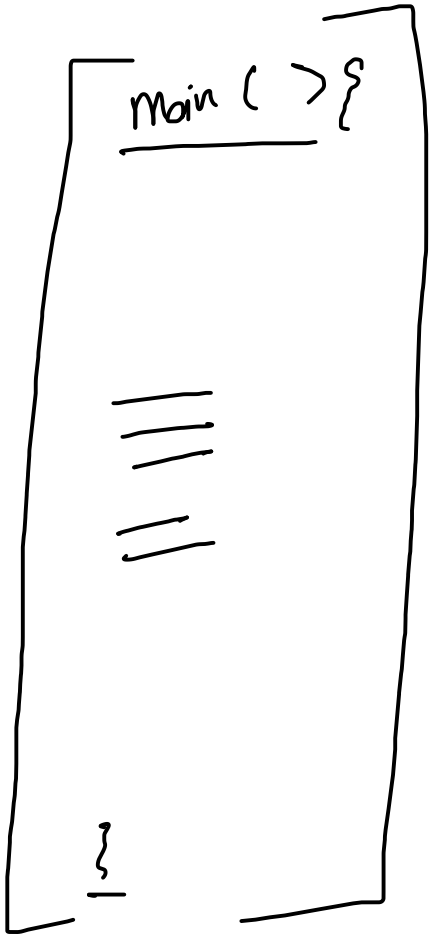
Modularity →



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

n, r

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

$5! = 5 \times 4 \times 3 \times 2 \times 1$

$n = 5$

$i = 1, 2, 3, 4, 5,$

nfact $= 1 \times 1 = 1 \times 2$

$= 2 \times 3$

$= 6 \times 4$

$= 24 \times 5$

$= 120$

```
int nfact = 1;

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

i <= 5

Java

fact(n, r)

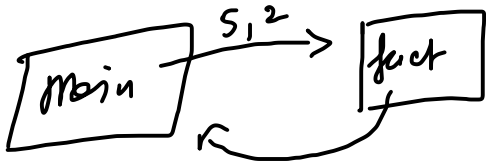
Program stack

Heap

```java
import java.util.*;

public class Main{

public static void main(String[] args) {

    Scanner scn = new Scanner(System.in);
    int n = scn.nextInt();
    int r = scn.nextInt();

    int nfact = 1;

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

    int rfact = 1;

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

    int nMrfact = 1;

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

    System.out.println(nfact/(rfact*nMrfact));

 }
}
```

main()

nMrf	6
rfact	2
nfact	120 ~~12~~ ~~6~~ ~~24~~
r	2
n	5



type of value function is expected to return

returntype

functionNam

Parameter list

```
public static int fact(int n,int r){

}
```

```java
public static int fact(int n){
    int fact = 1;
    for(int i = 1 ; i <= n ; i++){
        fact = fact * i;
    }

    return fact;
}

public static int nCr(int n,int r){
    int nfact = fact(n);
    int rfact = fact(r);
    int nMrfact = fact(n-r);

    return (nfact/(rfact*nMrfact));
}
public static void main(String[] args) {

    Scanner scn = new Scanner(System.in);
    int n = scn.nextInt();
    int r = scn.nextInt();

    int val = nCr(n,r);

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

120 2 * 6

10

main

^nC_r

fact fact fact

Heap

return

nothing to process

Sample Input

994543234

4

Sample Output

3

10 | 994543234 %

10 | 99454323 4

10 | 9945432 3

994543 2

```java
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    int n = scn.nextInt();
    int d = scn.nextInt();
    int f = getDigitFrequency(n, d);
    System.out.println(f);
}

public static int getDigitFrequency(int n, int d) {
    int count = 0 ;
    while(n != 0){
        int digit = n % 10;

        if(digit == d){
            count++;
        }
        n = n/10;
    }

    return count;
}
```

// 12131

// 1

3

main

f	3
d	1
n	12131

3

Arrays → Continuous memory allocation of $\boxed{\text{same type}}$ ✓

Function
- → Code efficient
- → reuse ✓
- → readable
- → Debugging

int arr[] = new int[5]; → length of the array

Indexing → Block access

(0 – len-1)

0	1	2	3	4
~~0~~ 5	~~0~~ 10	0	0	0

arr[0] = 5;

arr[1] = 10

```java
import java.util.*;

public class Main{

public static void main(String[] args) {

    // declare
    int arr[];

    // size allocate
    arr = new int[5];

    // int arr1[] = new int[5]; //  declare + size allocate

    // assign
    arr[0] = 10;
    arr[1] = 5;
    arr[3] = 100;
    arr[4] = 25;

    // print - for
    for(int idx = 0 ; idx <= 4 ; idx++){
        System.out.println(idx+" --> "+arr[idx]);
    }
    // print - for each
    for(int val : arr){
        System.out.print(val+" ");
    }

    System.out.println();
    // length
    System.out.println(arr.length);
}
```

new → Heap

0 --> 10
1 --> 5
2 --> 0
3 --> 100
4 --> 25

10 5 0 100 25

5

int i;
int idx
int val

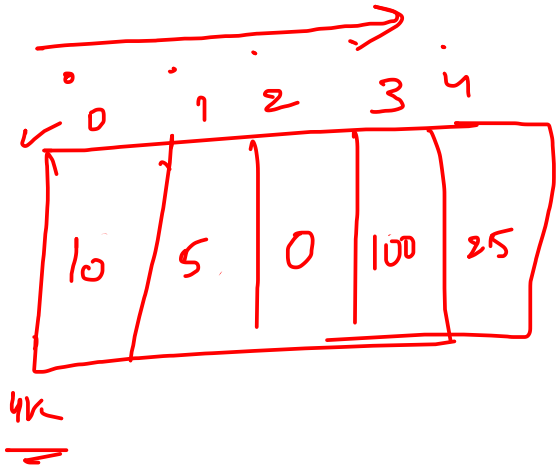
length

main

Program stack

char [] arr = new char[] Heap

String[] arr = new String[];



6 15 30 40 4 11 9

easy

0	1	2	3	4	5
15	30	40	4	11	9

Span = 40 - 4
= 36

val + 0 = val

val * 1 = val

Min(val, +∞) = val

Max(val, -∞) = val

Max = ~~-∞~~
~~-15~~
~~30~~
40

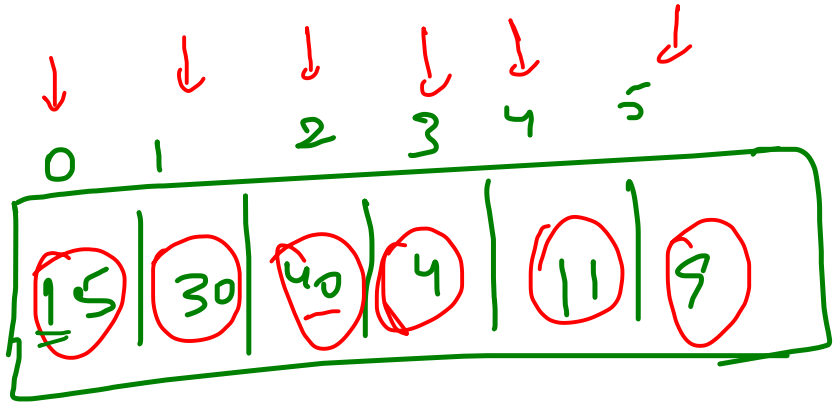
Min = ~~+∞~~
~~15~~
4

addition → identity = 0 ; Sum, Count

multi → identity = 1 ; nfact

min → identity = +∞, minimum = Integer.MAX_VALUE;

max → identity = -∞, maximum = Integer.MIN_VALUE;



max = -∞ 15 30 40 40 40 40

min = +∞ 15 15 15 4 4 4

36

```java
public static int span(int arr[]){
    // span of the array

    int max = Integer.MIN_VALUE, min = Integer.MAX_VALUE;

//    for(int idx = 0 ; idx < arr.length ; idx++){
//        if(arr[idx] > max){
//            max = arr[idx];
//        }

//        if(arr[idx] < min){
//            min = arr[idx];
//        }
//    }

    for(int val : arr){
        max = Math.max(max,val);
        min = Math.min(min,val);
    }

    return max - min;
}
```

Search

Linear search

Binary search

idx → -1

ele = 400

15	30	40	4	11	9
0	1	2	3	4	5

6

idx = -1

arr.length;

Integer.MAX_VALUE;

Integer.MIN_VALUE;

6 15 30 40 4 11 9 40

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

public class Main {

    public static void main(String[] args){

        Scanner scn = new Scanner(System.in);

        int n = scn.nextInt();
        int arr[] = new int[n];
        for(int i = 0 ;  i < n ; i++){
            arr[i] = scn.nextInt();
        }
        int ele = scn.nextInt();

        System.out.println(find(arr,ele));
    }

    public static int find(int arr[],int ele){
        // code
    }

}
```

```java
public static void main(String[] args){

    Scanner scn = new Scanner(System.in);

    int n = scn.nextInt();
    int arr[] = new int[n];
    for(int i = 0 ;  i < n ; i++){
        arr[i] = scn.nextInt();
    }
    int ele = scn.nextInt();

    System.out.println(find(arr,ele));
}

public static int find(int arr[],int ele){
    // code
    for(int idx = 0 ; idx < arr.length ; idx++){
        if(arr[idx] == ele){
            return idx;
        }
    }

    // implication
    return -1;
}
```

Element doesn't exist

2 | -1

n=6

0	1	2	3	4	5
15	30	40	4	11	9

ele = 400

idx = ~~0~~ 2

6

15
30
40
4 11
9

400

Input

0	1	2	3	4
1	2	3	4	5

Output

0	1	2	3	4
5	4	3	2	1

5 1 2 3 4 5

arr =

0	1	2	3	4	5	6	7	8	9	10	11
a	b	c	d	e	g f	f g	h	i	j	k	l

hi

lo

$l_o < h_i$

0	1	2	3	4	5	6	7	8
10	20	30	40	50	60	70	80	90
90	80	70	60		40	30	20	10

lo hi

```java
public class Main {
    public static void display(int[] a) {
        for(int val : a){
            System.out.print(val+" ");
        }
        System.out.println();
    }

    public static void reverse(int[] arr) {
        int lo = 0 , hi = arr.length-1;

        while(lo < hi){
            int temp = arr[lo];
            arr[lo] = arr[hi];
            arr[hi] = temp;

            lo++;
            hi--;
        }
    }

    public static void main(String[] args) throws Exception {
        Scanner scn = new Scanner(System.in);

        int n = scn.nextInt();
        int[] a = new int[n];
        for (int i = 0; i < n; i++) {
            a[i] = scn.nextInt();
        }

        reverse(a);
        display(a);
    }
```

function
- return
- nothing to do

Heap → charge → main, fun1, fun2

reverse(4K)

temp	30
hi	~~6~~ ~~5~~ ~~4~~ ~~3~~
lo	~~0~~ ~~1~~ ~~2~~ 3
arr	4K

main

a	4K
n	7

hi
lo

0	1	2	3	4	5	6
~~10~~ ~~70~~	~~20~~ 60	~~30~~ 50	40	~~50~~ 30	~~60~~ 20	~~70~~ 10

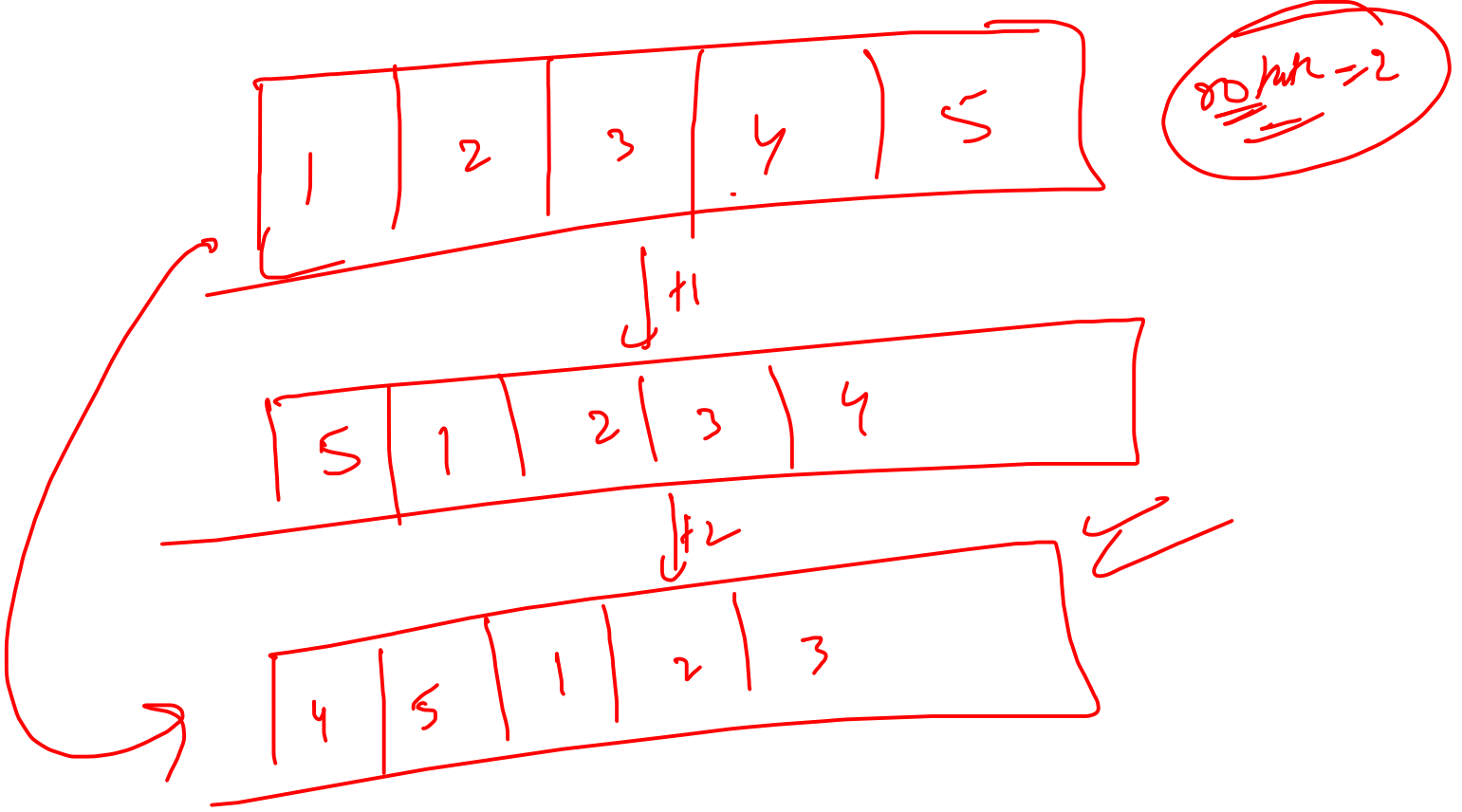
4K

70 60 50 40 30 20 10

1. n/r → fact → Practice
2. Functions
3. Memory management
4. Arrays [declare, define/memory, traverse (for/while/foreach), length]
5. span [max, min] → Indentities
6. find (arr, el) → linear search
7. reverse of an array
8. rotate an array

H.W.

H.W.



so pair = 2

StringBuilder → Interning → Mutable

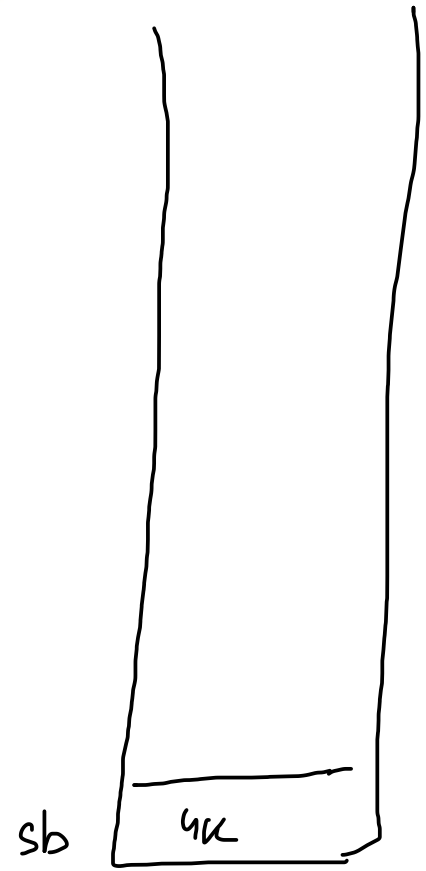
→ "hello"

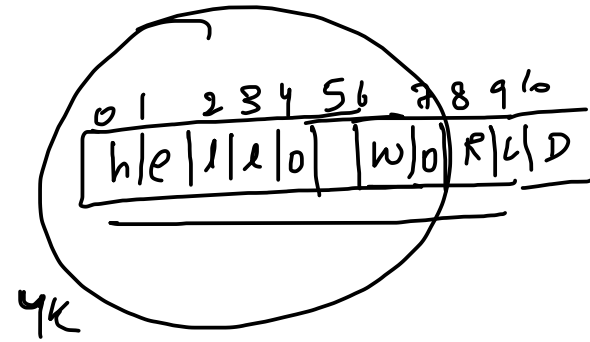
h	e	l	l	o

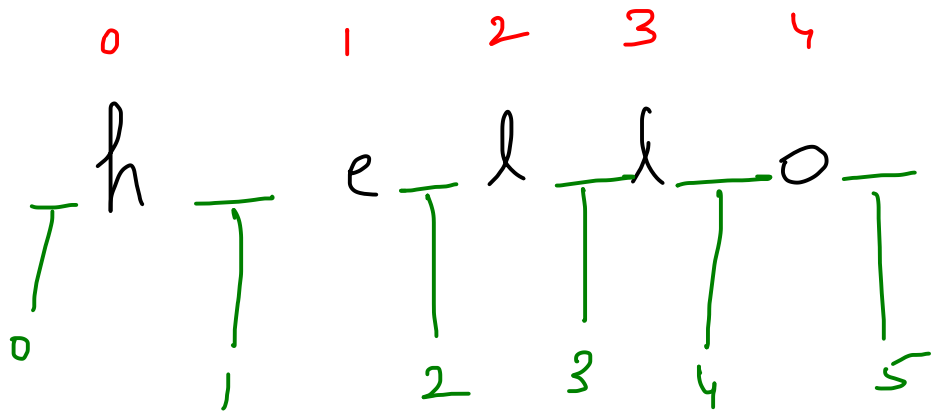
advantages → multiple operations → SB

```
StringBuilder sb = new StringBuilder("hello");

System.out.println(sb);

System.out.println(sb.length());

System.out.println(sb.charAt(0));

sb.append(" world");
System.out.println(sb);
```

hello
5
h







idx, "ch"

(0, len)

insert(idx, ch)

charAt(idx)

(0, len-1)

```java
import java.util.*;

public class Main{

public static void main(String[] args) {

    StringBuilder sb = new StringBuilder("hello");

    System.out.println(sb);

    System.out.println(sb.length());

    System.out.println(sb.charAt(0));

    // sb.append(" world");
    // System.out.println(sb);

    sb.insert(1,'z');

    System.out.println(sb);

    System.out.println(sb.reverse());

    sb.deleteCharAt(0);
    System.out.println(sb);
  }
}
```

hello
5
h
hzello
lleezh

length()
charAt()
substring()
insert()
reverse()
deleteCharAt()
append()

sb

str

4K

0	1	2	3	4	5
h	l	e	z	l	l

4K

hello

hello world

Cases toggle

Lowercase → Uppercase

Input

pepCODinG

Output

"PEPcodING"

operator +, >, ch → int (ASCII)

a, b, c, ..., z
(97, 98, 99, ..., 122)

A, B, C, ..., Z
(65, 66, 67, ..., 90)

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
32	20	[SPACE]	64	40	@	96	60	`
33	21	!	65	41	A	97	61	a
34	22	"	66	42	B	98	62	b
35	23	#	67	43	C	99	63	c
36	24	$	68	44	D	100	64	d
37	25	%	69	45	E	101	65	e
38	26	&	70	46	F	102	66	f
39	27	'	71	47	G	103	67	g
40	28	(	72	48	H	104	68	h
41	29	)	73	49	I	105	69	i
42	2A	*	74	4A	J	106	6A	j
43	2B	+	75	4B	K	107	6B	k ch
44	2C	,	76	4C	L	108	6C	l
45	2D	-	77	4D	M	109	6D	m
46	2E	.	78	4E	N	110	6E	n
47	2F	/	79	4F	O	111	6F	o
48	30	0	80	50	P	112	70	p
49	31	1	81	51	Q	113	71	q
50	32	2	82	52	R	114	72	r
51	33	3	83	53	S	115	73	s
52	34	4	84	54	T	116	74	t
53	35	5	85	55	U	117	75	u
54	36	6	86	56	V	118	76	v
55	37	7	87	57	W	119	77	w
56	38	8	88	58	X	120	78	x
57	39	9	89	59	Y	121	79	y
58	3A	:	90	5A	Z	122	7A	z
59	3B	;	91	5B	[	123	7B	{
60	3C	<	92	5C	\	124	7C	|
61	3D	=	93	5D	]	125	7D	}
62	3E	>	94	5E	^	126	7E	~
63	3F	?	95	5F	_	127	7F	[DEL]

① ch ∈ Uppercase / lowercase

ch = 'm' >

if (ch >= 'a' && ch <= 'z') {
  ch ∈ Lowercase
}

if (ch >= 'A' && ch <= 'Z') {
  ch ∈ Uppercase
}

② ✓ $\boxed{\underbrace{'M' - 'A'}_{UC} = \underbrace{'h' - 'a'}_{LC}}$

• ch ∈ Uppercase

ch − 'A' = x − 'a'

$\boxed{x = (ch - 'A') + 'a'}$

Ascii of lowercase (char)

ch ∈ Lowercase

x − 'A' = ch − 'a'

$\boxed{x = (ch - 'a') + 'A'}$

Ascii of Uppercase

p e p C O D i n G

0 1 2 3 4 5 6 7 8

p	-11	e	11	p	-45	C	12	O

ch1 Diff ch2

e - 101

p → 112

C = 67

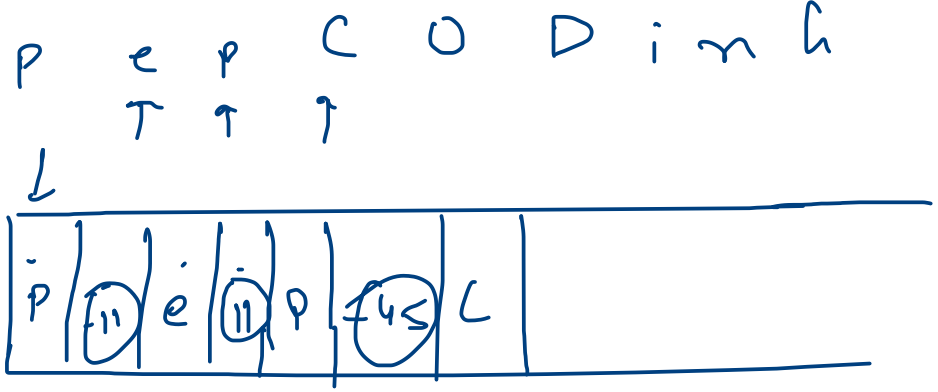
O → 79

pepCODinG

Sample Output

p-11e11p-45C12O-11D37i5n-39G

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
32	20	[SPACE]	64	40	@	96	60	`
33	21	!	65	41	A	97	61	a
34	22	"	66	42	B	98	62	b
35	23	#	67	43	C	99	63	c
36	24	$	68	44	D	100	64	d
37	25	%	69	45	E	101	65	e
38	26	&	70	46	F	102	66	f
39	27	'	71	47	G	103	67	g
40	28	(	72	48	H	104	68	h
41	29	)	73	49	I	105	69	i
42	2A	*	74	4A	J	106	6A	j
43	2B	+	75	4B	K	107	6B	k
44	2C	,	76	4C	L	108	6C	l
45	2D	-	77	4D	M	109	6D	m
46	2E	.	78	4E	N	110	6E	n
47	2F	/	79	4F	O	111	6F	o
48	30	0	80	50	P	112	70	p
49	31	1	81	51	Q	113	71	q
50	32	2	82	52	R	114	72	r
51	33	3	83	53	S	115	73	s
52	34	4	84	54	T	116	74	t
53	35	5	85	55	U	117	75	u
54	36	6	86	56	V	118	76	v
55	37	7	87	57	W	119	77	w
56	38	8	88	58	X	120	78	x
57	39	9	89	59	Y	121	79	y
58	3A	:	90	5A	Z	122	7A	z
59	3B	;	91	5B	[	123	7B	{
60	3C	<	92	5C	\	124	7C	|
61	3D	=	93	5D	]	125	7D	}
62	3E	>	94	5E	^	126	7E	~
63	3F	?	95	5F	_	127	7F	[DEL]




```java
public static String solution(String str) {

    StringBuilder sb = new StringBuilder();
    sb.append(str.charAt(0));

    for(int idx = 1 ; idx < str.length() ; idx++){
        char curr = str.charAt(idx);
        char prev = str.charAt(idx-1);

        sb.append(curr-prev);
        sb.append(curr);
    }

    return sb.toString();
}
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    String str = scn.next();
    System.out.println(solution(str));
}
```

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
32	20	[SPACE]	64	40	@	96	60	`
33	21	!	65	41	A	97	61	a
34	22	"	66	42	B	98	62	b
35	23	#	67	43	C	99	63	c
36	24	$	68	44	D	100	64	d
37	25	%	69	45	E	101	65	e
38	26	&	70	46	F	102	66	f
39	27	'	71	47	G	103	67	g
40	28	(	72	48	H	104	68	h
41	29	)	73	49	I	105	69	i
42	2A	*	74	4A	J	106	6A	j
43	2B	+	75	4B	K	107	6B	k
44	2C	,	76	4C	L	108	6C	l
45	2D	-	77	4D	M	109	6D	m
46	2E	.	78	4E	N	110	6E	n
47	2F	/	79	4F	O	111	6F	o
48	30	0	80	50	P	112	70	p
49	31	1	81	51	Q	113	71	q
50	32	2	82	52	R	114	72	r
51	33	3	83	53	S	115	73	s
52	34	4	84	54	T	116	74	t
53	35	5	85	55	U	117	75	u
54	36	6	86	56	V	118	76	v
55	37	7	87	57	W	119	77	w
56	38	8	88	58	X	120	78	x
57	39	9	89	59	Y	121	79	y
58	3A	:	90	5A	Z	122	7A	z
59	3B	;	91	5B	[	123	7B	{
60	3C	<	92	5C	\	124	7C	|
61	3D	=	93	5D	]	125	7D	}
62	3E	>	94	5E	^	126	7E	~
63	3F	?	95	5F	_	127	7F	[DEL]

Arrays → Continuous memory allocation

1. not enough continuous memory
2. fixed size

array

```
int arr[] = new int[5];
```

0	1	2	3	4

add 10
add 20
add 30
add 40
add 50

size() => 5

ArrayList => array + dynamic behaviour

0	1	2	3	4
10	20	30	40	50

Primitive Data types

int →

double →

float →

char →

boolean →

Wrapper Class

Integer

Double

Float

Character

Boolean

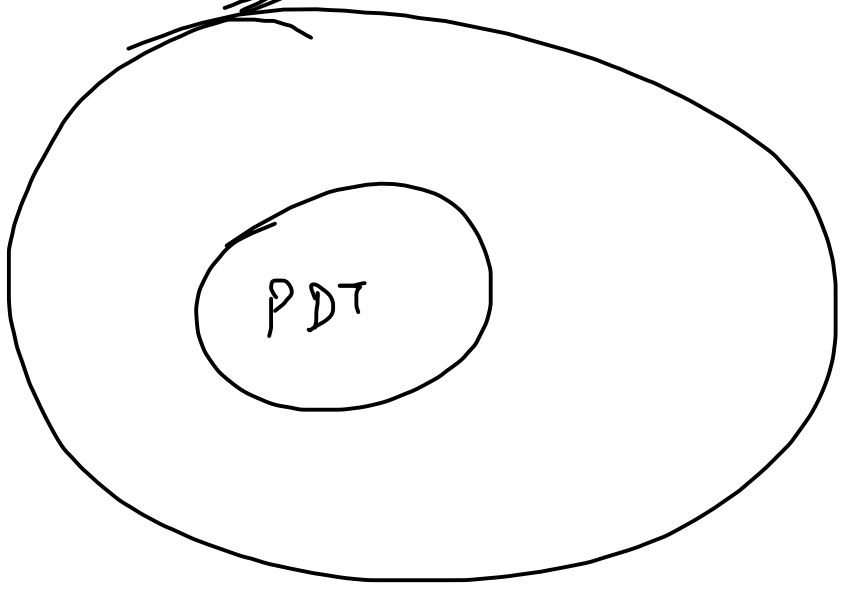
String

Wrapper Class

Integer.MAX_VALUE

Integer.MIN_VALUE

ArrayList<Integer>



① ArrayList<_> list = new ArrayList<>();

↳ Wrapper Class

[ ]

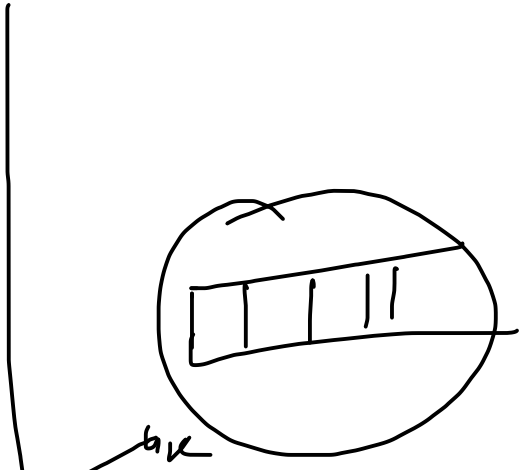
10	20	30	40	50
0	1	2	3	4

② functionalities

- ↳ size();
- ↳ idx => 0 - (size()-1)
- ↳ add(val) => adds val to the end.
- ↳ remove(idx) => removes idx val from arraylist
- ↳ get(idx) => returns val according to idx.
- ↳ set(idx, val) => update.
- ↳ add(idx, val) => insert

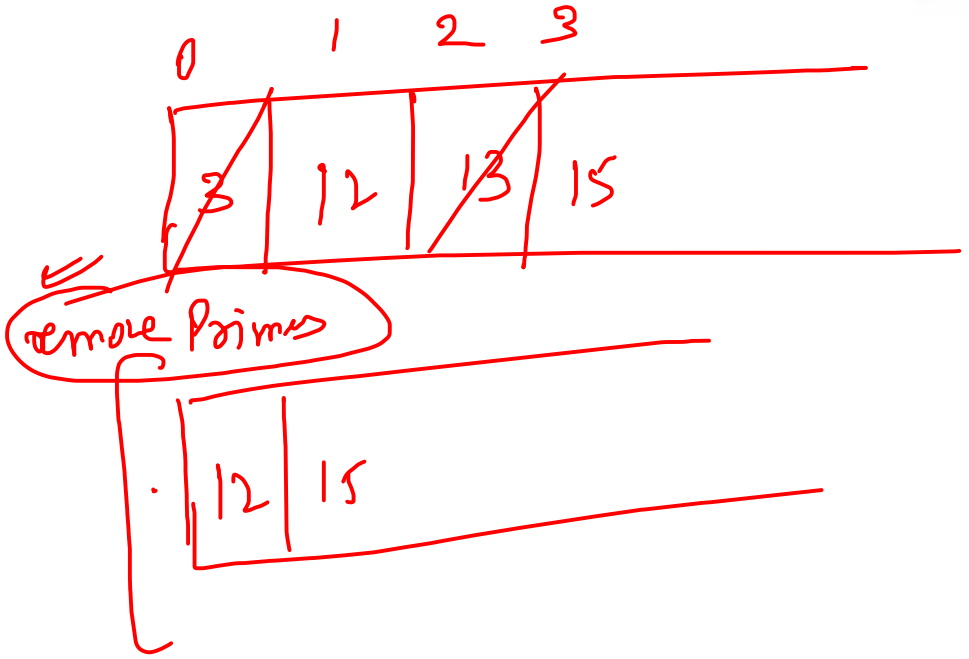
10	20	30
0	1	2

get(1) => 20



```
4
3 12 13 15
```

```java
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    int n = scn.nextInt();   //4
    ArrayList < Integer > al = new ArrayList < > ();
    for (int i = 0; i < n; i++) {
        al.add(scn.nextInt());
    }
    solution(al);
    System.out.println(al);
}
```



~~5~~	7	13	19	20	60
0	1	2	3	4	5

idx = ~~0~~ ~~1~~ ~~2~~ ~~3~~ 4

7	~~13~~	19	20	60
0	1	2	3	4

7	19	20	60
0	1	2	3

① idx update + shifting

skip

5	7	13	~~19~~	20	60
0	1	2	3	4	5

5	7	~~13~~	20	60
0	1	2	3	4

5	~~7~~	20	60
0	1	2	3

~~5~~	20	60
0	1	2

20	60
0	1

idx = ~~5~~ ~~4~~ ~~3~~ ~~2~~ ~~1~~ ~~0~~ -1

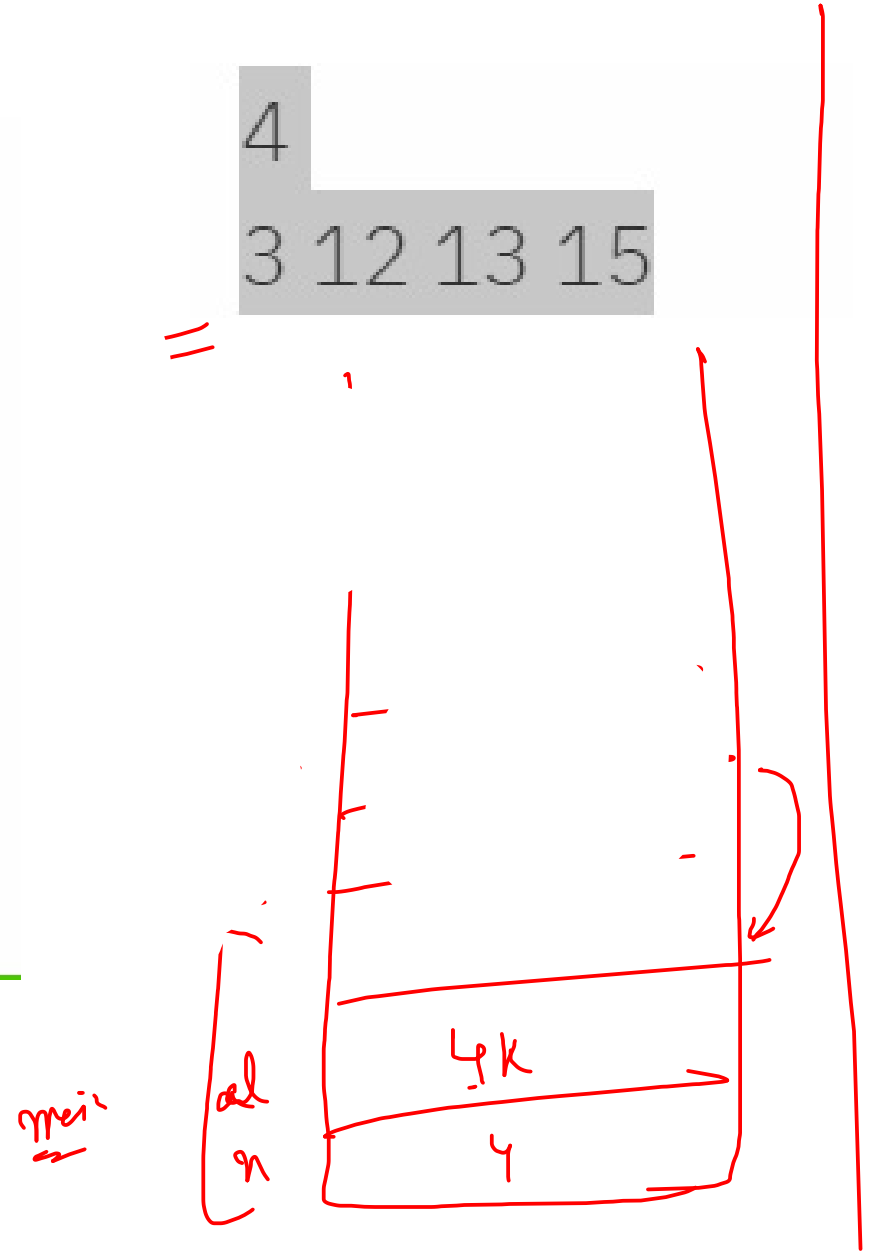
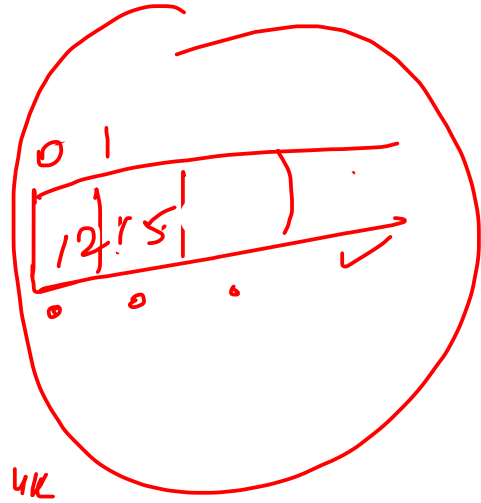
no skip

shifting

+

① loop

right to left

```java
public static boolean isPrime(int num){
    for(int fac = 2 ; fac * fac <= num ; fac++){
        if(num % fac == 0){
            // number is not a prime
            return false;
        }
    }

    return true;
}

public static void solution(ArrayList < Integer > al) {
    for(int idx = al.size()-1; idx >= 0 ; idx--){
        if(isPrime(al.get(idx))){
            al.remove(idx);
        }
    }
}
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    int n = scn.nextInt();
    ArrayList < Integer > al = new ArrayList < > ();
    for (int i = 0; i < n; i++) {
        al.add(scn.nextInt());
    }
    solution(al);
    System.out.println(al);
}
```

```
4
3 12 13 15
```