

Loops in Java → Used to perform task again and again

print star 20 times

*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*

→ System.out.println("*");
→ System.out.println("*");

→ System.out.println("*");
→ System.out.println("*");
→ System.out.println("*");
→ System.out.println("*");
→ System.out.println("*");
→ System.out.println("*");
→ System.out.println("*");
→ System.out.println("*");
→ System.out.println("*");
→ System.out.println("*");
→ System.out.println("*");
→ System.out.println("*");
→ System.out.println("*");
→ System.out.println("*");
→ System.out.println("*");
→ System.out.println("*");
→ System.out.println("*");
→ System.out.println("*");
→ System.out.println("*");



- ↳ ① for loop
- ↳ ② while loop
- ↳ ③ do while loop } - H.W.

int i = 1;

for (_____ ; i <= 20;) {

i++;

5

```
int i=1;  
for(;i<=20;){  
    System.out.println("*");  
    i++;  
}
```

```
int i=1;  
for ( j ; i<=20; )  
{  
    sys("/*");  
}
```

→ infinite time

$i = 1 \leq 20(1) \rightarrow *$
 $i = 1 \leq 20(2) \rightarrow *$
 $i = 1 \leq 20(3) \rightarrow *$
 $i = 1 \leq 20(4) \rightarrow *$

while loop

syntax →

initialization

while (condition) {

increment/decrement;

}

Q - Print star 5 times

int i = 1;

while (i <= 5) {

 System.out.println ("*");
 i++;

3

i = 1 <= 5 (T)

i = 2 <= 5 (T)

i = 3 <= 5 (T)

i = 4 <= 5 (T)

i = 5 <= 5 (T)

i = 6 <= 5 (F)

*

*

*

*

*

int i = 5;

while (i >= 1) {

 System.out.println ("*");
 -- i;

2

Check prime

$n = 13 \rightarrow$ prime , not prime

↳ Prime no. \rightarrow exactly 2 factors $\rightarrow 1, n$

↳ Calculate the factors of a given no.

$n = 2^4 \rightarrow \min = 1$) \rightarrow max = 24
 $\text{int count} = 0;$
 $\text{for (int } i=1; i < n; i++) \{$
 $\quad \text{if } (n \% i == 0)$
 $\quad \quad \text{count} += 1;$

}

$\text{if } (\text{count} == 2)$
 $\quad \text{sysoc("prime")}$
 else
 $\quad \text{sysoc("not prime")};$

$n =$
 $\text{for (int } i=1; i < n; i++)$

$\boxed{n=8}$
 $\{2, 7\}$

$\text{for (int } i=2; i < n; i++)$
 $\quad \text{if } (n \% i == 0) \{$
 $\quad \quad \text{sysoc("prime");}$
 $\quad \quad \text{return;}$

$\text{for (int } i=2; i < n; i++) \rightarrow \boxed{1, n}$

$\quad \text{if } (n \% i == 0)$
 $\quad \quad \text{sysoc("not prime");}$

$n = 8$, count = 0

$i = 1 \leftarrow 0 \text{ (T)}$	$8 \% 1 = 0 \text{ (T)}$	count = 1
$i = 2 \leftarrow 1 \text{ (T)}$	$8 \% 2 = 0 \text{ (T)}$	count = 2
$i = 3 \leftarrow 2 \text{ (F)}$	$8 \% 3 = 0 \text{ (F)}$	count = 2
$i = 4 \leftarrow 3 \text{ (T)}$	$8 \% 4 = 0 \text{ (T)}$	count = 3
$i = 5 \leftarrow 4 \text{ (T)}$	$8 \% 5 = 0 \text{ (F)}$	count = 3
$i = 6 \leftarrow 5 \text{ (F)}$	$8 \% 6 = 0 \text{ (F)}$	count = 3
$i = 7 \leftarrow 6 \text{ (F)}$	$8 \% 7 = 0 \text{ (F)}$	count = 3
$i = 8 \leftarrow 7 \text{ (T)}$	$8 \% 8 = 0 \text{ (T)}$	count = 4
$i = 9 \leftarrow 8 \text{ (F)}$		
count = 2	$\Rightarrow 4 \neq 2 \times$	

Toggle the character

$$\underline{ch = 'a'} \leftrightarrow 'A'$$

$$ch = 'A' \rightarrow 'a'$$

$$ch = 'a' \rightarrow 'A'$$

$$ch = 'w' \rightarrow 'W'$$

$$\frac{'c'}{\text{character}}$$

ASCII value -

↳ if ($ch \geq 'a'$ & & $ch \leq 'z'$) {

$$ch = (\underline{\text{char}})(ch - 'a' + 'A');$$

} else {

$$ch = (\underline{\text{char}})(\underline{ch - 'A'} + \underline{'a'});$$

↳ integers

↳

$$\underline{ch = 'c'}$$

→ ASCII value -

$$ch - 'a' \rightarrow \begin{array}{l} \overbrace{\text{---}}^1 \\ \overbrace{c - a}^{\downarrow} + \overbrace{A}^{\downarrow} = 'C' \end{array}$$

$$99 - 97 = \underline{2} + \underline{65} = \underline{67}$$

$$ch = 'C'$$

$$\frac{'C' - 'A' + 'a'}{67 - 65 + 97} = \underline{99} \rightarrow 'C'$$

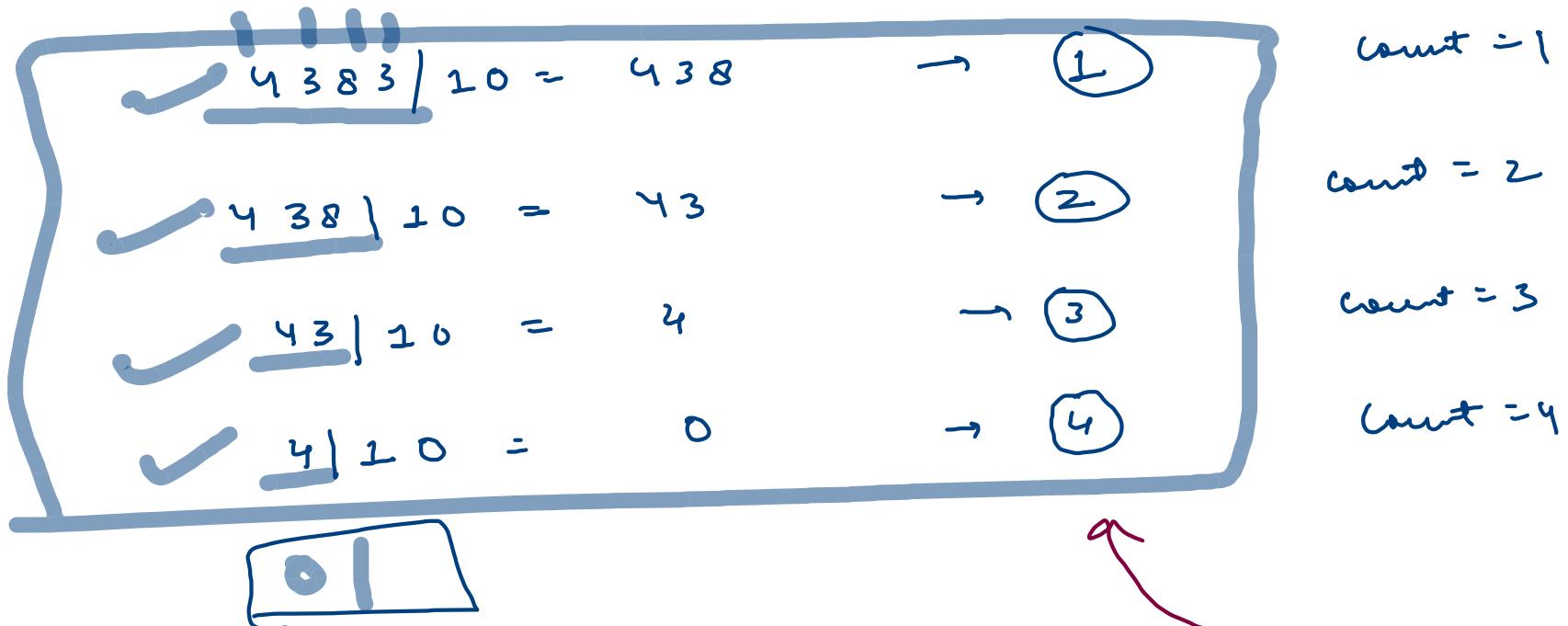
Practice Session

↳ 5 mins

$n = \underline{4} \underline{3} \underline{8} \underline{3}$ →

$\overbrace{\underline{4} \underline{3} \underline{8} \underline{3}}$ → $\underline{4}$

$\overbrace{\underline{5} \underline{3} \underline{6} \underline{5} \underline{4} \underline{3}}$ → ⑥



$(n > 0) \quad n \leftarrow 10 ; \text{count}++;$

$- 4383 \times -1 = \underline{\underline{4383}}$

$\underline{- 4383} \rightarrow \boxed{4}$

$$\overline{839593} \Big|_{10} = 53654 \rightarrow \text{count} = 1$$

$$53654 \Big|_{10} = 5365 \rightarrow \text{count} = 2$$

$$5365 \Big|_{10} = 536 \rightarrow \text{count} = 3$$

$$536 \Big|_{10} = 53 \rightarrow \text{count} = 4$$

$$53 \Big|_{10} = 5 \rightarrow \text{count} = 5$$

$$5 \Big|_{10} = 0 \rightarrow \text{count} = 6$$

0 $\Big|_{10}$ X

count = 6

C

~~70-230~~

520

150

230

\$ f \{ \phi \{ \} \circ \} \}

⑥

$$\rightarrow 230 / 10 = 2\overline{3} \rightarrow \text{count} = 1$$

$$\rightarrow 23 / 10 = 2 \rightarrow \text{count} = 2$$

$$\rightarrow 2 / 10 = 0 \rightarrow \text{count} = 3$$

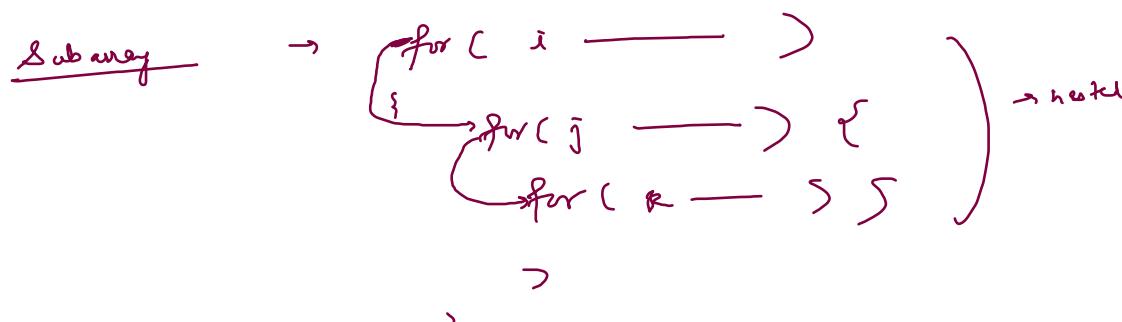
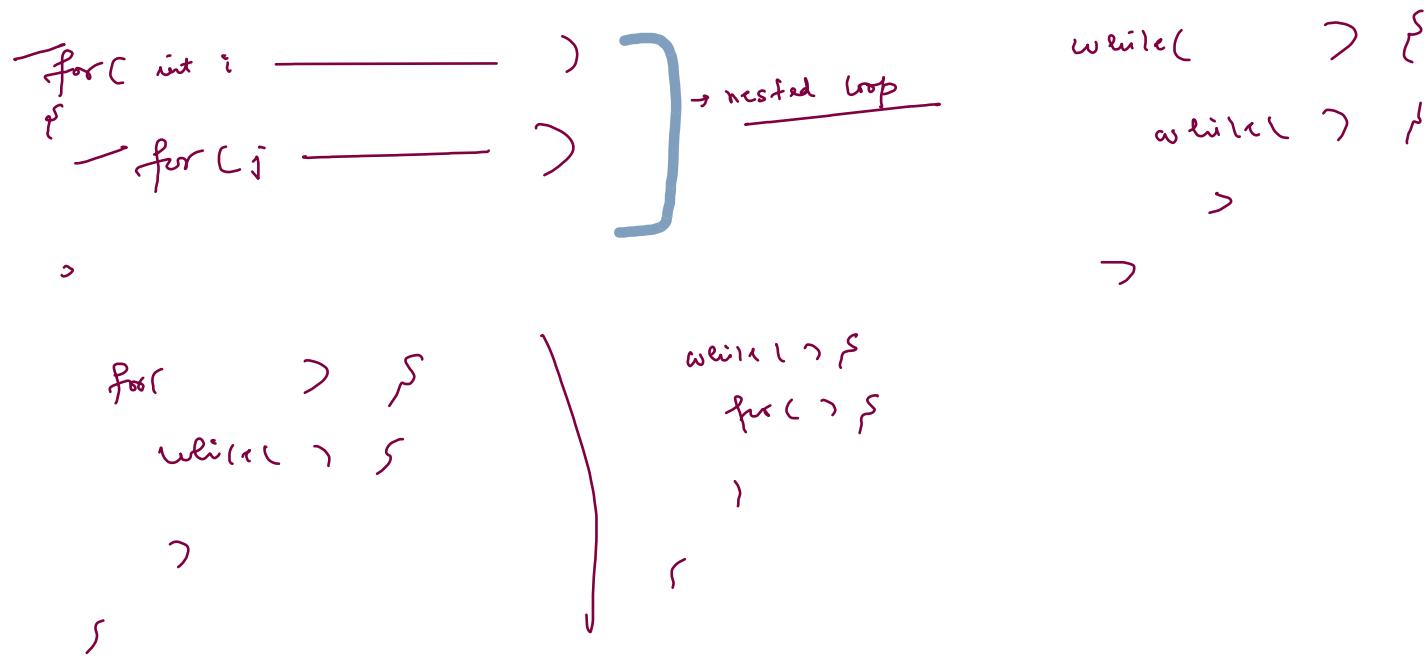
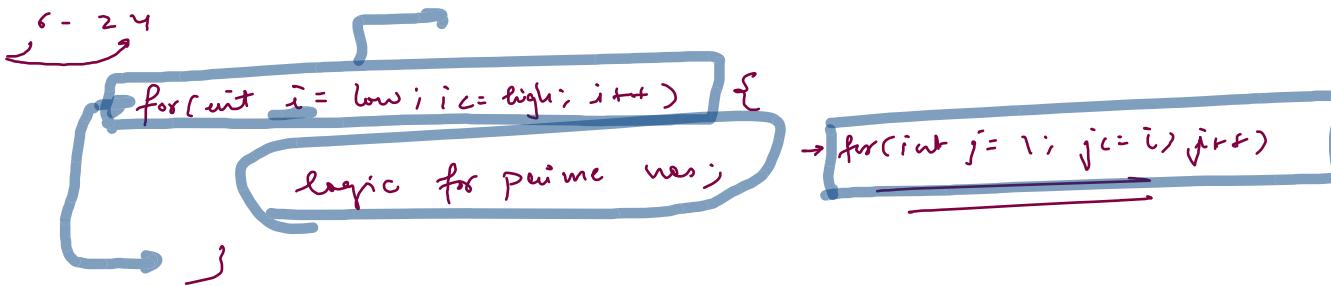
C
203. ~~200~~ X

~~10 23~~ → 23

1 C = n C = 70^-9

↳ ~~23~~ → 23

Print all prime nos.



```

for(int i=low;i<=high;i++) {
    int count = 0;
    for(int j = 1;j<=i;j++){
        if((j%2) == 0) count++;
    }
    if (count == 2) {
        System.out.println(i);
    }
}

```

for 1 value of i ,
 j loops max i times

$low = 6$, $high = 12$

$i = 6 \leftarrow 12(T)$

$count = 0$

$j = 1 \leftarrow 6(T)$

$6 \% 1 = 0(T)$

$count = 1$

$j = 2 \leftarrow 6(F)$

$6 \% 2 = 0(T)$

$count = 2$

$j = 3 \leftarrow 6(T)$

$6 \% 3 = 0(T)$

$count = 3$

$j = 4 \leftarrow 6(T)$

$6 \% 4 = 0(F)$

$count = 3$

$j = 5 \leftarrow 6(T)$

$6 \% 5 = 0(F)$

$count = 3$

$j = 6 \leftarrow 6(T)$

$6 \% 6 = 0(T)$

$count = 4$

$j = 7 \leftarrow 6(F)$

$count = 0$

$j = 1 \leftarrow 7$ — — —

$i = 8 \leftarrow 12(T)$

$i = 9$

$i = 10$

$i = 11 \quad i = 12 \quad \underline{i = 13} \leftarrow 12(F)$

$$n = 5$$

\times
 $\times \times$
 $\times \times \times$
 $\times \times \times \times$
 $\times \times \times \times \times$

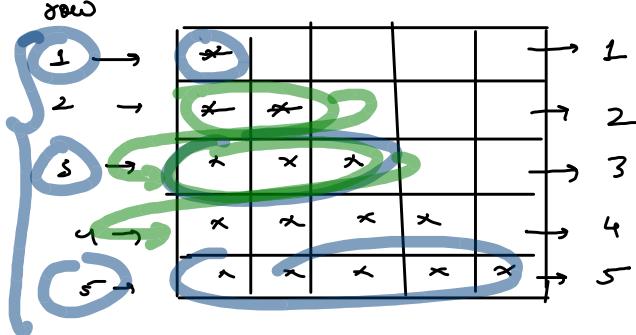
$$n = 3$$

\times
 $\times \times$
 $\times \times \times$

$$n = 7$$

\times
 $\times \times$
 $\times \times \times$
 $\times \times \times \times$
 $\times \times \times \times \times$
 $\times \times \times \times \times \times$
 $\times \times \times \times \times \times \times$

$n = 5$



$i \rightarrow 1$

In general

$i^{th} \rightarrow n^{st}$ of i

`for (int i = 1; i <= n; i++) { } → iterating over the rows.`

{

`for (int j = 1; j <= i; j++) { } → print next for a particular row.`

{

`System.out.print("*\t");`

`System.out.println();`

→ $n=5$

$i=1 \leftarrow 5(T)$

$j = 1 \leftarrow 1(T)$

*

$j = 2 \leftarrow 1(F)$

*

*

*

$i=2 \leftarrow 5(T)$

$j = 1 \leftarrow 2(T)$

$j = 2 \leftarrow 2(T)$

$j = 3 \leftarrow 2(F)$

$i=3 \leftarrow 5(T)$

$j = 1 \leftarrow 3(T)$

$j = 2 \leftarrow 3(T)$

$j = 3 \leftarrow 3(F)$

$j = 4 \leftarrow 3(F)$

$n = 5$

```

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

	1	2	3	4	5
1	*	*	*	*	*
2	*	*	*	*	
3	*	*	*		
4	*	*			
5	*				

row
 1
 2
 3
 4
 5

nest
 5 → n
 4 → n-1
 3 → n-2
 2 → n-3
 1 → n-4

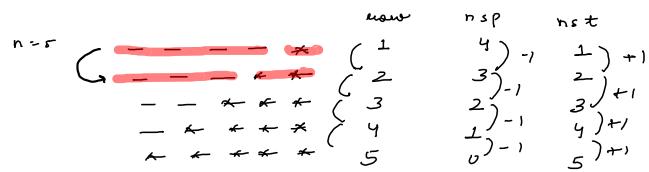
$n = 3$

```

* * *
* *
*
```

int nest = n;
 for(int i=1; i<=n; i++) - row
 {
 for(int j=1; j<=nest; j++) {
 cout << "*" << endl;
 }
 nest--;
}

i = 1	nest = 5
i = 2	nest = 4
i = 3	nest = 3
i = 4	nest = 2
i = 5	nest = 1



$n = 3$ $nsp = 2 \ (n-1)$

```

  - - x
  - x x
  x x x
  
```

$\text{int } nsr = 1, nsp = n-1$

$\text{for (int } i = 1; i < n; i++) \{$

$\left[\begin{array}{l} \text{for (int } j = 1, j < nsp; j++) \{ \\ \quad \text{sys}(\text{"*"}); \\ \} \\ \text{for (int } j = 1, j < nsr; j++) \{ \\ \quad \text{sys}(\text{"*\n"}); \\ \} \\ nsp--; \\ nsr++; \\ \text{sys}(); \\ \end{array} \right]$

$i = 4 \ l = 3 \ (F)$

$n = 3$ ~~$x \ x \ y$~~

$nsr = \cancel{x}, nsp = \cancel{x} \ x \ x - 1$

$i = 1 \ l = 3 \ (T)$

$j = 1 \ l = 2 \ (T)$

$j = 2 \ l = 2 \ (T)$

$j = 3 \ l = 2 \ (F)$

$j = 1 \ l = 1 \ (T)$

$j = 2 \ l = 1 \ (F)$

$i = 2 \ l = 3 \ (T)$

$j = 1 \ l = 1 \ (T)$

$j = 2 \ l = 1 \ (F)$

$j = 1 \ l = 2 \ (T)$

$j = 2 \ l = 2 \ (T)$

$j = 3 \ l = 2 \ (F)$

$i = 3 \ l = 3 \ (T)$

$j = 1 \ l = 0 \ (F)$

$j = 1 \ l = 3 \ (T)$

$i = 2 \ l = 3 \ (T)$

$j = 3 \ l = 3 \ (T)$

$j = 4 \ l = 3 \ (F)$

Pattern 04 $n = 5$

0 \times $\leftarrow x \times \leftarrow x$
 — $\times \leftarrow x \times$
 — $\leftarrow \leftarrow \leftarrow x$
 — $\leftarrow x \times$
 — $\leftarrow \leftarrow x$

row

1
2
3
4
5

$nsp = 0$ $nst = n$

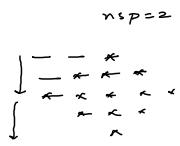
nsp	nst
0 $\downarrow +1$	5 $\downarrow -1$
1 $\downarrow +1$	4 $\downarrow -1$
2 $\downarrow +1$	3 $\downarrow -1$
3 $\downarrow +1$	2 $\downarrow -1$
4 $\downarrow +1$	1 $\downarrow -1$

 $n = 5$

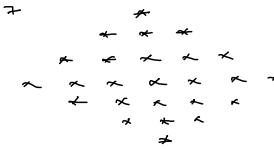
$\times \times \times \times \times$
 $\times \times \times \times$
 $\times \times \times$
 $\times \times$
 \times

Pattern - 5

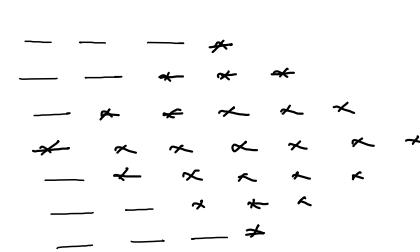
$n=5$



$n=7$



$n=7$



row

1

2

3

4

5

6

7

nsp

3
2
1

3
2
1

5
4
3

7
6
5

1
0
7

1
0
7

5
4
3

3
2
1

1
0
7

nst

2
3
5

3
2
5

5
4
7

7
6
4

5
4
7

3
2
5

1
0
7

$nsp = \frac{n}{2}$
 $nst = \frac{n}{2}$

$nsp +$
 $nst - = 2$

$n=5$

$$\frac{10}{2} = 2$$

$n=7$

$$\frac{7}{2} = 3$$

```

import java.io.*;
import java.util.*;

public class Solution {
    public static void main(String[] args) {
        Scanner scn = new Scanner(System.in);
        int n = scn.nextInt();
        int nsp = n/2, nst = 1;
        for(int i=1;i<=n;i++) {
            for(int j=1;j<=nsp;j++) {
                System.out.print(" ");
            }
            for(int j=1;j<=nst;j++) {
                System.out.print("*");
            }
            if (i<=n/2) {
                nsp--;
                nst+=2;
            } else {
                nsp++;
                nst-=2;
            }
            System.out.println();
        }
    }
}

```

$i = 1 \leftarrow s(i)$
 $\hookrightarrow j = 1 \leftarrow 2(T)$
 $j = 2 \leftarrow 2(T)$
 $j = 3 \leftarrow 2(F)$

 $\nearrow j = 1 \leftarrow 1(T)$
 $j = 2 \leftarrow 1(F)$

 $i = 3 \leftarrow s(T)$
 $\hookrightarrow j = 1 \leftarrow 0(F)$
 $j = 2 \leftarrow s(T)$
 $j = 3 \leftarrow s(T)$
 $j = 4 \leftarrow s(T)$
 $j = 5 \leftarrow s(T)$
 $j = 6 \leftarrow s(F)$

$n = 5$

$i = 2 \leftarrow s(T)$
 $\hookrightarrow j = 1 \leftarrow 1(T)$
 $j = 2 \leftarrow 1(F)$
 $\hookrightarrow j = 3 \leftarrow 1(F)$
 $j = 2 \leftarrow 3(LT)$
 $j = 3 \leftarrow 2(LT)$
 $j = 4 \leftarrow 3(F)$

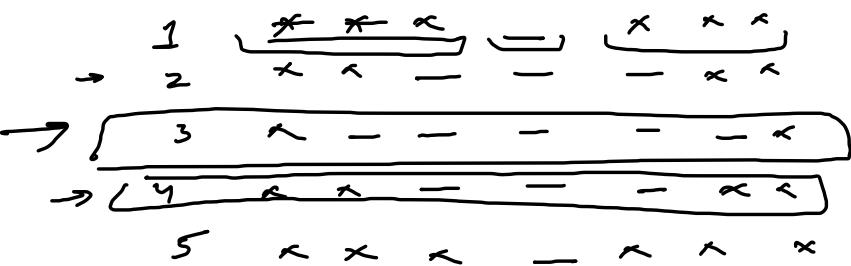
 $i = 4 \leftarrow s(T)$
 $\hookrightarrow j = 1 \leftarrow 1(T)$
 $j = 2 \leftarrow 1(F)$

 $j = 1 \leftarrow 3(T)$
 $j = 2 \leftarrow 3(T)$
 $j = 3 \leftarrow 3(T)$
 $j = 4 \leftarrow 3(F)$

$nsp = \frac{5}{2} = 2.5$ ~~$\times X \otimes X$~~ $\nearrow 3 \leftarrow$
 $nst = X \beta \not{X}$ $\nearrow -1 \times \beta$

~~$\times \times \times$~~
 ~~$\times \times \times \times \times$~~
 ~~$\times \times \times$~~
 ~~\times~~

$i = 6 \leftarrow s(F)$



row	nst 1	nsp	nst 2
1	$\boxed{3}$	$\frac{1}{2}) + 2$	$\frac{3}{2}) \rightarrow$
2	2	$3) + 2$	$1) \rightarrow$
3	1	$5) + 2$	$1) \rightarrow$
4	$2) + 1$	$3) - 2$	$2) + 1$
5	$3) + 1$	$1) - 2$	$3) + 1$

int nsp = 1, nst = $\frac{n}{2} + 1$

$i > 5$

~~nst += i;~~
~~nsp -= i;~~

~~3c : $\frac{5}{2}$~~

$3 > 2$

$n > 2$

$5 > 2$

$i=1 \quad nst = 3, nsp = 1$
 $\sum_{i=2}^{n/2} -2 + 1 \quad i=2 \quad nst = 2, nsp = 3$
 $i=3 \quad nst = 1, nsp = 5$

```

if (i<=n/2) {
    nst--;
    nsp+=2;
} else {
    nst++;
    nsp-=2;
}

```

$i=4 \quad nst = 2, nsp = 3$
 $i=5 \quad nst = 3, nsp = 1$

$$\underline{n = 5}$$

$\ast \ast \ast$	-	$\ast \ast \ast$	$\overset{3}{\underset{2}{\underset{1}{\underset{3}{)}}}}$	$\overset{1}{\underset{3}{\underset{5}{\underset{2}{)}}}}$	$\overset{3}{\underset{2}{\underset{1}{\underset{3}{)}}}}$
$\ast \ast$	-	-	$\overset{2}{\underset{1}{)}}$	$\overset{3}{\underset{5}{)}}$	$\overset{2}{\underset{1}{)}}$
\ast	-	-	$\overset{1}{\underset{2}{)}}$	$\overset{5}{\underset{3}{)}}$	$\overset{1}{\underset{2}{)}}$
$\ast \ast$	-	-	$\overset{2}{\underset{1}{)}}$	$\overset{3}{\underset{1}{)}}$	$\overset{2}{\underset{1}{)}}$
$\ast \ast \ast$	-	$\ast \ast \ast$	$\overset{3}{\underset{1}{)}}$	$\overset{1}{\underset{2}{)}}$	$\overset{3}{\underset{1}{)}}$

$$n=5, nst = 3$$

$$\frac{n}{2} = \frac{5}{2} = 2 + 1 = 3$$

if ($i < h/2$) {

1st - = 1

$nsp + -2;$

{ else

$$n s t + x = 1$$

$$\underline{n = 7}$$

n_{st1}	n_{sp}	n_{st2}
4	1	4
3	3	3
2	5	2
1	2	1
2	5	2
3	3	3
4	1	4

$$n=7, ns+t=4$$

$$\frac{n}{2} = \frac{7}{2} = 3 + 1 = 4$$

$$nsp = 1$$

$$n_{st} = \frac{n}{2} + 1$$

Functions

$${}^n C_r = \frac{n!}{r!(n-r)!}$$

$$n! = 1 * 2 * 3 * 4 * 5 * \dots * n$$

$$5 C_3 \Rightarrow \frac{5!}{3! * 2!} \Rightarrow \frac{5 * 4 * \cancel{3!}}{\cancel{3!} * 2} \Rightarrow \frac{20}{2} = 10$$

```
p. o. void main ( string [ ] args ) {
```

```
    int n = 5, r = 3;
```

factorial

```
    int ans1 = 1;  
    for ( int i=1; i<=n; i++ ) {  
        ans1 *= i;  
    }
```

```
    int ans2 = 1;  
    for ( int i=1; i<=r; i++ ) {  
        ans2 *= i;  
    }
```

```
    int ans3 = 1;  
    for ( int i=1; i<=(n-r); i++ ) {  
        ans3 *= i;  
    }
```

```
int ans = ans1  
(ans2 * ans3)
```

```
sysv( ans );
```

$$S1 := 1 * 2 * 3 * 4 * 5 = \underline{120}$$

$$ans1 = 1$$

$$\underline{i=1} \quad \underline{l=5 \text{ (T)}}$$

$$ans1 = 1 * 1 = 1$$

$$n(r = \underline{n!}) \\ \cancel{r!}(n-r) ?$$

$$i=2 \quad l=5 \text{ (T)}$$

$$ans1 = 1 * 2 = 2$$

$$n!, \quad r!,$$

$$i=3 \quad l=5 \text{ (T)}$$

$$ans1 = 2 * 3 = 6$$

$$(n-r)!$$

$$i=4 \quad l=5 \text{ (T)}$$

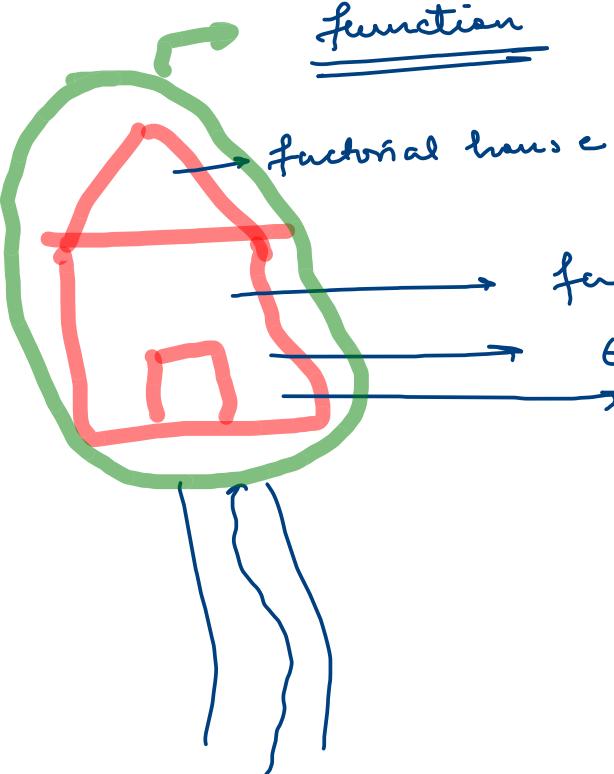
$$ans1 = 6 * 4 = 24$$

$$i=5 \quad l=5 \text{ (T)}$$

$$ans1 = 24 * 5 = 120$$

$$i=6 \quad l=\cancel{5 \text{ (F)}}$$

→ DRY → Don't repeat yourself



$n=5$

$n=6$

$n=10$

5!

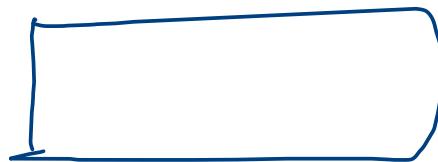
factorial(n)

6!

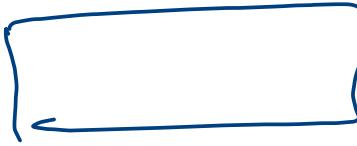
70!



→ factorial
(integer)



→ multiplication



→ addition



→ substraction

Syntax of function in Java

void

what type of value return by the function

public static return-type function-name (arguments) {

}

}

```
public static int factorial (int n) {  
    int ans = 1;  
    for (int i = 1; i <= n; i++) {  
        ans *= i;  
    }  
    return ans;  
}
```

```
main fn ()  
{  
    int ans1 = factorial(n);  
    int ans2 = factorial(x);  
    int ans3 = factorial(n - x);  
    int ans = ans1 / (ans2 * ans3);  
    System.out.println(ans);  
}
```

```
public static void main(String[] args) {  
    int n = 5, r = 3;  
  
    int ans1 = 1;  
    for(int i=1;i<=n;i++) {  
        ans1*=i;  
    }  
  
    int ans2 = 1;  
    for(int i=1;i<=r;i++) {  
        ans2*=i;  
    }  
  
    int ans3 = 1;  
    for(int i=1;i<=(n-r);i++) {  
        ans3*=i;  
    }  
  
    int ans = ans1/(ans2*ans3);  
  
    System.out.println(ans);  
}
```

function definition

```
public static int factorial(int x) {  
    int ans = 1;  
    for(int i=1;i<=x;i++) {  
        ans*=i;  
    }  
    return ans;  
}
```

```
public static void main(String[] args) {  
    int n = 5, r = 3;  
  
    int ans1 = factorial(n); - fn. calling  
    int ans2 = factorial(r);  
    int ans3 = factorial(n-r);  
  
    int ans = ans1/(ans2*ans3);  
  
    System.out.println(ans);  
}
```

function prototype

and fn defn.



no

implementation

"implementation"

```

public static int factorial(int x) {
    int ans = 1;
    for(int i=1;i<=x;i++) {
        ans*=i;
    }
    return ans;
}

```

```

public static void main(String[] args) {
    int n = 5, r = 3;

    int ans1 = factorial(n); → 120
    int ans2 = factorial(r);
    int ans3 = factorial(n-r);

    int ans = ans1/(ans2*ans3);
    System.out.println(ans); → 10
}

```

factorial(2) $i = 1 \leftarrow 2 \text{ (T)}$ $i = 3 \leftarrow 2 \text{ (F)}$ $ans = 1 * 1 * 2 = 2$	$i = 1 \leftarrow 3 \text{ (T)}$ $i = 3 \leftarrow 3 \text{ (T)}$ $i = 2 \leftarrow 3 \text{ (T)}$ $j = 4 \leftarrow 3 \text{ (F)}$ $ans = 1 * 1 * 2 * 3 = 6$	$i = 1 \leftarrow 5 \text{ (T)}$ $i = 3 \leftarrow 5 \text{ (T)}$ $i = 2 \leftarrow 5 \text{ (T)}$ $i = 4 \leftarrow 5 \text{ (T)}$ $i = 6 \leftarrow 5 \text{ (F)}$ $ans = 1 * 1 * 2 * 3 * 4 * 5 = 120$	$ans = 120 / (6 * 2) = 10$ ans3 = factorial(2) ans2 = factorial(3) 6 ans1 = factorial(5) 120 $n = 5, r = 3$
			Stack

Find frequency

$$n = 111 \underset{1}{2} \underset{3}{3} \underset{4}{4} \underset{3}{3} \rightarrow \textcircled{2}$$

$d = 3$

$$n = \underset{1}{1} \underset{1}{1} \underset{1}{1} \underset{2}{3} \underset{3}{3} \underset{4}{4} \underset{3}{3}$$
$$d = 1 \rightarrow \textcircled{3}$$

Count digit \rightarrow To reduce no. \rightarrow 1/10

\rightarrow % 10

$n = 1112343 \rightarrow n > 0$ $\frac{d=3}{}$ $\text{Count} = 8/12 \rightarrow 2$

$n = 1112343$
 ↳ $1112343 \% 10 \Rightarrow \underline{\underline{3}}$
 ↳ $1112343 / 10 \rightarrow \underline{\underline{111234}}$

$n = 111234$
 ↳ $111234 \% 10 = \underline{\underline{4}}$
 ↳ $111234 / 10 = \underline{\underline{11123}}$

$n = 11123$
 ↳ $11123 \% 10 \Rightarrow \underline{\underline{3}}$
 ↳ $11123 / 10 \rightarrow \underline{\underline{1112}}$

1112
 ↳ $1112 \% 10 \Rightarrow \underline{\underline{2}}$
 $1112 / 10 \rightarrow \underline{\underline{111}}$

111
 ↳ $111 \% 10 = \underline{\underline{1}}$
 $111 / 10 = \underline{\underline{11}}$

11
 ↳ $11 \% 10 = \underline{\underline{1}}$
 $11 / 10 = \underline{\underline{1}}$

1
 ↳ $1 \% 10 = \underline{\underline{1}}$
 $1 / 10 = \underline{\underline{0}}$

Decimal to Binary

Assumption

$n > 0$

$$(47)_{10} \rightarrow (?)_2$$

Subtract remainder

2	47	(1)
2	2 ³	(1)
2	11	(1)
2	5	(1)
2	2	(0)
2	1	(1)

$\frac{n}{2} \rightarrow 0$
divide by 2
 $n = 47$ remainder \Rightarrow part of answer
 \downarrow subtract
new value of n

$(\text{position digit}) \times 2^{\text{position}}$

5	3	2	1	0
(10)	(1)	(1)	(1)	
$\times 2^0$	$\times 2^1$	$\times 2^2$	$\times 2^3$	$\times 2^4$
1	2	4	8	16

$\rightarrow 32 + 0 + 8 + 4 + 2 + 1$
 $\rightarrow 47$

$$(74)_{10} \rightarrow (?)_2$$

$$(1001010)_2$$

2	74	0
2	37	1
2	18	0
2	9	1
2	4	0
2	2	0
2	1	1
	0	

Answer

decimal to binary

→ answer → reverse

dividing by 2
 $n = 42 \xrightarrow{\text{remainder}}$
 ↓ quotient
 $\text{new value of } n$

part of answer

int answer = 0;

while ($n > 0$) {

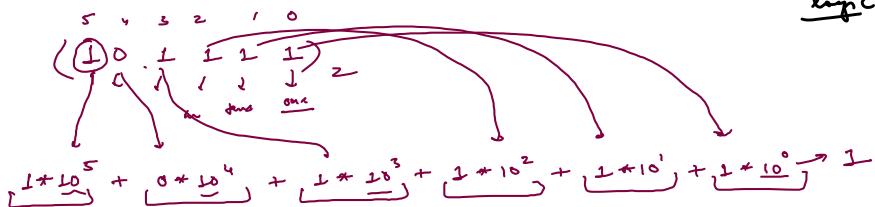
int rem = $n \% 2$;

int quo = $n / 2$;

answer = $\frac{1}{\downarrow} \frac{1}{\downarrow}$ $\frac{\text{answer} * 10 + \text{rem}}{\downarrow}$;
 $n = \text{quo};$

3 return reverse(answer);

logic



$n = 42$ answer = 0
 $42 > 0 \rightarrow \text{rem} = 1 \quad \text{answ} = 23 \quad \text{answer} = 0 * 10 + 1$
 $= 1$

$n = 23 > 0 \rightarrow \text{rem} = 1 \quad \text{answ} = 11 \quad \text{answer} = 1 * 10 + 1$
 $= 11$

$n = 11 > 0 \rightarrow \text{rem} = 1 \quad \text{answ} = 111 \quad \text{answer} = 11 * 10 + 1$
 $= 111$

$n = 5 > 0 \rightarrow \text{rem} = 1 \quad \text{answ} = 1111 \quad \text{answer} = 1111 * 10 + 1$
 $= 11111$

$n = 2 > 0 \rightarrow \text{rem} = 0 \quad \text{answ} = 11110 \quad \text{answer} = 11110 * 10 + 0$
 $= 111100$

$n = 1 > 0 \rightarrow \text{rem} = 1 \quad \text{answ} = 111101 \quad \text{answer} = 111101 * 10 + 1$
 $= 1111011$

$n = 0 > 0$

reverse(111101) \Rightarrow 101111

Decimal to binary

→ method - 02

dividing by 2
 $n = 42$ remainder
 ↓ quotient
 new value of n

stopping condition when $n=0$

part of answer

```
int answer = 0, power = 1;
while (n > 0) {
    int rem = n % 2;
    int quo = n / 2;
}
```

answer = power * rem + answer;

$n = \text{quo}$

power *= 10;

}

return answer;

}

answer = 0, power = 1;

$n = 42 > 0$

↳ rem = 1

↳ quo = 23

answer = $1 * 1 + 0$
 = $\underline{\underline{1}}$

power = $10 + 10 = 10$

$n = 23 > 0$

↳ rem = 1

↳ quo = 12

answer = $10 * 1 + 1$
 = 11

power = 100

$n = 11 > 0$

↳ rem = $\underline{\underline{1}}$

↳ quo = 5

answer = $100 + 11$
 = 111

power = 100 0

$n = 5 > 0$

↳ rem = 1

↳ quo = 2

answer = $1000 + 122$
 = 1122

power = 100 0 0

$n = 2 > 0$

↳ rem = 0

↳ quo = 1

answer = $0 * \underline{\underline{\underline{\underline{1}}}} + 1111$
 = 1111

power = 100000 0

$n = 1 > 0$

↳ rem = 1

↳ quo = 0

answer = $100000 + 1111$
 = 10111

power = 10^6

$n = 0 > 0 \times$

decimal to binary

(2)
1

b = 5

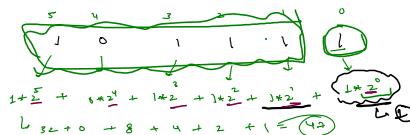
```
import java.io.*;
import java.util.*;

public class Solution {
    public static int dtob(int n) { → ①
        int answer = 0, power = 1; → ②
        while(n>0) { → ③
            int rem = n%2; → ④
            int quo = n/2; → ⑤
            answer += (rem*power); → ⑥
            power *=10; → ⑦
            n = quo; → ⑧
        }
        return answer; → ⑨
    }

    public static void main(String[] args) {
        /* Enter your code here. Read input from STDIN. Print output to STDOUT. Your class should
        be named Solution. */
        System.out.println(dtob(47));
    }
}
```

binary to decimal

$$(10111)_2 \rightarrow (77)_{10}$$



binary to decimal

(position digit) * 2^{position}

$$n = 10111 \backslash ;$$

answer = 0, power = 1 → ①

while ($n > 0$) { → ②

int rem = $n \% 10$ → ③

int q = $n / 10$ → ④

answer = answer + rem * power → ⑤

power *= 2 → ⑥

$n = q$; → ⑦

$$n = 10111 \backslash ;$$

answer = 0, power = 1

$$n = 10111 > 0$$

↳ rem = 1

↳ q = 10111, power = $1 \times 2 = 2$

$$n = 10111 > 0$$

↳ rem = 1

↳ q = 1011

answer = $1 + 1 \times 2 = 3$

$$n = 1011 > 0$$

↳ rem = 1

↳ q = 101

answer = $1 + 1 \times 2 = 3$

power = $1 \times 2 = 2$

$$n = 101 > 0$$

↳ rem = 1

↳ q = 10

answer = $3 + 1 \times 2 = 7$

power = $1 \times 2 = 2$

$$n = 101 > 0$$

↳ rem = 1

↳ q = 10

answer = $7 + 1 \times 2 = 15$

power = $1 \times 2 = 2$

$$n = 10 > 0$$

↳ rem = 0

↳ q = 1

answer = $15 + 0 \times 2 = 15$

power = $1 \times 2 = 2$

$$n = 1 > 0$$

↳ rem = 1

↳ q = 0

answer = $15 + 1 \times 2 = 17$

power = $1 \times 2 = 2$

$$n = 0 > 0 (\text{False})$$

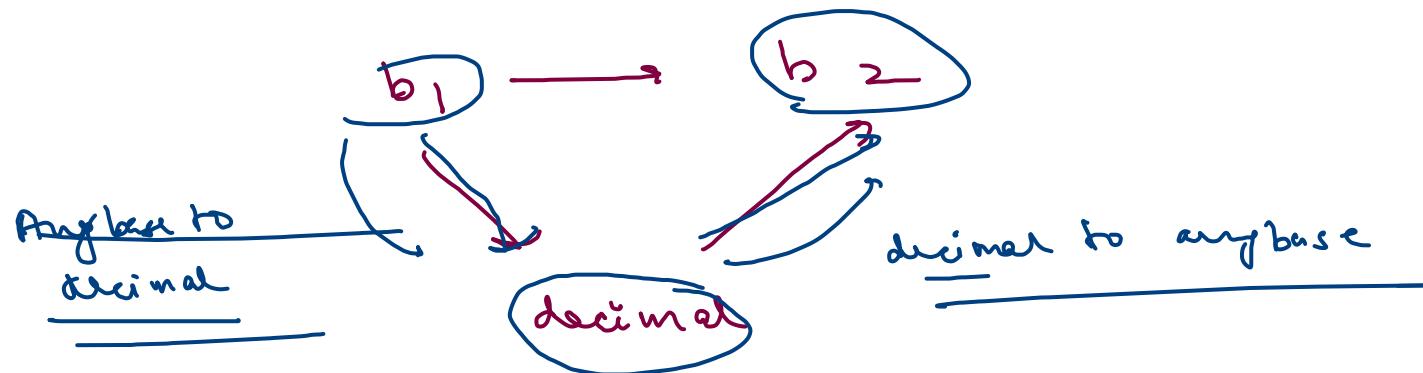
(decimal) $_{10}$

$b_1 \rightarrow$ decimal

decimal $\rightarrow b_2$

$b_1 \rightarrow b_2$

(n) b_1 \rightarrow (?) b_2



Any base addition

(10)

$$1(10^0) \quad 14(10^1) \quad 14(10^2) \quad 12(10^3)$$

$$\rightarrow n_1 = 0 \quad 7 \quad 9 \quad 7$$

$$n_2 = 0 \quad 6 \quad 4 \quad 5$$

$$h_1=0, h_2=0,
carry=0$$

$$\begin{array}{r} & 1^0/10 & 14^0/10 & 14^1/10 & 12^2/10 \\ \hline & 1 & 4 & 4 & 2 \\ \hline \end{array}$$

$$9+4+1 = 14 \quad 14/10 = 1$$

$$14/10 = 1$$

$$7+6+1 = 14 \quad 14/10 = 1$$

$$14/10 = 1$$

$$7+5 =$$

$$12 \% /10 = 2 \rightarrow \text{answer}$$

$$12/10 = 1 \rightarrow \text{carry}$$

$$0+0+1 = 1 \quad 1/10 = 1$$

$$1/10 = 0$$

carry → 0

$n_1 \rightarrow \begin{array}{r} 1 \\ 0 \\ \hline 0 \end{array}$

$n_2 \rightarrow \begin{array}{r} 1 \\ 9 \\ 4 \\ 5 \\ 7 \\ 6 \\ \hline + 0 \quad 8 \quad 7 \quad 6 \end{array}$

$\xrightarrow{\qquad\qquad\qquad}$

$1^{\circ}/_{10} \quad 10^{\circ}/_{10} \quad 13^{\circ}/_{10} \quad 13^{\circ}/_{10} \quad \underline{13^{\circ}/_{10}}$

$1 \quad 0 \quad 3 \quad 3 \quad \textcircled{3}$

1 0 3 3 3

$n_1 = 0$
 $n_2 = 0$
carry 1 = 0

$$7+6 = 13^{\circ}/_{10} = 3 - \text{answer}$$

$$13^{\circ}/_{10} = \underline{1} \rightarrow \text{carry}$$

$$5+7+1 = \underline{13^{\circ}/_{10}} = 3$$

4 answer

$$13^{\circ}/_{10} = \underline{1} \rightarrow \text{carry}$$

$$4+8+1 = 13^{\circ}/_{10} = 3$$

4 answer

0 0

0 4 5

0 5 4

 +

9 9

$$5+4+0 = 9^{\circ}/_{10} = 9$$

$$9/_{10} = 0$$

$$4+5+0 = 9^{\circ}/_{10} = 9$$

$$9/_{10} = 0$$

$[3/_{10} = 1]$
a carry.

$$0+9+1 = 10^{\circ}/_{10} = 0$$

$$10/_{10} = 1$$

$$5+0+1 = 1^{\circ}/_{10} = 1$$

$$1/_{10} = 0$$

$$\text{int sum} = \frac{(n_1 + n_2)}{10} = 1$$

$$\sum = \frac{n_1 + n_2}{10}$$

base = b

$$\begin{array}{r} n_1 = 7 \\ 7 \\ + 4 \\ \hline 11 \\ \text{carry} = 1 \end{array}$$

int answer = 0; int carry = 0;

int power = 1;

```
while (n1 > 0 || n2 > 0 || carry > 0) {
    int d1 = n1 % 10; -①
    int d2 = n2 % 10; -②
    int sum = d1 + d2 + carry; -③
    answer = answer + (sum % 10) * power; -④
    carry = sum / 10; -⑤
    n1 = n1 / 10; -⑥
    n2 = n2 / 10; -⑦
    power = power * 10; -⑧
}
```

$n_1 = 739, n_2 = 45, \text{answer} = 0, \text{carry} = 0,$

$$\begin{array}{r} n_1 = 739 \\ n_2 = 45 \\ \hline 784 \end{array}$$

$$739 \% 10 = 9$$

$$45 \% 10 = 5$$

$$9 + 5 + 0 = 14 \% 10 = 4$$

$$\text{carry} = 14 / 10 = 1$$

$$739 / 10 = 73 \rightarrow n_1$$

$$45 / 10 = 4 \rightarrow n_2$$

[0 - 9]

$$(7 \quad 7 \quad 7) \rightarrow [0 - 7]$$

$$\begin{array}{r} 0 \quad 2 \quad 1 \quad 1 \\ \times 0 \quad 0 \quad 7 \quad 7 \quad 7 \\ \hline 0 \quad 0 \quad 0 \quad 0 \quad 1 \\ 0/8 = 0 \quad 8/8 = 0 \quad 8/8 = 0 \\ \hline 1 \end{array}$$

$$\begin{array}{r} 1000 \quad (b) \end{array}$$

$$7 + 1 + 0 = 8^{\circ} / 8 = 1$$

$$n_1 = 2, n_2 = 0$$

$$n_1 > 0 (\text{True})$$

$$\text{sum} = 7 + 0 + 0 = 7$$

$$d_1 = 7 \% 10 = 7$$

$$d_2 = 0 \% 10 = 0$$

$$\text{answer} = 84 + 7 * 100 = 784$$

$$\text{carry} = 7 / 10 = 0$$

$$\text{power} = 100$$

$$n_1 = 0, n_2 = 0, \text{carry} = 0$$

$$F \quad F \quad F$$

$$7 + 0 + 1 = 8^{\circ} / 8 = 0$$

$$8 / 8 = 1$$

$$7 + 0 + 1 = 8^{\circ} / 8 = 0$$

$$8 / 8 = 1$$

$$1 + 0 + 0 = 1^{\circ} / 8 = 1$$

$$1 / 8 = 0$$

$$\begin{array}{r} 7 \quad 7 \quad 7 \\ \times 2 \quad 8 \\ \hline 1 \quad 0 \quad 0 \quad 0 \quad \text{decimal (10)} \end{array}$$

```

public static int anyBaseAddition(int b, int n1, int n2) {
    int answer = 0, power = 1, carry = 0;

    while(n1>0 || n2>0 || carry>0) {
        int d1 = n1%10;
        int d2 = n2%10;

        int sum = d1+d2+carry;
        answer = answer + (sum%b)*power;
        carry = sum/b;

        n1 = n1/10;
        n2 = n2/10;
        power = power*10;
    }

    return answer;
}

```

$\rightarrow 5 \text{ mins}$

$$\begin{aligned}
 n_1 &= 845 \\
 n_2 &= 736 \\
 b &= 6
 \end{aligned}$$

$\rightarrow 10 \text{ mins}$

→ day min

$$\begin{aligned}
 b &= 8, n_1 = 227, n_2 = 1 \\
 \text{answer} &= 0, \text{power} = 1, \text{carry} = 0
 \end{aligned}$$

10:23 pm

$n_1 > 0 \text{ (T)}$

$$\begin{aligned}
 \hookrightarrow d_1 &= 227 \% 10 = 7, \quad \text{sum} = 221 + 0 = 8 \\
 \hookrightarrow d_2 &= 1 \% 10 = 1 \quad \text{answer} = 0 + (8 \% 8) * 1
 \end{aligned}$$

$$= 0$$

$$\text{carry} = 8 / 8 = 1$$

$$\begin{aligned}
 n_1 &= 227 / 10 = 22 \quad \text{power} = 10 \\
 n_2 &= 1 / 10 = 0
 \end{aligned}$$

1

$n_1 > 0 \text{ (T)}$

$$\hookrightarrow d_1 = 22 \% 10 = 2$$

$$\hookrightarrow d_2 = 0 \% 10 = 0$$

$$\text{sum} = 220 + 1 = 8$$

$$\begin{aligned}
 \text{answer} &= 0 + (8 \% 8) * 10 \\
 &= 0
 \end{aligned}$$

0 20 10

$$\text{carry} = 8 / 8 = 1$$

$$n_1 = 22 / 10 = 2, \quad n_2 = 0 / 10 = 0, \quad \text{power} = 100$$

$n_1 > 0 \text{ (T)}$

$$\hookrightarrow d_1 = 2 \% 10 = 2$$

$$\hookrightarrow d_2 = 0 \% 10 = 0$$

$$\text{sum} = 2 + 0 + 1 = 3$$

$$\begin{aligned}
 \text{answer} &= 0 + (3 \% 6) * 100 \\
 &= 0
 \end{aligned}$$

$$\begin{aligned}
 \text{carry} &= 3 / 6 = 1, \quad n_1 = 2 / 10 = 0, \quad n_2 = 0 / 10 = 0, \\
 \text{power} &= 1000
 \end{aligned}$$

$n_1 = 0, n_2 = 0, \text{carry} = 1 \text{ (F)}$

$$\text{sum} = 0 + 0 + 1 = 1$$

$$\hookrightarrow d_1 = 0 \% 10 = 0$$

$$\hookrightarrow d_2 = 0 \% 10 = 0$$

$$\text{answer} = 0 + (1 \% 6) * 1000$$

$$= 0 + 1 * 1000 \rightarrow 1000$$

$$\begin{aligned}
 \text{carry} &= 1 / 6 = 0, \quad n_1 = 0 / 10 = 0, \quad n_2 = 0 / 10 = 0, \\
 \text{power} &= 10000
 \end{aligned}$$

$n_1 = 0, n_2 = 0, \text{carry} = 0$

$\downarrow \quad \downarrow \quad \downarrow$

(F) (F) (F) X

$\cancel{1 - 108 = 7}$

```

public static int anyBaseAddition(int b, int n1, int n2) {
    int answer = 0, power = 1, carry = 0;

    while(n1>0 || n2 > 0 || carry > 0) {
        int d1 = n1%10;
        int d2 = n2%10;

        int sum = d1+d2+carry;
        answer = answer + (sum%b)*power;
        carry = sum/b;

        n1 = n1/10;
        n2 = n2/10;
        power = power*10;
    }

    return answer;
}

```

$$n_1 = 845 \quad n_2 = 736, \quad b = 6$$

answer = 0, power = 1, carry = 0

$$\begin{aligned} n_1 &> 0 \text{ (T)} \\ \hookrightarrow d_1 &= 845 \% 10 = 5 & \text{sum} &= 5 + 6 + 0 = 11 \\ \hookrightarrow d_2 &= 736 \% 10 = 6 & \text{answer} &= 0 + (11 \% 6) * 1 \\ & & &= 0 + 5 * 1 = 5 \\ n_1 &= 845 / 10 = 84 \\ n_2 &= 736 / 10 = 73 \\ \text{carry} &= 11 / 6 = 1 & \text{power} &= 10 \end{aligned}$$

$$\begin{aligned} n_1 &> 0 \text{ (T)} \\ \hookrightarrow d_1 &= 84 \% 10 = 4 & \text{sum} &= 4 + 3 + 1 = 8 \\ \hookrightarrow d_2 &= 73 \% 10 = 3 & \text{answer} &= 5 + (8 \% 10) * 10 \\ & & &= 5 + 20 = 25 \\ \text{carry} &= 8 / 6 = 1 \\ n_1 &= 84 / 10 = 8, n_2 = 73 / 10 = 7, \quad \text{power} = 100 \end{aligned}$$

$$\begin{aligned} n_1 &> 0 \text{ (T)} \\ \hookrightarrow d_1 &= 8 \% 10 = 8 & \text{sum} &= 8 + 7 + 1 = 16 \\ \hookrightarrow d_2 &= 7 \% 10 = 7 & \text{answer} &= 25 + (16 \% 6) * 100 \\ & & &= 25 + 4 * 100 = 425 \end{aligned}$$

$$\begin{aligned} \text{carry} &= 16 / 6 = 2 \\ n_1 &= 8 / 10 = 0, n_2 = 7 / 10 = 0, \quad \text{power} = 1000 \end{aligned}$$

$$\begin{aligned} n_1 &> 0 \text{ (F)}, \quad n_2 > 0 \text{ (F)}, \quad \text{carry} > 2 \text{ (T)} \\ \hookrightarrow d_1 &= 0 \% 10 = 0 & \text{sum} &= 0 + 0 + 2 = 2 \\ \hookrightarrow d_2 &= 0 \% 10 = 0 & \text{answer} &= 425 + (2 \% 6) * 1000 \\ & & &= 425 + 2000 = 2425 \\ n_1 &= 0 / 10 = 0, n_2 = 0 / 10 = 0, \quad \text{power} = 10000 \end{aligned}$$

$$n_1 > 0 \text{ (F)}, \quad n_2 > 0 \text{ (F)}, \quad \text{carry} > 0 \text{ (F)}$$

Take in a character as an input from the user,

A. Condition 1 : If the character is an alphabet then you need to toggle the character first,

For example, if the entered character is 'a', then convert it into 'A', and if the entered character is 'A' then convert it into 'a', this simply means that if the entered character is a capital case then convert it into a small case character and vice-versa. After toggling the character, a. if the resultant character is not 'a', 'A', 'b', 'B', then take two jumps to the left and print the character, for eg. If the toggled character is 'c' then print 'a', If the toggled character is 'Z', then print 'X'. b. If the toggled character is 'a', 'A', 'b', 'B', then print "Sorry". B. Condition 2: Otherwise print "No alphabet"

ch = 'c'

→ 'C'

A B C

→ 'a'

a, A, b, B → Sorry

'Z' → 'X'

ch = 'P'

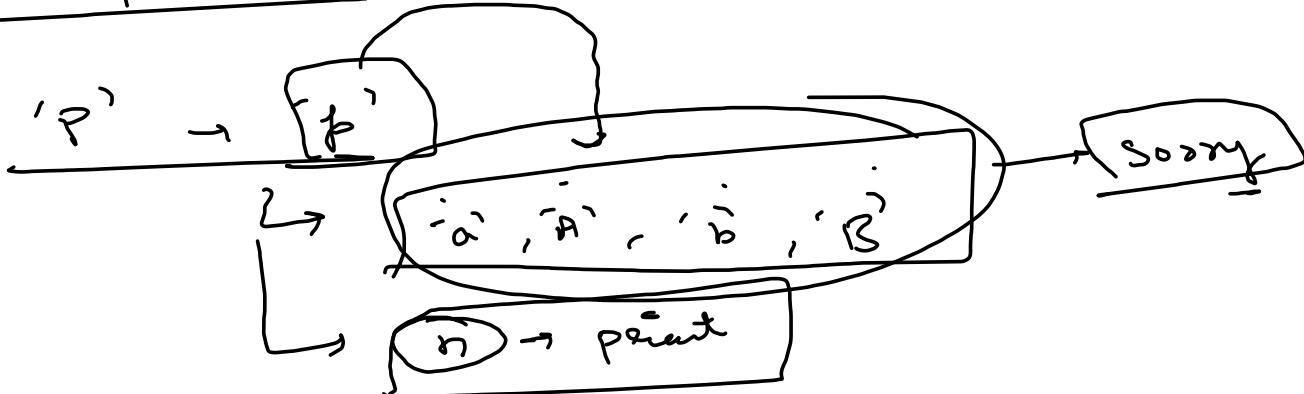
ch = 'Z'

Check given character is an alphabet \rightarrow a-z, A-Z

\hookrightarrow No \Rightarrow No alphabet

\hookrightarrow Yes \rightarrow

'P'



a b c d e f g h i j k l m n o p q r — —

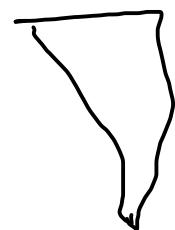
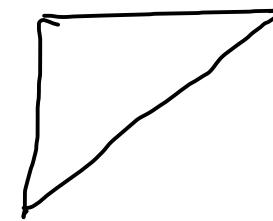
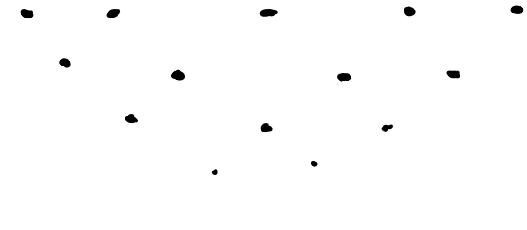
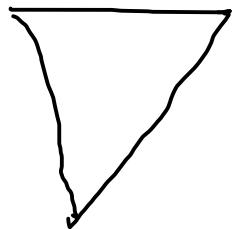
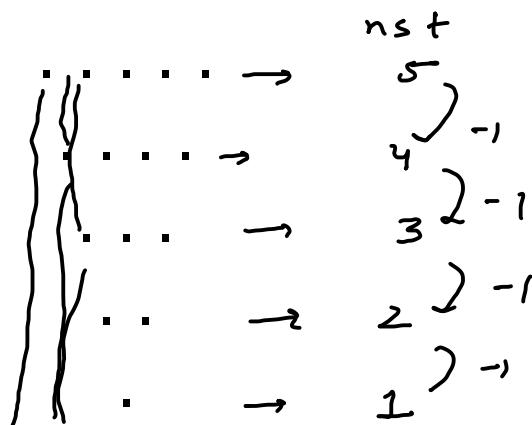
HW - Pattern -01 and Pattern -02 (done
but need to fix the test cases)

Coderbyte questions -

9-11 AM

* * * x x
x x x x
x x x
x x
x

→ 2 min



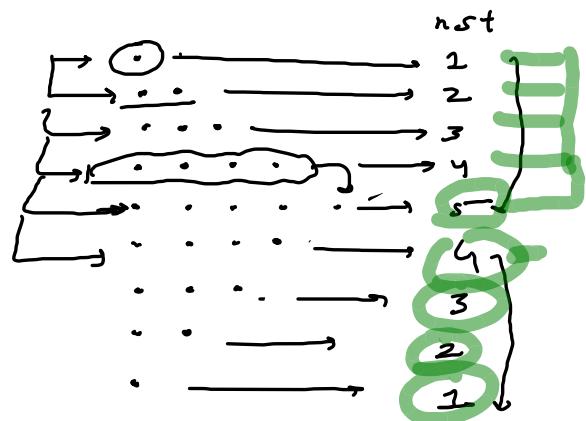
$n = 5$

```
public static void pattern(int n) {  
    int nsp = 0, nst = n;  
    for(int i=1;i<=n;i++) { → ↗ ↗ ↗  
        for(int j=1;j<=nsp;j++) { }  
        System.out.print(" ");  
    }  
    for(int j=1;j<=nst;j++) { }  
    System.out.print(". "); }  
    nst--;  
    nsp++; }  
    System.out.println(); } }
```

nst	nsp
5	0)
4	1)
3	2)
2	3)
1	4)

Pattern-02

$n = 5$



$n = 5$

$2 \times n - 1 \rightarrow \text{rows}$

$nst = 1$

$\text{if } (i \leq (n-1))$

$nst++;$

else

$nst--;$

$\rightarrow \text{next rows}$

$nst = 1$

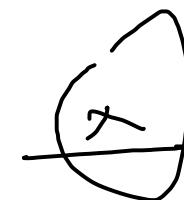
.....
.....
.....
.....
.....

if

Maximum Discent

↳ Sarah \rightarrow Bill \rightarrow Y

↳ $15^\circ/\circ \rightarrow \frac{Y \times 15}{1^\circ} =$



↳ 20

Maximum Discent

$f(x > 20)$ return ∞ ;

else return 20;

Find Sum

n = 15

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

ans = 7