Date / /
Page No.

JAVA Basics

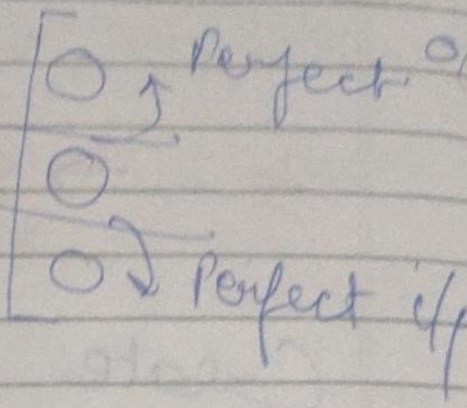
Problem Solving

↓

Sep^rn of Concern

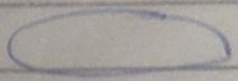
↙

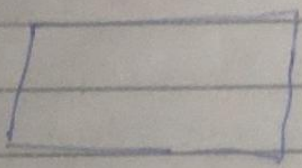
Break problem

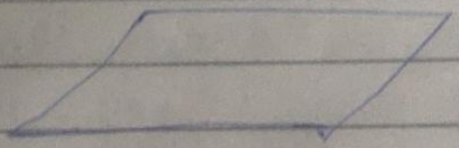


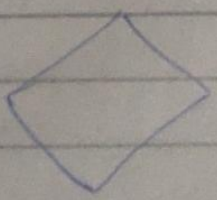
Flow Chart

4 shapes

 - Start/end

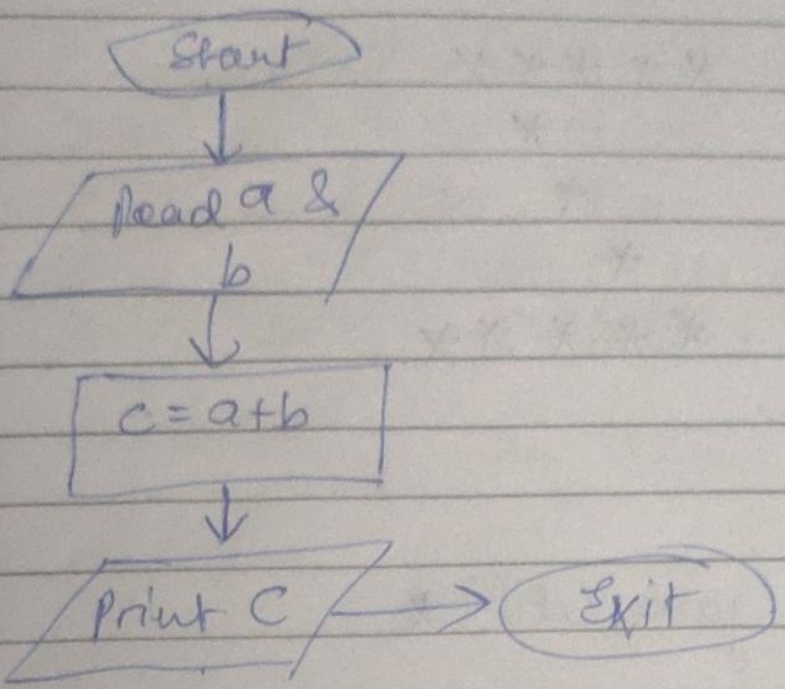
 - I/o Processing

 - ~~Processing~~ I/O

 - Decision

System.out => Console / Screen
System.in => Keyboard / i/p dev.

Q Add 2 nos.



Program

```
import java.util.*;
public class Main {

  public static void main (String args[]){

    int a=10, b=20;

    int c = a+b;

    System.out.println( c );

    }

}
```

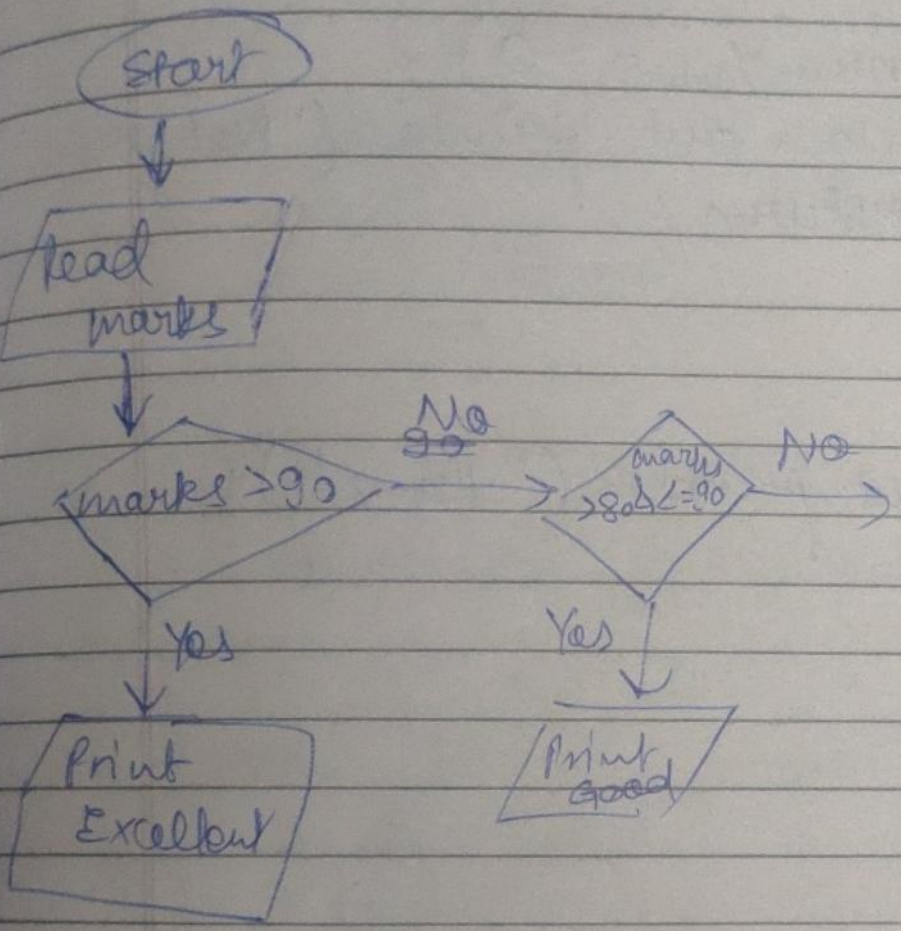
Q. Print pattern

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

```
import java.util.*
public class Main {
    public static void main(String args[]){
        System.out.println("*****");
        System.out.println("  *  ");
        System.out.println(" *   ");
        System.out.println("*    ");
        System.out.println("*****");
    }
}
```

2. Print Grade

>90	Excellent
>80 & <=90	Good
>70 & <=80	Fair
>60 & <=70	Meets Expectation
<=60	Below par.



Q. Pro Check if prime

→ Check for 2 to (num-1), if any divides it then not prime

```
for(int i=2; i<num; i++)
  {
     if ( num%i==0) {
        System.out.println("Not prime");
        return;
     }
  }

System.out.println("Prime");
}
```

2nd Approach

```
Scanner sen = new Scanner (System.in);
int n = sen.nextInt();
int lv = 2;

while (lv < n) {
   if (n % lv == 0) {
      System.out.println("not prime");
      return;
   }
   else
   {
      lv++;
   }
}

System.out.println("prime");
```

For 50, there is not factor b/w 25 & 50

i.e. $\frac{n}{2} + 1 \rightarrow n-1$

∴ we can change while condition to

~~not~~ while $\left(lv < \frac{n}{2}\right)$

& at last if $\left(lv == \frac{n}{2}\right)$

2nd Approach

for 36

Agar yahan nhi

1 x 36
2 x 18
3 x 12
4 x 9

$\sqrt{n}$ — 6 — x 6 — — — $\sqrt{n}$

to yahan bhi nhi

9 x 4
12 x 3
18 x 2
36 x 1

∴ coinde ($lv <= \sqrt{n}$) use this

This can be expressed as.

```
while (lv * lv <= n)
```

```
if (lv * lv > n) => prime
```

GCD — Euclidian

b > a

HCF(b, a) = HCF(a, b % a)

when a , a % b = 0

↑ This is HCF

I.g.

35, 20

```
      1
20 | 35
   | 20
   ------
      1
13 | 20
   | 13
   ------
      3
 5 | 15
   | 15
   ----
     X
```

HCF

b → 20 | 35 ← a

1

20

15 ↓ .

Inverse no.

:ften ?

→ how ?

~~int new_num = 0~~ ;

for (int i = 0 ; ~~i < n~~ n != 0 ; i++)

{

~~int~~ rem = n % 10 ; (int)

~~int~~ new_num = ~~rem~~ x (Math.pow(10, rem-1) ;

n = n / 10 ;

}

1	4	5	8 ← face val
4	3	2	1 ← place val

ofv ← orig. face val

orig place val

n=	n%10	op	ip	ifv	ifv x 10^(ip-1)
2578	8	1	8	1	1 x 10^(8-1)
257	7	2	7	2	2 x 10^(7-1)
25	5	3	5	3	3 x 10^(5-1)
2	2	4	2	4	1 x 10^(2-1)
0					

Add all to get the answer

HW

Prime factorization

take all facts from

fac = (2 - (n-1)) till n → 1

```
{
while (n % fac == 0)
{
n = n / fac;
print (fac);
}
fac ++;
}
```

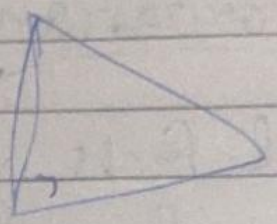
1440	2	fac = 2
720	2	
360	2	
180	2	
90	2	
45	3	45 % 2 ≠ 0
15	3	fac = 3
5	5	45 % 3 = 0
1 ✓		fac = 4
		5 % 4 ≠ 0
		fac = 5

Date: / /
Page No.

Pythagorean Triplet

3, 4, 5

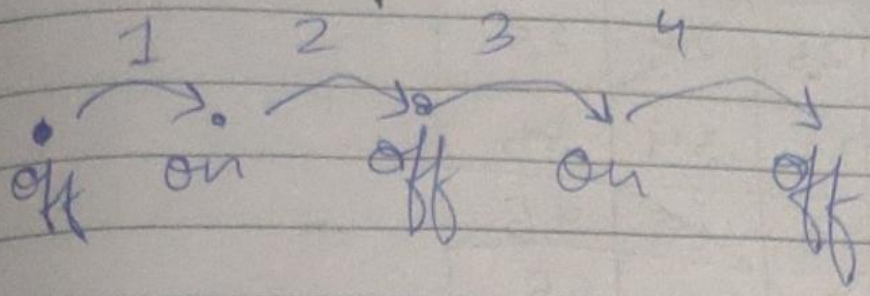
$a^2 + b^2 = c^2$?



1) Figure out ↑est of three
2) ↑es = c
3) a & b are other ones.

now, check $a^2 + b^2 = c^2$

Curious Case of Benjamin Bulbs



Even flucⁿ → OFF

1×18
2×9
3×6
6×3
9×2
18×1

→ Count no. of fluctuations

→ If fluc^n = odd ⇒ ON

" " = even ⇒ OFF

For. 20

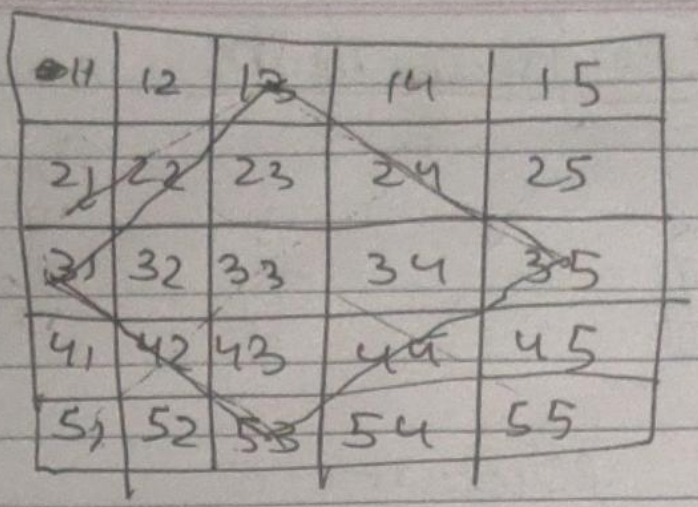
Observation : 1 4 9 16 . . .

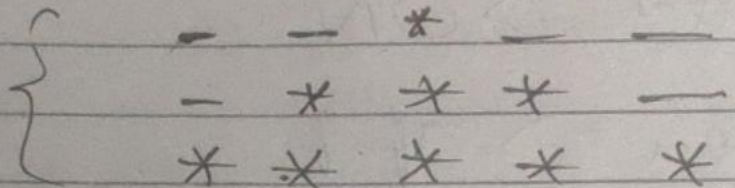
i.e. Find all the ~~prod~~ perfect squares less than or equal to "n".

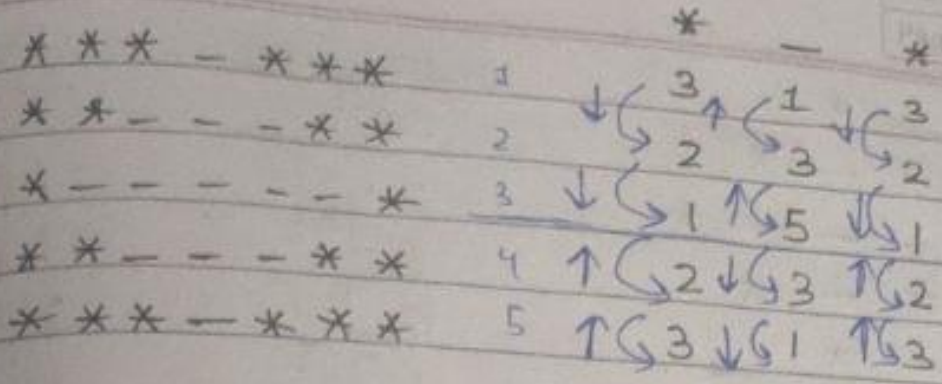
NOTE: Perfect sq. have only odd no. of factors.

Check if no. of factors is odd.

Date : / /
Page No.


11 12 13 14 15
21 22 23 24 25
31 32 33 34 35
41 42 43 44 45
51 52 53 54 55






```
int n = sc.nextInt();
int st = n/2 + 1; // initially stars for n(5) = n/2+1 i.e 3
int sp = 1; // initially spaces for n(5) = 1

for(int i=1; i<=n; i++) { // loop for no. of rows

    for(int j=1; j<=st; j++) { // print *'s
        System.out.print("*\t");
    }

    for(int j=1; j<=sp; j++) { // print _'s
        System.out.print("\t");
    }

    for(int j=1; j<=st; j++) { // print *'s
        System.out.print("*\t");
    }

    if(i<=n/2) { // for i<=n/2,
        st--;    // no. of star increases 2
        sp+=2;   // no. of spaces decrease
    }
    else {
        st++;    // opp. is true
        sp-=2;
    }
    System.out.println(); // Go to next line
}
```

```
_ _ *
_ * _ *
* _ _ _ *
_ * _ *
_ _ *
```

outer space (OS)	inner space (IS)
2] -1	-1] +2
1]	1]
0] -1	3] +2
1] +1	1] -2
2] +1	-1] -2

```
int n = sc.nextInt();

int os = n/2; // outer space initially n/2 i.e 2
int is = -1; // inner space initially -1 i.e no space

for(int i=1; i<=n; i++) { // loop for rows
  System.out
  for(int j=1; j<=os; j++){ // print outer spaces
    System.out.print("\t");
  }

  System.out.print("*\t"); // print 1 *

  for(int j=1; j<=is; j++){ // print inner space
    System.out.print("\t");
  }

  if(i>1 && i<n) { // print 1 * for all rows
    System.out.print("*\t"); // except 1st & last
  }

  System.out.println(); // Go to next line
```

```
if(i<=n/2){ // till n/2 i.e. 2 row
    os--; // outer spaces ↓ by 1
    is+=2; // inner spaces ↑ by 2
  }
else{
    os++; // opposite is true
    is-=2;
  }
}//end of for loop.
```

```
0
1   1
2   3   5
8   13  21  34
```

```
int n= scn.nextInt();
int a=0;
int b=1;
for(int i=1; i<=n; i++){
  for(int j=1; j<=i; j++){
      System.out.print(a + "\t");
      int c=a+b;
      a=b;
      b=1;
  }
  System.out.println();
}
}
```

$^nC_0 = 1$

$^nC_n = 1$

$^nC_k = {}^nC_{n-k}$

```
1
1   1
1   2   1
1   3   3   1
1   4   6   4   1
1   5   10  10  5   1
```

$^5C_0$ $^5C_1$ $^5C_2$ $^5C_3$ $^5C_4$ $^5C_5$

use formula

$$^nC_{k+1}^{(j+1)} = \frac{^nC_k \cdot (n-k)}{k+1}$$

initially put val = 1

$\rightarrow {}^5C_0 = 1$

$$^5C_1 = \frac{^5C_0 (5-0)}{0+1} = \frac{1(5)}{1} = 5$$

$$^5C_2 = \frac{^5C_1 (5-1)}{1+1} = \frac{5 \times 4}{2} = 10$$

start for $0 \rightarrow (n-1)$

$iCj = 1$ initially

```
for(int i=0; i<n; i++){ int iCj=1;
  for(int j=0; j<=i; j++){
    System.out.print(iCj+"\t");   // print iCj
    iCjp1 = iCj*(i-j)/(j+1);      // use formula for iCj+1
    iCj = iCjp1;                  // iCj = iCj+1 for next.
  }
  System.out.println();           // Go to next line
}
```

start from $0 \rightarrow (n-1)$ included

```
- - 1
- 2 3 2
3 4 5 4 3
- 2 3 2
- - 1
```

Steps to make the above pattern

a)

```
- - *
- * * *
* * * * *
- * * *
- - *
```

b)

```
- - 1
- 1 1 1
1 1 1 1 1
- 1 1 1
- - 1
```

c)

```
- - 1
- 2 2 2
3 3 3 3 3
- 4 4 4
- - 5
```

↓

d)

```
- - 1
- 2 2 2
3 3 3 3 3
- 4 4 4
- - 5
```

e)

```
- - 1
- 2 3 4
3 4 5 6 7
- 2 3 4
- - 1
```

f)

```
- - 1
- 2 3 2
3 4 5 4 3
- 2 3 2
- - 1
```

a) inside i & j loop.

```
if (i<=n/2) {
    sp--;
    st+=2;
} else {
    sp++;
    st-=2;
}
```

b) print 1

c) take val = 1 outside & ~~ini~~ after if else

```
val++;
```

~~d) take val = 1~~

F)
```
int n = sc.nextInt();
int sp = n/2;
int st = 1;
int val = 1; // take val for value at each row
for(int i=1; i<=n; i++){
  for(int j=1; j<=sp; j++){ //spaces
    System.out.print("\t");
  }

  int cval = val; // change value column-wise
  for(int j=1; j<=st; j++) { // values (stars)
    System.out.print(cval+"\t");

    if(j<=st/2) { // value ↑es till mid-column
      cval++;
    } else { // value ↓es after mid column
      cval--;
    }
  }

  if(i<=n/2){ // spaces, values & column
    sp--;     // values changes after mid row
    st+=2;
    val++;
  } else {
    sp++;
    st-=2;
    val--;
  }
  System.out.println(); //Go to next line
}
```

n rows

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

2n-3 spaces

2n-1 columns

Steps

```
*  -  -  -  -  -  *
*  *  -  -  -  *  *
*  *  *  -  *  *  *
*  *  *  *  *  *  *
```

1)

```
int n = sc.nextInt();
int sp = 2*n - 3
int st = 1;
for(int i=1; i<=n; i++) {
   for(int j=1; j<=st; j++){
      System.out.print("*\t");
   }

for(int j=1; j<=sp; j++) {
   System.out.print("\t");
}

if(i==n){
   st--
}
for(int j=1; j<=st; j++){
   System.out.println("*\t");
}
st++;
sp-=2;
System.out.println();
}
```

2) Change code s.t. 1 is printed instead of * like

```
*                 *          1                 1
* *               * *        1 1               1 1
* * *   =>        * * *      1 1 1     =>      1 1 1
* * * *           * * * *    1 1 1 1           1 1 1 1
```

3) Put int val=1; inside for(i...) loop as every row starts with 1;

```
1                 2
1 2               3 4
1 2 3             4 5 6
1 2 3 4           5 6 7
```

4) Put val++ inside 1'st & 3rd for(j...) loops like.

1st for loop

```
for(int j.....){
    Sysout(....);
    val++;
}
```

3rd for loop

```
for(int j.....){
    val++;
    Sysout(...);
}
```

5) Put val-- inside if(i==n){ ; } to handle last row.

```
int n = scn.nextInt();
int sp = 2*n - 3;
int st = 1;

for(int i=1; i<=n; i++) {
    int val = 1;
    for(int j=1; j<=st; j++){
        System.out.print(val + " ");
    }

    for(int j=1; j<=sp; j++) {
        System.out.print(" ");
    }

    if(i==n) {
        st--;
        val--;
    }

    for(int j=1; j<=st; j++) {
        val--;
        System.out.print(val + " ");
    }

    st++;
    sp-=2;
    System.out.println();
}
```

Pattern - 5 PepCoding

```
- - *
- * * *
* * * * *
- * * *
- - *
```

sp, st

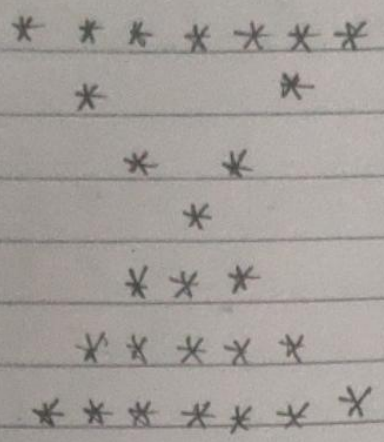
2, 1 → 1 st↑ sp↓
1, 3 → 2 st↑ sp↓
0, 5 → 3
1, 3 → 4 st↓ sp↑
2, 1 → 5 st↓ sp↑

```
sp = n/2; st = 1;
for(int i=1; i<=n; i++) {
  // Sysout(sp + "," + st); prints -'s & *'s
  for(int j=1; j<=sp; j++){
    System.out.print("\t");
  }
  for(int j=1; j<=st; j++){
    System.out.print("*\t");
  }

  if(i<=n/2){
    st++; st+=2;
    sp--;
  }
  else{
    st-=2;
    sp++;
  }
}
```

Pattern 17

```
- - *
- - * *
* * * * *
- - * *
- - *
```

```
int sp = n/2;
int st = 1;

for (int i=1; i<=n; i++) {

  for (int j=1; j<= sp; j++) {

     if (i==n/2+1) {
        System.out.print("*\t");
      }
     else {
          System.out.print("\t");
      }
  }
  for (int j=1; j<=st; j++) {
     System.out.print("*\t");
   }
  if (i<= n/2) st++;
  else st--;

  System.out.println();
}
```



Hint

⇒ Print this

```
* * * * * * *
- * * * * *
- - * * *
- - - *
- - * * *
- * * * * *
* * * * * * *
```

sp	st
0	7
1	5
2	3
3	1
2	3
1	5
0	7

⇒ Now, remove the blue stars

if(i>1 && i<=n/2 && j>1 && j<st)

Empty this area.

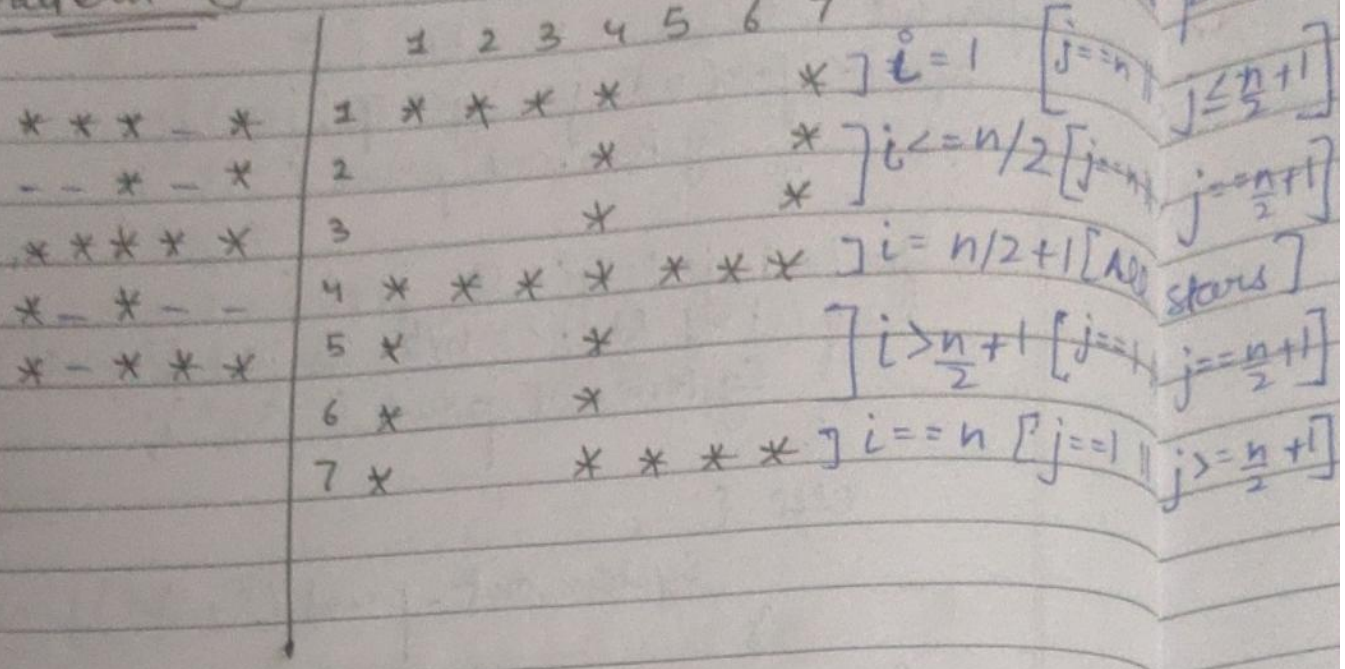
Code step 1

```
st=n; sp=0;
for( i→ n){
   for( j= 1 → sp) {
         print ("\t");
   }
   for( j=1 → st) {
         print ("*\t");
   }
if(i<=n/2) {sp++; st--=2;}  print(next line);
else {sp--; st+=2;}
                              }
```

```
int st = n*2, sp=0;
for(int i=0; i<=n; i++) {
  for(int j=1; j<=sp; j++) {
        System.out.print("\t");
    }
  for(int j=1; j<=st; j++) {
    if( i>=1 && i<=n/2 && j>1 && j<st) {
      System.out.print("\t");
    }
    else {
        System.out.print("*\t");
    }
  }
  if (i<=n/2) {
    sp++;
    st-=2;
  }
  else {
    sp--;
    st+=2;
  }
  System.out.println();
}
```

Date:
Page No.

Pattern - 19



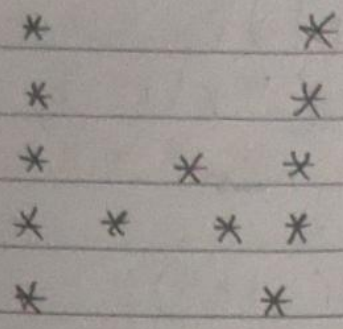
(Star pattern)

```
int n=scn.nextInt();

for(int i=1; i<=n; i++) {

  for(int j=1; j<=n; j++) {

    if(i==1) {
      if(j==n || j<=n/2+1){
        System.out.print("*\t");
      }
      else {
        System.out.print("\t");
      }
    }else if(i<=n/2){
      if(j==n/2+1 || j==n){
        System.out.print("*\t")
      }
      else {
        System.out.print("\t");
      }
    }else if(i==n/2+1) { System.out.print("*\t");
```

```
else if (i<n) {

    if ( j==1 || j== n/2 +1){
        System.out.print (" * \t");
      }
    else{
        System.out.print ("\t");
      }
}else {
    if ( j==1 || j>= n/2 +1) {
        System.out.print (" * \t");
      }
    else {
        System.out.print ("\t");
      }
  }
  System.out.println();
}
```

Date :
Page No.

Pattern -20



11	12	13	14	15
21	22	23	24	25
31	32	33	34	35
41	42	43	44	45
51	52	53	54	55

```
int n=scn.nextInt();
for(int i=1; i<=n; i++){
  for(int j=1; j<=n; j++){
    System.out.print("*\t");
    if(j==1 || j==n){
      System.out.print("*\t");
    }
    else if(i>n/2 && (i==j || i+j==n+1)){
      System.out.print("*\t");
    }
    else{
      System.out.print("\t");
    }
  }
  System.out.println();
}
```