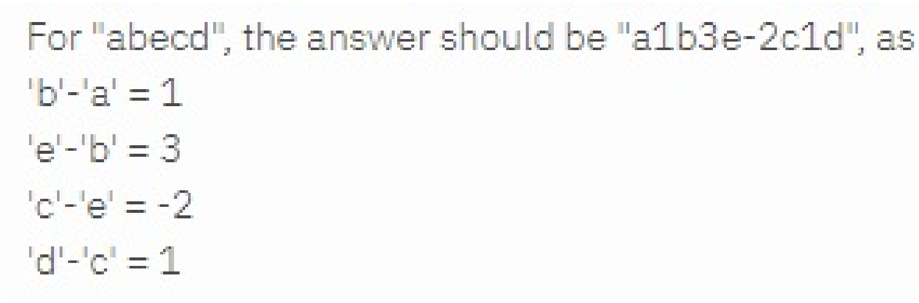
abecd

char

a 1 b 3 e -2 c 1 d

a
b
c
d
e

$\boxed{\text{a1b3e-2c1d}}$



For "abecd", the answer should be "a1b3e-2c1d", as

'b'-'a' = 1

'e'-'b' = 3

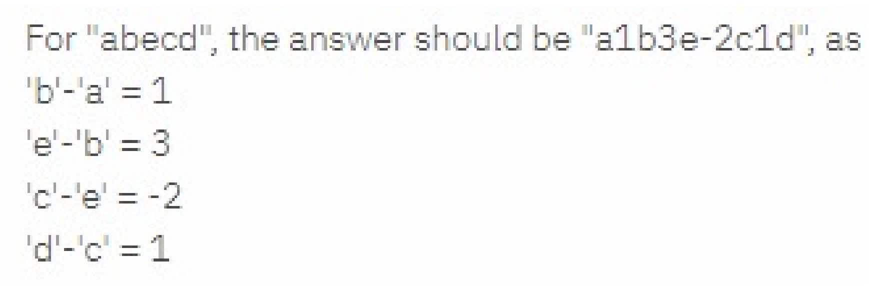
'c'-'e' = -2

'd'-'c' = 1

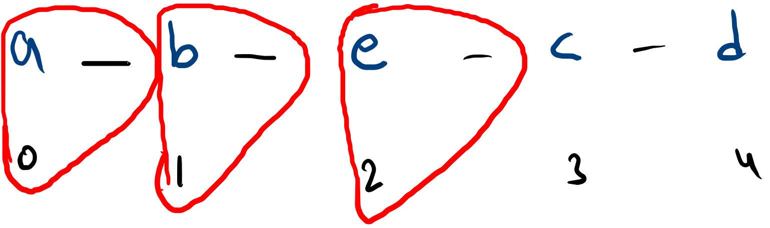
x = 'b'

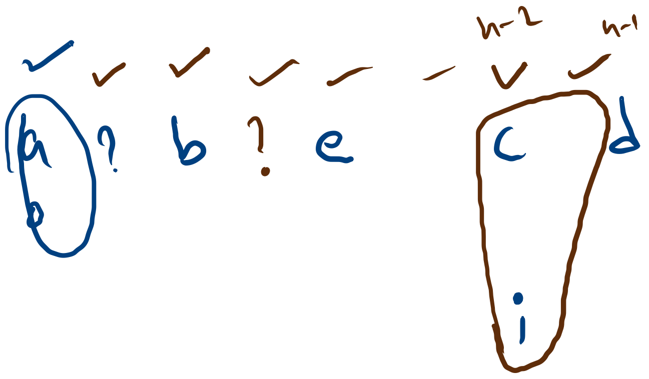
y = 'a'

x-y


For "abecd", the answer should be "a1b3e-2c1d", as
'b'-'a' = 1
'e'-'b' = 3
'c'-'e' = -2
'd'-'c' = 1

str = " abecd





```
i ← 0 to n-2
[
    sb append ith char
    diff = (i+1)th - (i)th
    append diff
]
append last character
```

sb a-b-e

Sb=a1b3e-2C1

'c' - 'e' = -2

'e' - 'b'

'd' - 'c' = 1

Iteration

str → abc
0 1 2

bc
1

b

acb

n % 3 — 0
— 1
— 2

n % 2 → 0
→ 1

abc

bac

cab

acb

bca

cba

abc → n   3! = 6 → $\boxed{0 \text{ to } 5}$ $\boxed{1 \text{ to } 6}$ $\boxed{2 \text{ to } 7}$

0

	q	r
3	0	
2	0	0
1	0	0
	0	0

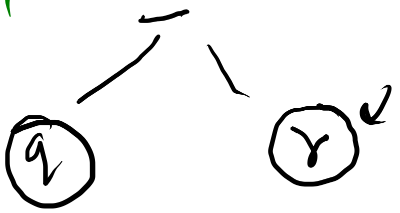
~~a~~bc
0 1 2

~~b~~c
0 1

c

ans : $\boxed{abc}$

1



2

```
total = n!

val ← 0 to total-1
    new stringbuilder (str);
    ans SB
    int n = val;
    div ← n to 1
        q   n / div
        r   n % div
        append rth char
        delete rth char
        n = q
```

3

	n	r
div	q	
div		

sub→ a~~b~~c

ans→

4

n	3	4
2	1	1
1	0	1
	0	0

a~~b~~c
0 1 2

a~~c~~
0 1

a
0

ans bca

5

3	5	
2	1	2
1	0	1
	0	0

ab~~c~~
0 1 2

a~~b~~
0 1

a
0

ans $\boxed{cba}$

DS → Array

size fixed

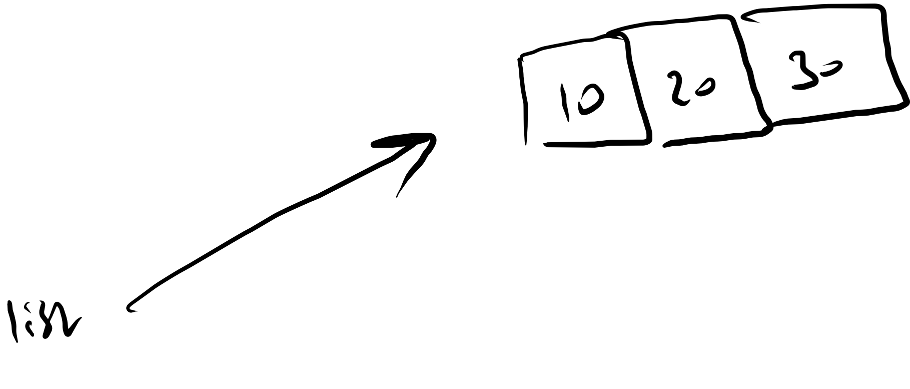
Arraylist

insert / delete

70	80	90


```
Arraylist<Integer> list = new Arraylist<>();
```

Array	Arraylist
ar.length	list.size()
X	list.add(10)
n= ar[i]	n= list.get(i)
ar[i]= n	list.set(i, n);



al → [$\dot{3}$, 12 , $\dot{13}$, 15]

remove(i)

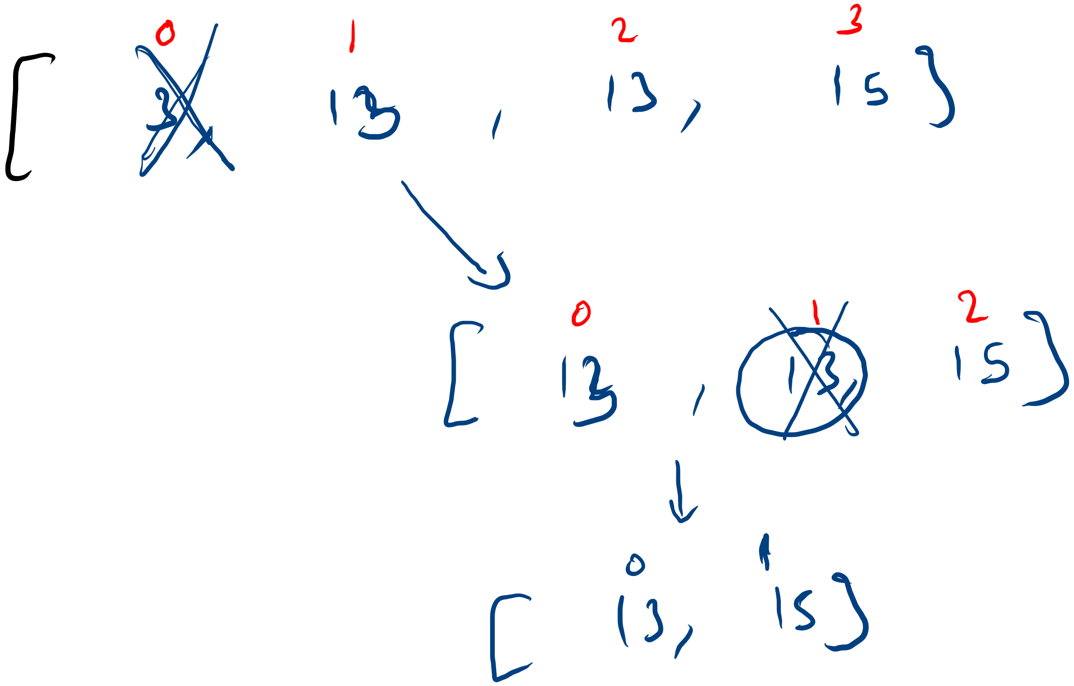
[12, 15]

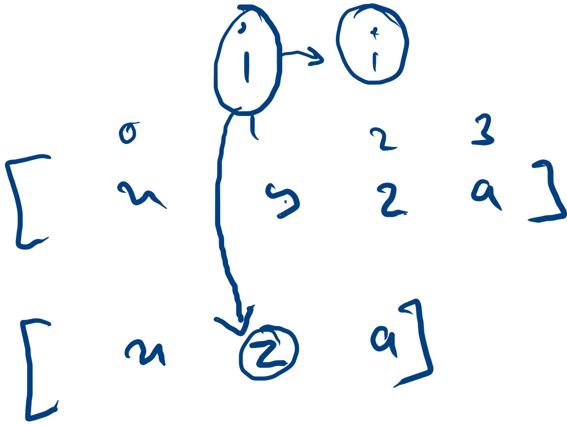
[3 , 13, 12, 15]

3 12 13 15

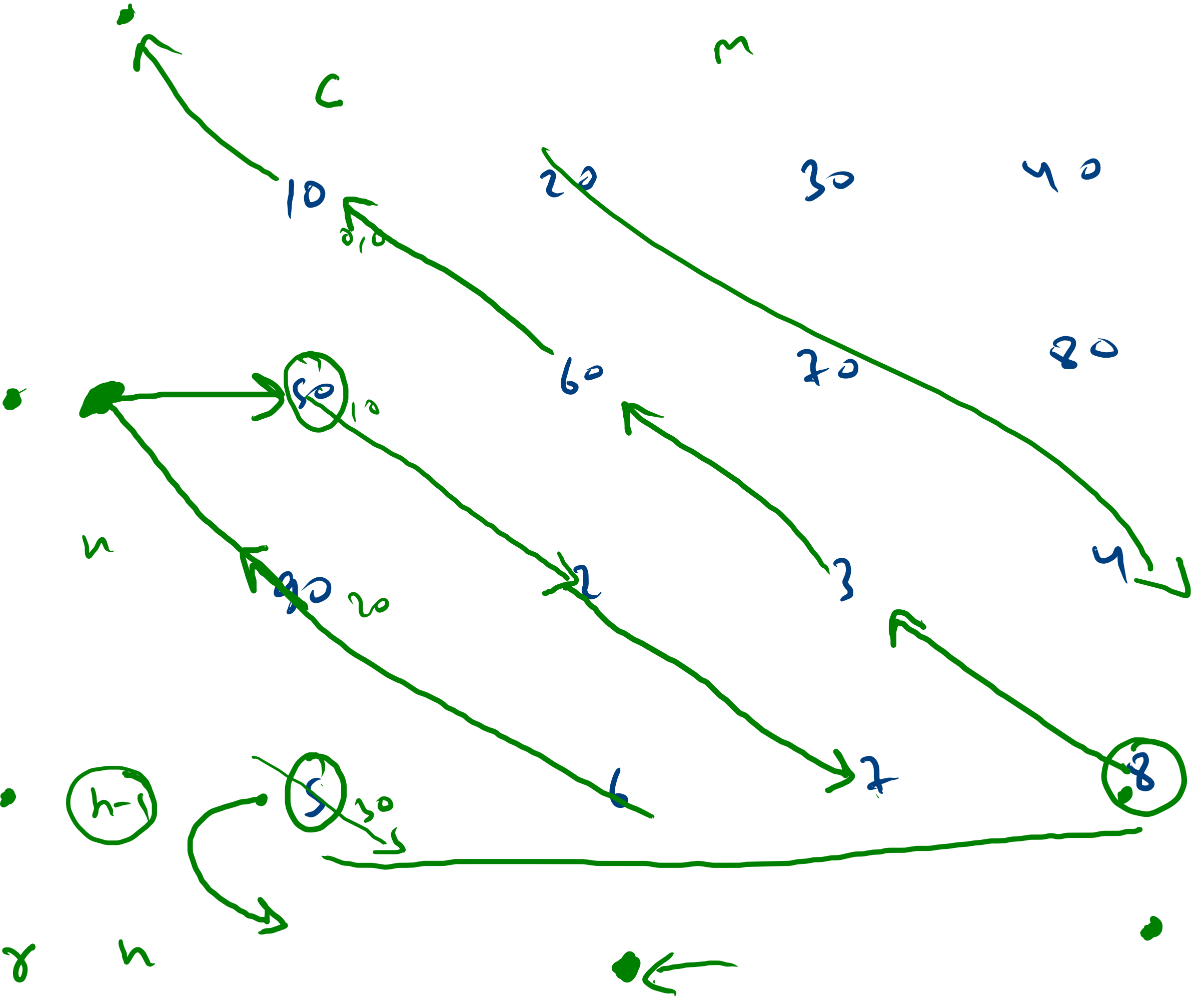
for 0 to size -1

1





prime	remove
not prime	i++



i++, j++

i--, j--

r = n-1

c = 0

```
while( r<n && ??   ){
    print( ar[r][c] );
    r++;
    c++;
}
r--
while(   ??   ){
    print (ar[r][c]);
    r--;
    c--;
}
c++;
```

2 condition

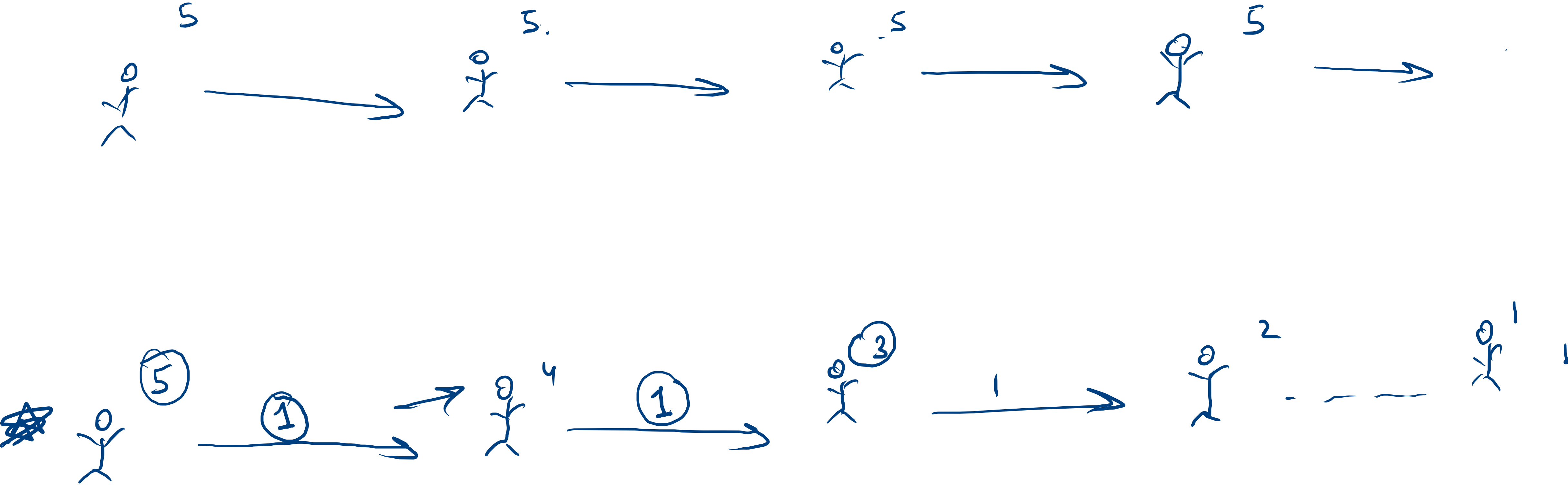
2 condition

easy

Iteration $5! \rightarrow 1 \times 2 \times 3 \times 4 \times 5$

Recursion $5! \rightarrow 5 \times 4!$

$sum(5) \rightarrow 5 + sum(4)$


5
5.
5
5
5
1
4
1
3
1
2
1

Recursion

$5! \rightarrow 5 \times 4!$

$\text{sum}(5) \rightarrow 5 + \text{sum}(4)$

Expectation

factorial 5

faith

4!

n = 5

Expectation
n = 5

n	$\boxed{5}$
n-1	4
n-2	3
⋮	2
1	1

faith
n-1 4

4
3
2
1

n = 3

3
2
1

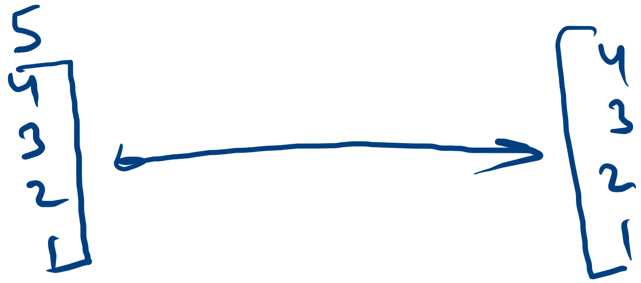
self work

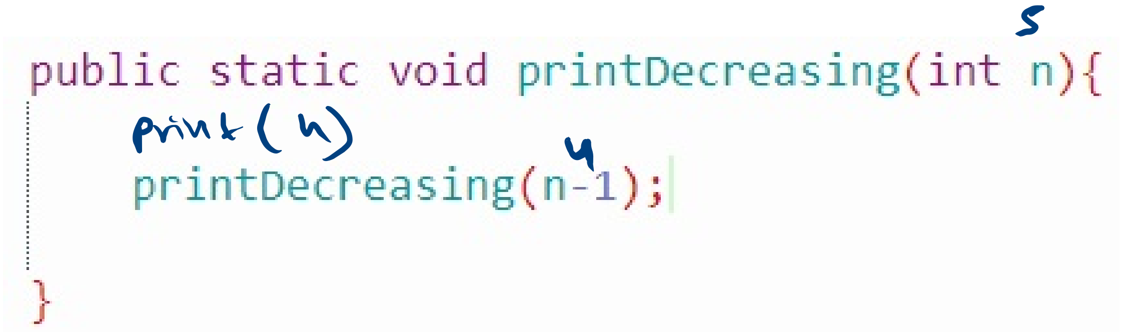
faith

n = 4

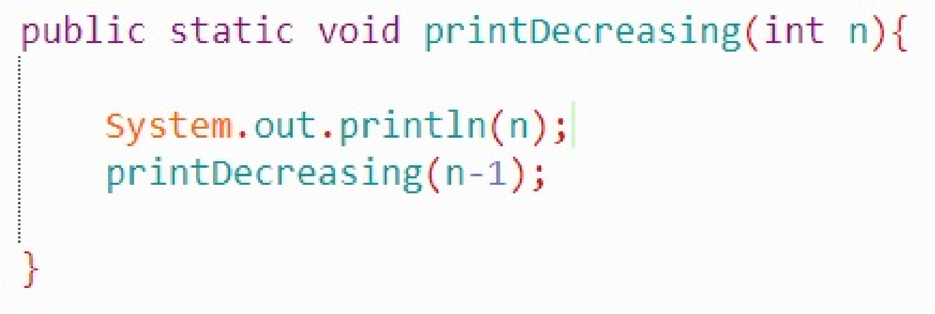
4
3
2
1

PD (5)

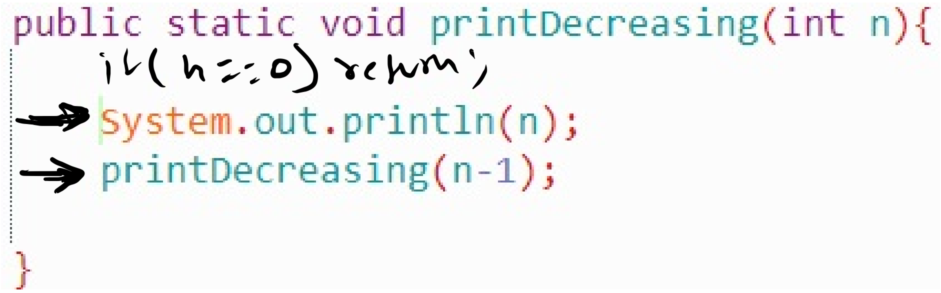




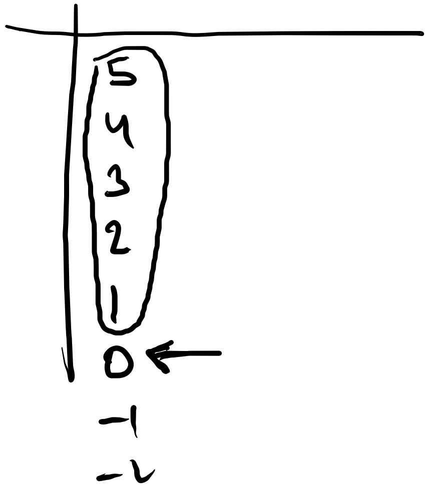
```
                                                   5
public static void printDecreasing(int n){
     print ( n)
                            4
     printDecreasing(n-1);

}
```

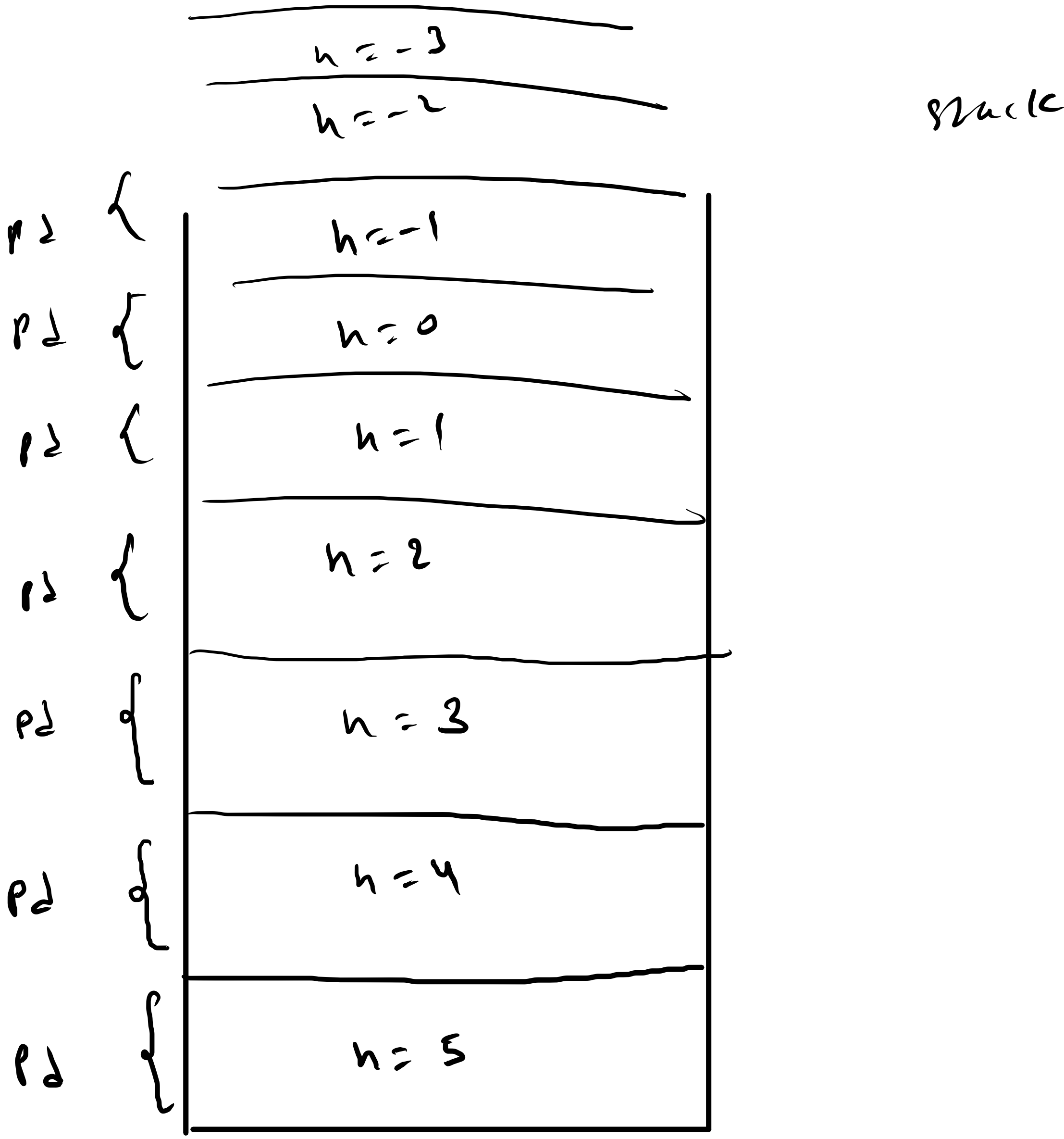


```
public static void printDecreasing(int n){

    System.out.println(n);
    printDecreasing(n-1);

}
```

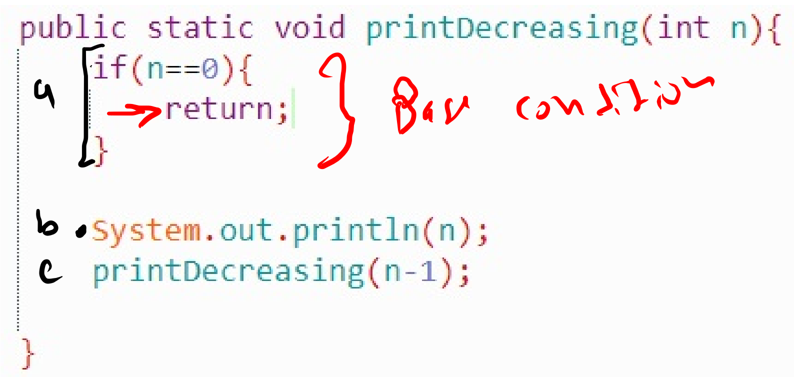


```
public static void printDecreasing(int n){
    if(n==0) return;
    System.out.println(n);
    printDecreasing(n-1);

}
```



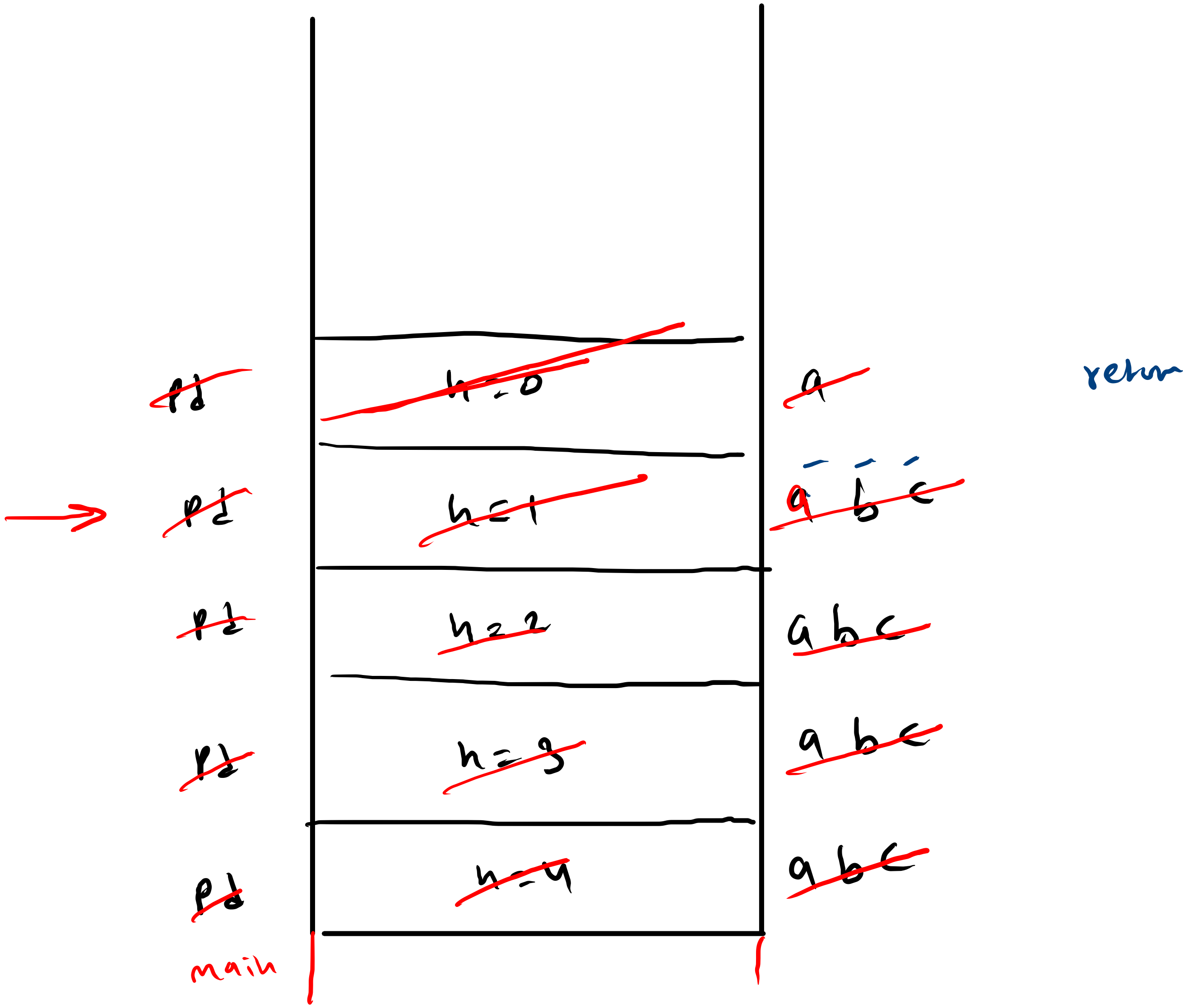


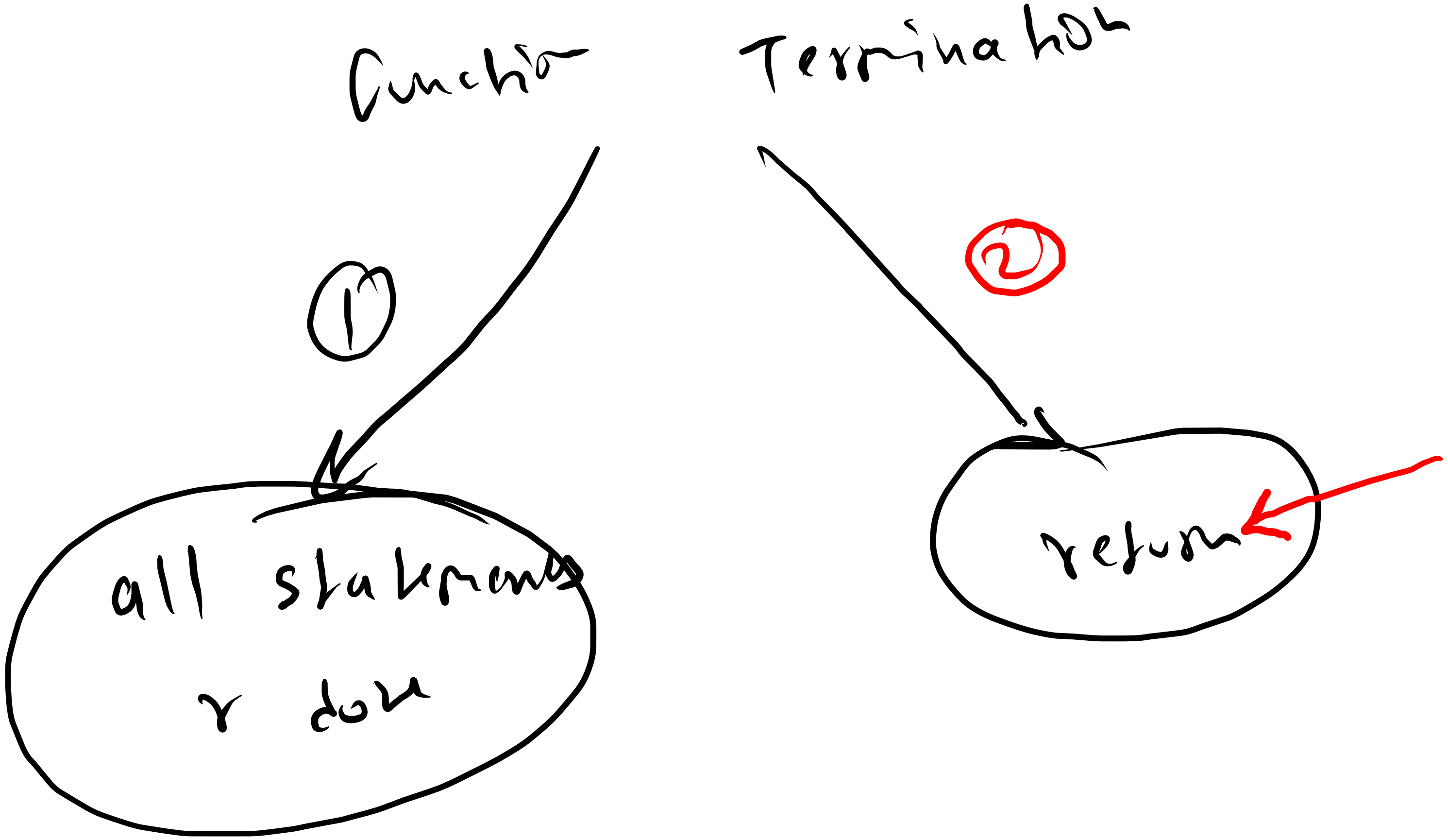
4



```
public static void printDecreasing(int n){
    if(n==0){
        return;
    }

    System.out.println(n);
    printDecreasing(n-1);

}
```

4
3
2
1




Function Termination
1
all statements r done
2
return

1
⋮
5

n = 5

5

Expectation

1
2
3
4 faith

5

4

faith

1
2
3
4

PI(4)

print(5)

n

expectation

1
2
3
⋮
n-1 ✓
n

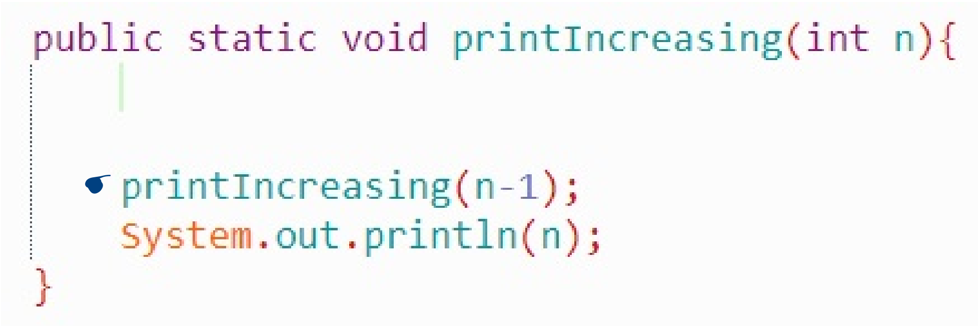
faith

1
2
3
⋮
n-1
n

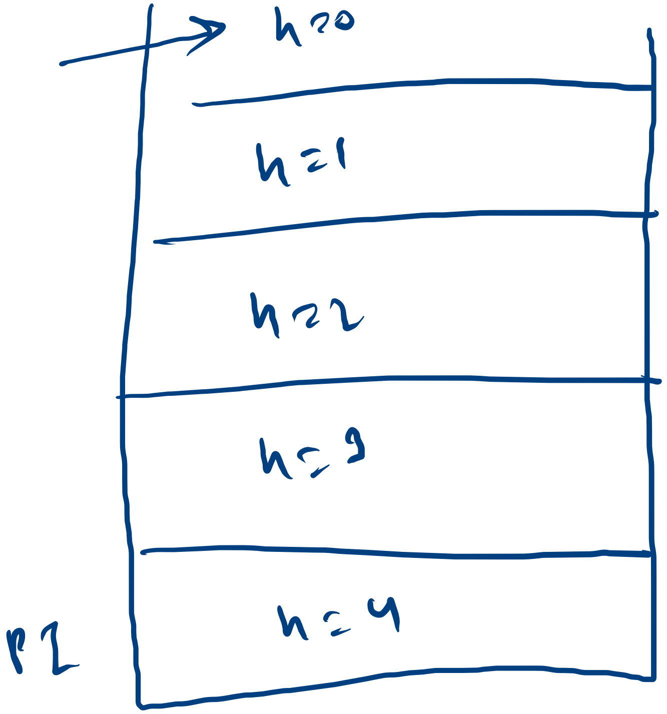
PI(n-1)

print(n)

4

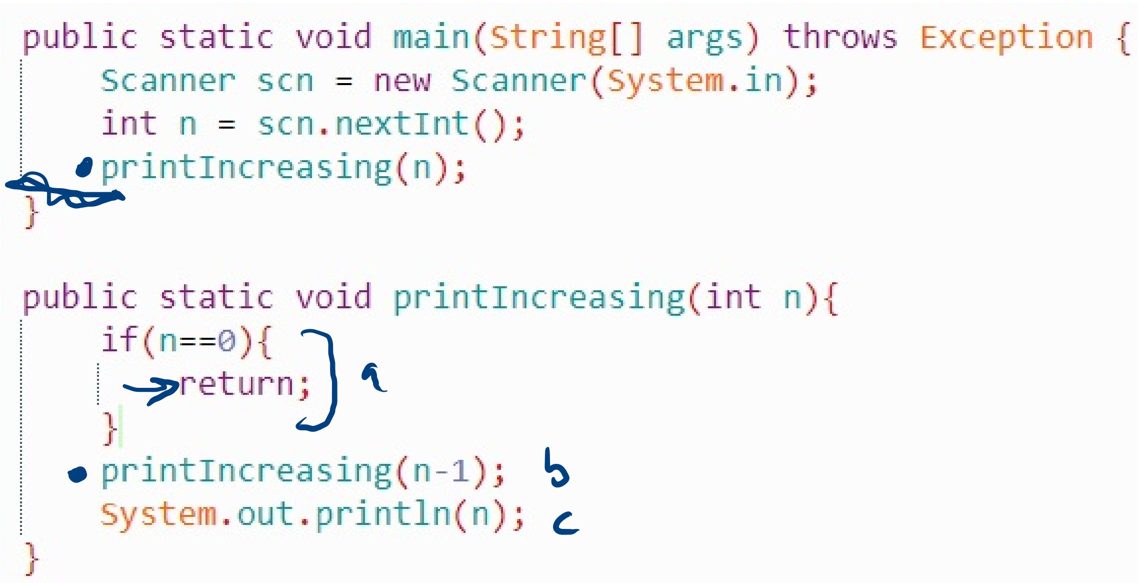


```
public static void printIncreasing(int n){


    printIncreasing(n-1);
    System.out.println(n);
}
```

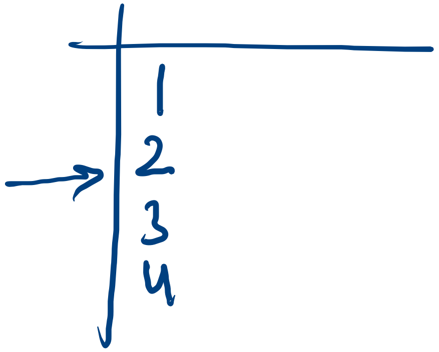


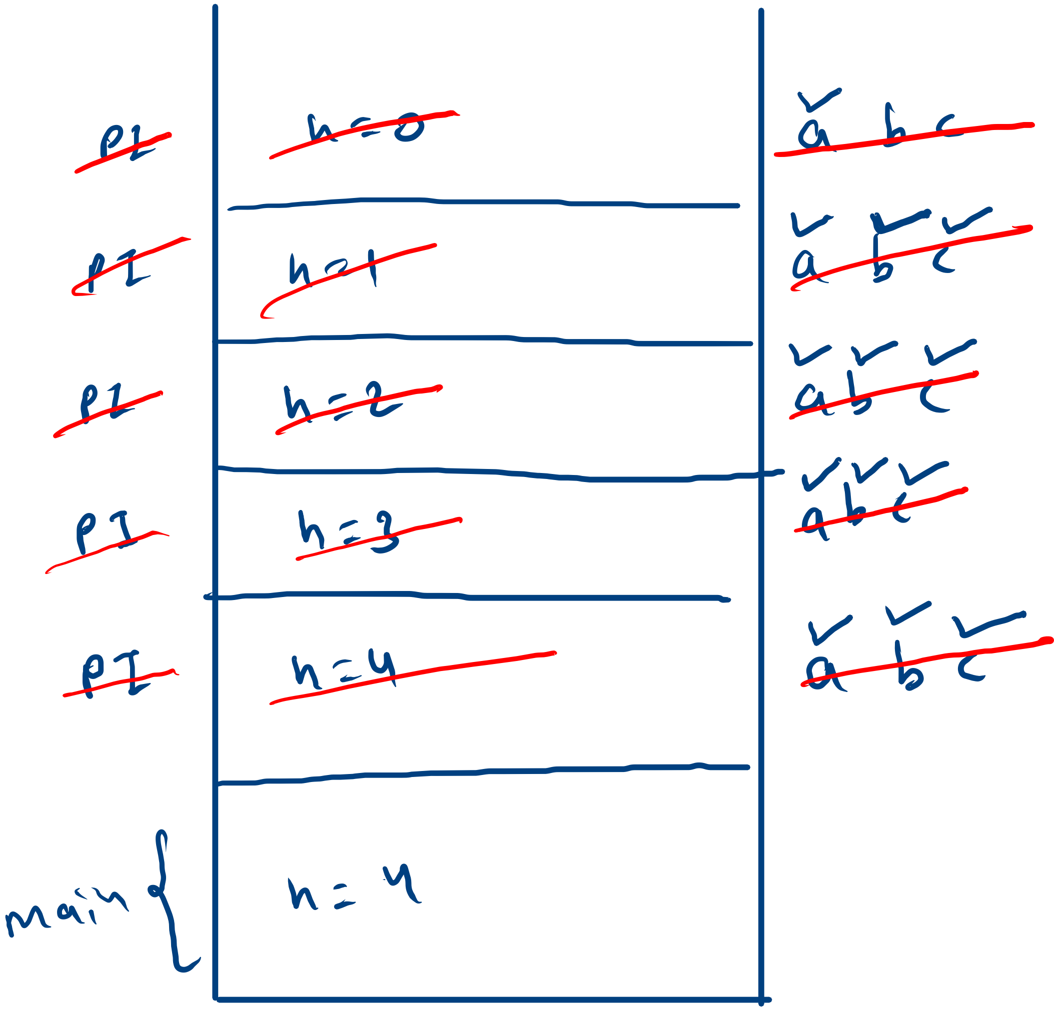
1
:
n

n = 4



```
public static void main(String[] args) throws Exception {
    Scanner scn = new Scanner(System.in);
    int n = scn.nextInt();
    printIncreasing(n);
}

public static void printIncreasing(int n){
    if(n==0){
        return;     a
    }
    printIncreasing(n-1);  b
    System.out.println(n);  c
}
```




~~PI~~
~~n=0~~
~~a b c~~
~~PI~~
~~n=1~~
~~a b c~~
~~PI~~
~~n=2~~
~~a b c~~
~~PI~~
~~n=3~~
~~a b c~~
~~PI~~
~~n=4~~
~~a b c~~
main
n = 4