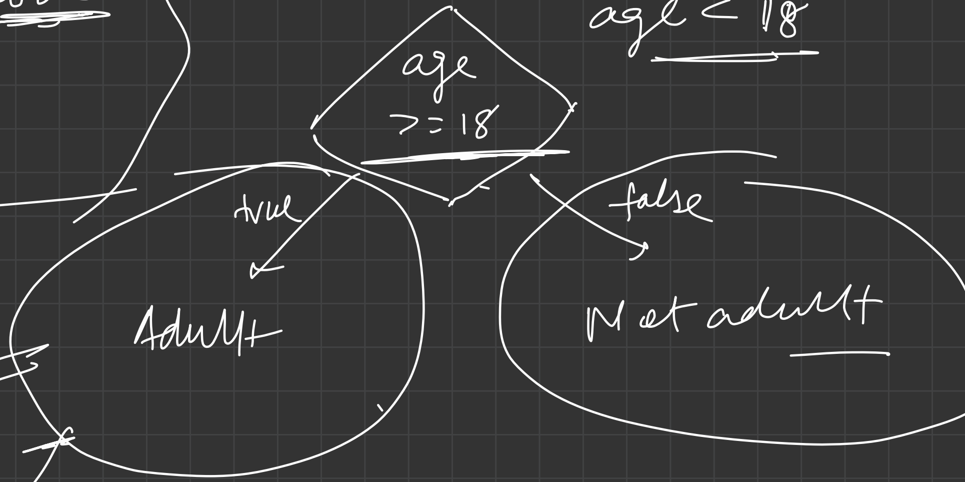
age >= 18 => adult



true

He/she is an adult

age < 18



```
if ( age >= 18 ) {

   cout << "Adult";

}

else??
```

```
if (          ) { ---- }

{ if( true ) { ---- } }
  else { ---- }
        false
```

```
if( age >= 18 ){

    cout<<"Adult";
}
else {
    cout<<"Child";
}
```

0 - 13	14-17	18 >=
child	teen	adult

Outn, 1>

code 2 min

```
if ( age <=13) {

    cout<<" Child";

}
        false                    true
if( age >=14 && age<=17){
X     cout<<"Teen";
}
          10 X=18 false
if ( age>=18) {
X     cout<<"Adult";
```

<=13 0-13

14>= && <=17 14-17

>= 18 18

```
      14
if( age <= 13 ){
false
    cout << "child";          13>
}   age >13
else if( age <= 17 ){         14, 15, 16, 17
    cout << "Teen";
}
else{         18, 19, 20, - - -
    cout << "Adult";
}
```

0, 1, 2, ..., 13

age > 13

age > 17

if, else if, else

if any of them becomes true all after it is ignored.

Take as input marks of five subjects. Calculate the percentage and grade according to the given conditions:

If percentage < 40% : Grade F
If percentage >= 40% : Grade E
If percentage >= 60% : Grade D
If percentage >= 70% : Grade C
If percentage >= 80% : Grade B
If percentage >= 90% : Grade A

Break till 9:30

9:30

if-else

loops → patterns

① Take input and then calculate percentage.

⇓

5 subjects

int 5 variables $\xRightarrow{\text{calculate}}$ percentage

② Determine the grade

< 40 : F ✓

>= 40 ; E ✓

>= 60 ; D

>= 70 ; C

>= 80 ; B

>= 90 ; A

Check whether grade is F

```
if(per < 40){
cout<<"grade F";
}
else { grade E, D, C, B, A
   per>=40 => sure
   if(per < 60) {
      cout<<"grade E";
   }
```

else { ~~per>=40~~ per>=60 : D, C, B , A.

```
        if( pw < 70) {
    cout << "grade D" ;
        }
        else      pw >= 70   :   C, B, A
  }
}
```

else if

0 - ~~40~~ 39

40 - 59

60 - 69

70 - 79

80 - 89

90 - 100

F f F E T

```
if( per < 40)
    F
else if( per < 60)
    E
else if (per< 70)
    D
else if (per <80)
    C
else if(per <90)
    B
else if( per < 100)
    A
```

{ 40 < per &&
per < 60 }

{ reverse
hona. }

Loops → for
→ while

repetition ⇑ Loops

initalization → starting cond^n

condition → repeat while this is true

end when this is false

updation

Print nos. from 1 to 100 → {to iterate over the loop}

```
1.    int i=1; // init
2.    while ( i<=100) { // condition
3.        cout<<i;
4.        i = i+1; // i++ , i+=1 / updation
5.    }
```

i=1, 2, 3, 4, - - , 100

i=1, 1, 1 - - - - -

```
for ( init ; condition ; updation ) {

        work;
                            repetitive
        body;               work

}
```

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

      cout << i;

}
```

8

```
* * * * * *
* * * * * *
* * * * * *
* * * * * *
* * * * * *
```

t = 1 - - - 50

1

✓

50

nesting ??

5 times repeat

10

```
cout << " * * * * * ";   cout << endl
```

1

× 5

100

'*'

"*"

char

sequence of char

rows

```
for(int i=1; i<=5; i++){
    cout << " * * * * * ";
    cout << endl;
}
```

10 *

no. of rows

2

```
loop {
    loop {
```

```
for( int i= 1; i<= ~~10~~ cols(3); i++ ) {
    cout << "x";
}
```

loop {
  x x
}
3

rows   cols

```
for (i=1; i<= rows; i++) {
   nesting for(j=1; j<= cols; j++) {
               cout << "x";
   }
   cout << endl;
}
```

2D-loop

3

1D linear

1-loop

2D

11	12	13
21	22	23
31	32	33

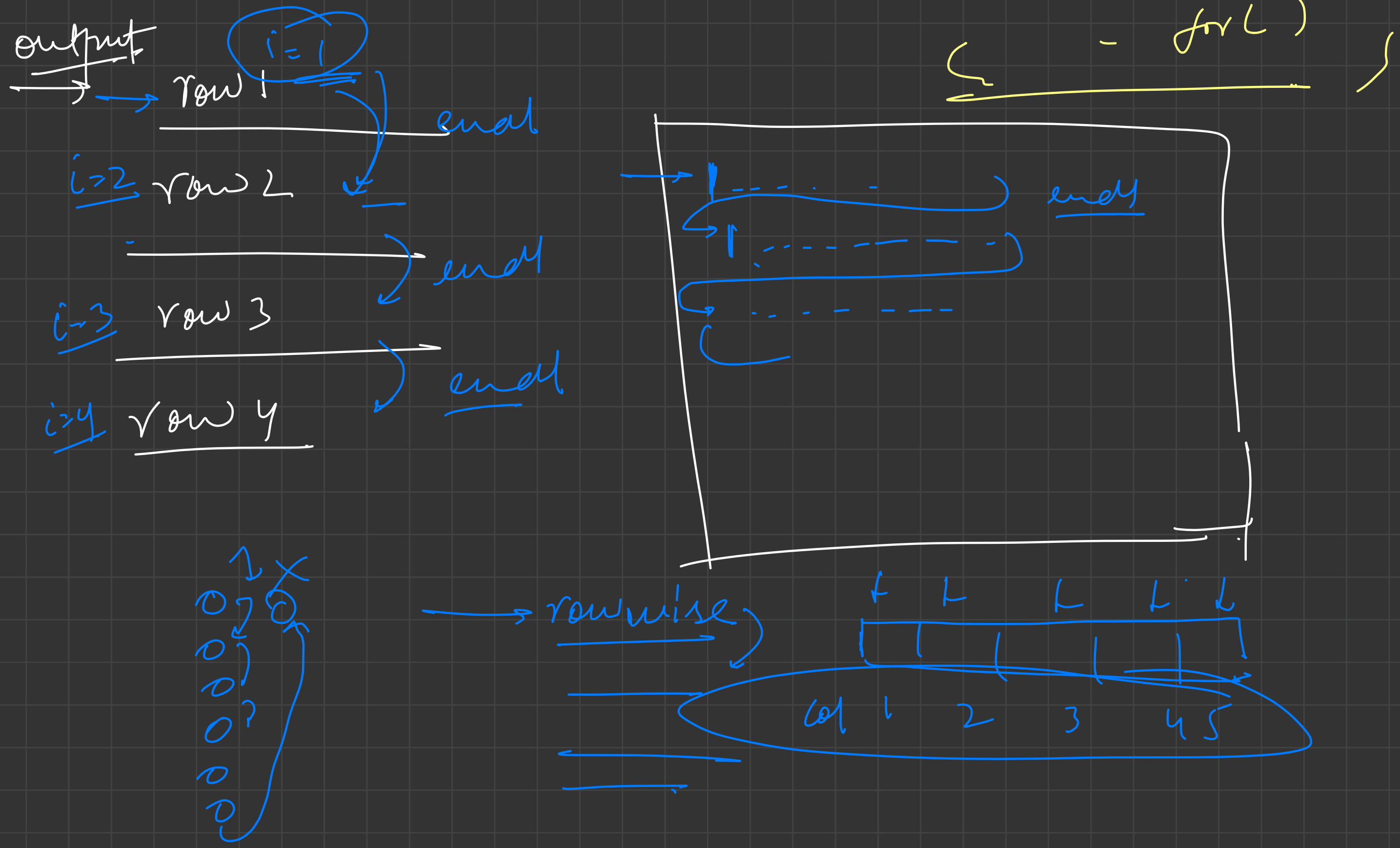
loop {
  loop {
  }
}

i=1 → endl
i=2 → endl

print → 1D → 2D ~~3D~~

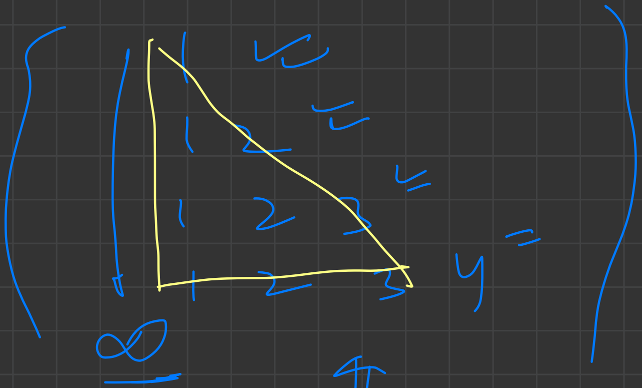
for (
   for (
)
)

Benefit


output
i=1
row 1
end
i=2 row 2
end
i=3 row 3
end
i=4 row 4
end
for ( )
endl
rowwise
col 1 2 3 4 5

n ⇒ no. of rows

cols pattern decide



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

⇑

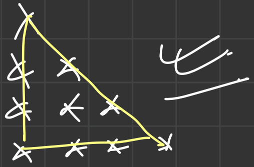
pattern ??

print this ??

code ??

logic

aptitude



same

Q.

```
      *
    * *
  * * *
* * * *
```

print this

input no. of rows

```
for( i=1 ; i<= row; i++){
    for( j=1 ; j<= i ; j++) {
        cout << "*";
    }
    cout << endl;
}
```

doubt??

→ 1

→ 1→2

A → 1→2→3

→ 1→2→3→4

every row starts with 1

as the column no. increases, value increases

col → 1 2 3 4

row
1 → 1
2 → 1 2
3 → 1 2 3
4 → 1 2 3 4

```
for( int i=1 ; i<=rows ; i++) {
    for( j=1 ; j<= i ; j++ {
        cout << j ;
    }
    cout<<endl;
```

D

1 2 3 4 5

1
1 3
1 3 5
1 3 5 7
1 3 5 7 9

⇒ (2* col no) - 1

⇒ { 2j - 1 }

k :-)

C

1	2	3	4	5
1				
2	3			
4	5	6		
7	8	9	10	
11	12	13	14	15

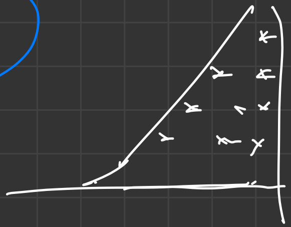
W

D

row \ col	1	2	3		
1	1				
2	0	1			
3	1	0	1		
	0	1	0	1	
	1	0	1	0	1

W.

0 , 1



```
if( i % 2 == 1 ) { odd          1 2 3 4 5 6 7 -
                                1 0 1 0 1 0 1 0 1 - - -
    if( j % 2 == 1 ) { odd
        cout<<"1 ";
    else {  cout << "0 ";      }
}
else { //even  col 1 2 3 4 5 6 7 8 9
               - 0 1 0 1 0 1 0 1 0 1

    if( j % 2 == 1 ) { %odd
        cout<<"0 ";
    }
    else {
        cout<<"1 ";
    }
}
```

i						spaces
1	-	-	-	-	*	4
2	-	-	-	*	*	3
3	-	-	*	*	*	2
4	-	*	*	*	*	1
5	*	*	*	*	*	0

i + spaces = n

output : left → right ← ✗

top ↓ down ↕ ✗

spaces = n - i

no. of rows = n

Q. how many spaces ? ✓

then how many * for each row ??

n - i => spaces

n - i => spaces

for each row

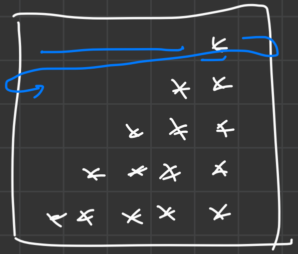
no. of cols = m

```
for( i=1 ; i<= n ; i++ ) {
    for( j=1 ; j <= n-i ; j++ ) {
        cout << " " ;
    }
    for( j=1 ; j<=i ; j++ ) {
```

spaces

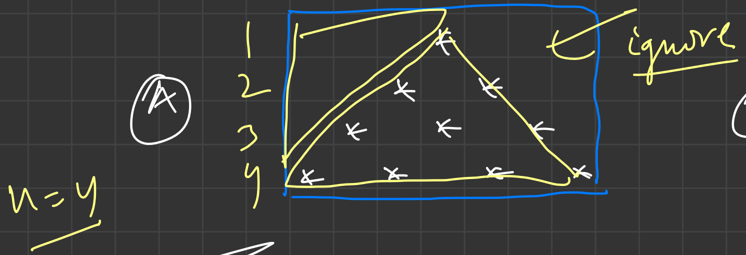
*

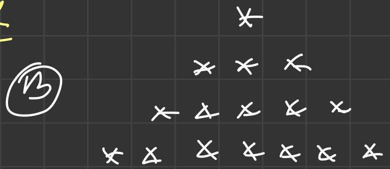
```
cout << " * ";
}
cout << endl;
}
```



i	space	*
1	4	1
2	3	2
3	2	3
4	1	4
5	0	5

no. of * = i



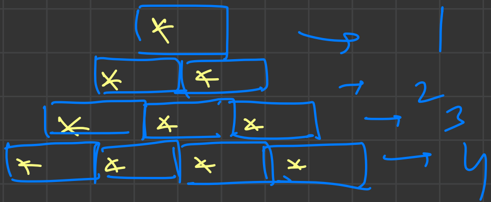


i	space
1	3
2	2

→ n-i

3 1

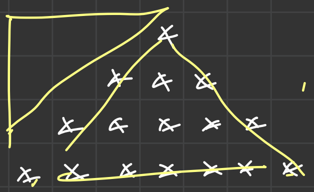
4 0



i

{* - 3

B



code??

B

on left before space

i	space	*
1	3	1
2	2	3
3	1	5
4	0	7
	$(n-i)$	$(2i-1)$

$2i-1$

1	2	3	4
1	3	5	7

$2(1)-1=1$

$2(2)-1=3$

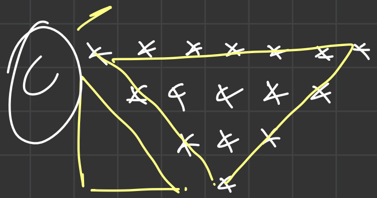
$2(3)-1=5$

$2(4)-1=7$

$2i-1$ → odd representation

$i=1$

$i=0 \Rightarrow 2i+1 \Rightarrow$ odd



i	Spaces	*
1	0	7 = 8 − 1 = 2n − 1
2	1	5 = 8 − 3 = 2n − 3
3	2	3 = 8 − 5 = 2n − 5
4	3	1 = 8 − 7 = 2n − 7

$(2 \times n) - 1$

$7 - 0$

$7 - 2$

$7 - 4$

$7 - 6$

$n = 4$

$2 \times n = 8$

formula $(i-1)$ $2(n-i)+1$ $2n-(2i-1)$

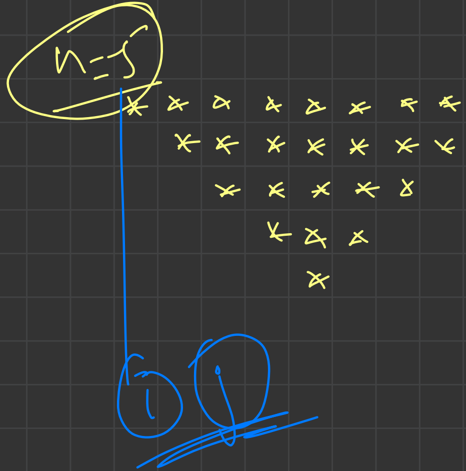
$2(n-i)-1 = 2n-i+1$??

✓

$2n-2i+1$

$2(n-i)+1$

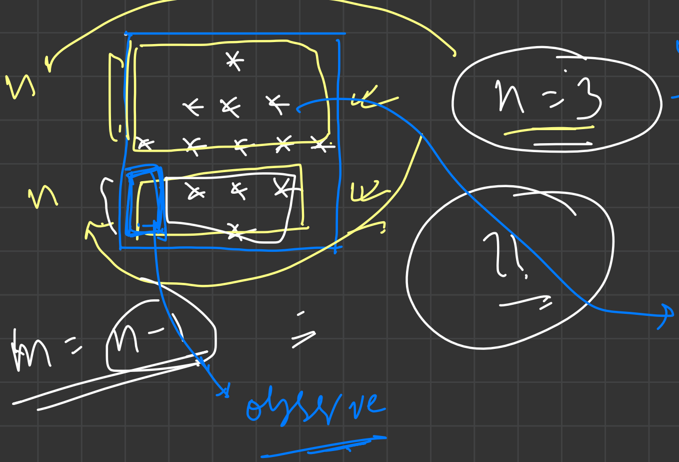
same



9	1	9 − 0
7	2	9 − 2
5	3	9 − 4
3	4	9 − 6
1	5	9 − 8

$2 \times n - 1$

$\{(2 \times n - 1) - (i-1) \times 2\}$



up

```
for(i=1 ; i<=n; i++){
   //space
   for(j=1; j<=n-i ;j++){
      cout<<'  ';
   }
   //*
   for( k=1 ; k<=2i-1 ; k++){
      cout<<'*';
   }
   cout<<endl;
}
```

lower part: // ~~dd~~

```
m=n-1
for( i=1 ; i<=m ; i++){

   for(j=1; j<=i-1; j++){
```

```
            cout<<' ';
       }
      for( k=1 ; k <= 2(m-i)+1 ; k++){
            cout<<'*';
       }
      cout<< endl;
  }
```

char	int
A →	65
B →	66
C →	67

char	int
a →	97
b →	98
c →	99

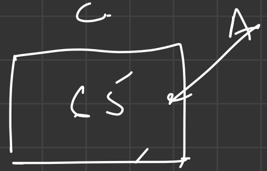
loops

conditionals

patterns

```
    A
  A B A
A B C B A
```

char → symbol



```
int x = 65;
cout << x;  ⇒ 65
cout << (char) 65 ⇒ A

char c = 'Z';
cout << c;  ⇒ Z
cout << (int) Z  ⇒ 90
```