DSA - DAY-7

* Revision Week-I

* How to Approach to solve Prog q.
* Algorithm
* Compiler
* Interpreter
* IDE
* Flowchart
* Components of Flowchart
* Pseudo Code

* Variable
* Data type. — Primitive — Float void. int
  Derived — Array
  User Defined — Class
* Variable Naming Convention
* SizeOF() — gives size of data.
* Storage of Data — (i) Positive No
  (ii) Negative No — 2's compl method
* Typecasting — (i) Implicit (ii) Explicit
* Reading an Number From memory
  (i) Positive
  (ii) Negative — 2's complement
* signed vs unsigned.

* Operators
  (i) Arithmetic :- (+, -, *, /, %)
  (ii) Relational :- (>, <, >=, <=, !=, ==)
  (iii) Logical :- (&& , || , !)
  (iv) Assignment :- (=)

* Conditional
  (i) IF-else
  (ii) if-else if

* Loops :- For loop (we want some task repeated then we use loops).

* Pattern (improving and logic) - rows & cols.

* Flowchart examples :-

(i) Add 2 numbers a & b
```
sum=a+b;
```

(ii) Find circumference of circle.
```
circum = 2*H*r;
```

(iii) Average of 3 Numbers.
```
avg = a+b+c/3;
```

(iv) Check No is odd or even
```
num % 2 == 0  — even
num % 2 != 0  — odd
```

(v) Grade Problem
```
if marks > 90 — A
if marks > 80 - B
if marks > 70 - C
if marks > 60 - P
if marks < 60 - F
```

(vi) Check No is +ve, -ve or 0

if (num > 0) — positive
if (num < 0) — (-ve)
else — 0.

(vii) Print counting from 1 to N

int i=1
if (i > N) — end
else print i & i+=1.

(viii) Add 'n' numbers from User.

int i=0;
int sum=0;
if (i > n) — print sum
else sum+=i; i=i+1;

(ix) Printing 1 to N but only odd no

int i=1;
if (i > N) - end
else print(i) i+=2;

(x) Find perimeter of triangle

per = a+b+c;

(xi) Find Simple Interest

SI = P*R*T/100;

(xii) Print counting From N to 1

int i=N;
if i<1 then end
else print i i-=1;

$0! = (n-1) * \ldots\ldots 1$.

q. Find Factorial of number.

```
int i=1  fact=1
if(i>N)  print fact
else     fact=fac*i;
         i++;
```

* Pattern Questions

n-row

①

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

row=0 col=0

(i) Space —— n-row-1 - print

(ii) First Half No — go for row+1 print col+1;

(iii) Second Half No — go for row print 2*row-col

②

```
— — — — *
— — — * *
— —   * * *
—     * * * *
* * * * *
```

row=0; row<n;

(i) Space :- (i) go for col < n-row-1
(ii) print " "

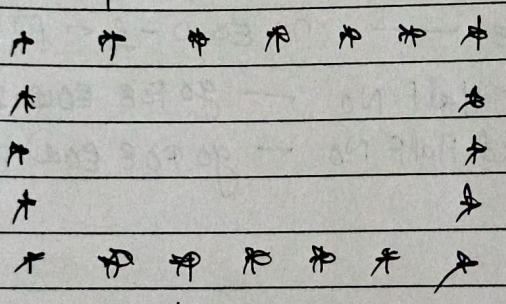
(ii) Star :- (i) go for col < row+1
(ii) print " * "

Q. 1 Numeric Half Pyramid

```
0   1
1   1 0
2   1 2 3
3   1 2 3 4 5
4   1 2 3 4 5 6
5   1 2 3 4 5 6
```

(i) ~~print~~ go for col < row+1
print col+1

Q. Hollow ~~Pyramid~~ Rectangle.

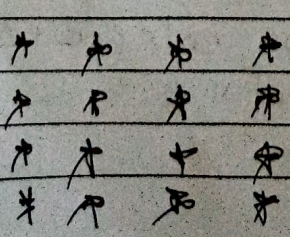


m & n input.

external for row < n

(i) print star
&
(ii) print space

(i) go for col < m
(ii) print if(row==0 || row==n-1 || col==0 || col==m-1) else print " "



Solid Square

external for row, rowcn.

(i) print star - go for row<n
print "*".