

DSA

Java

- ↳ Printing
- ↳ Conditionals
- ↳ Looping

} If you are just starting out

Printing

System.out.println("Let's gooo...");

Output:
Let's gooo...

Anything provided here in double quotes (" ") will be printed as it is.

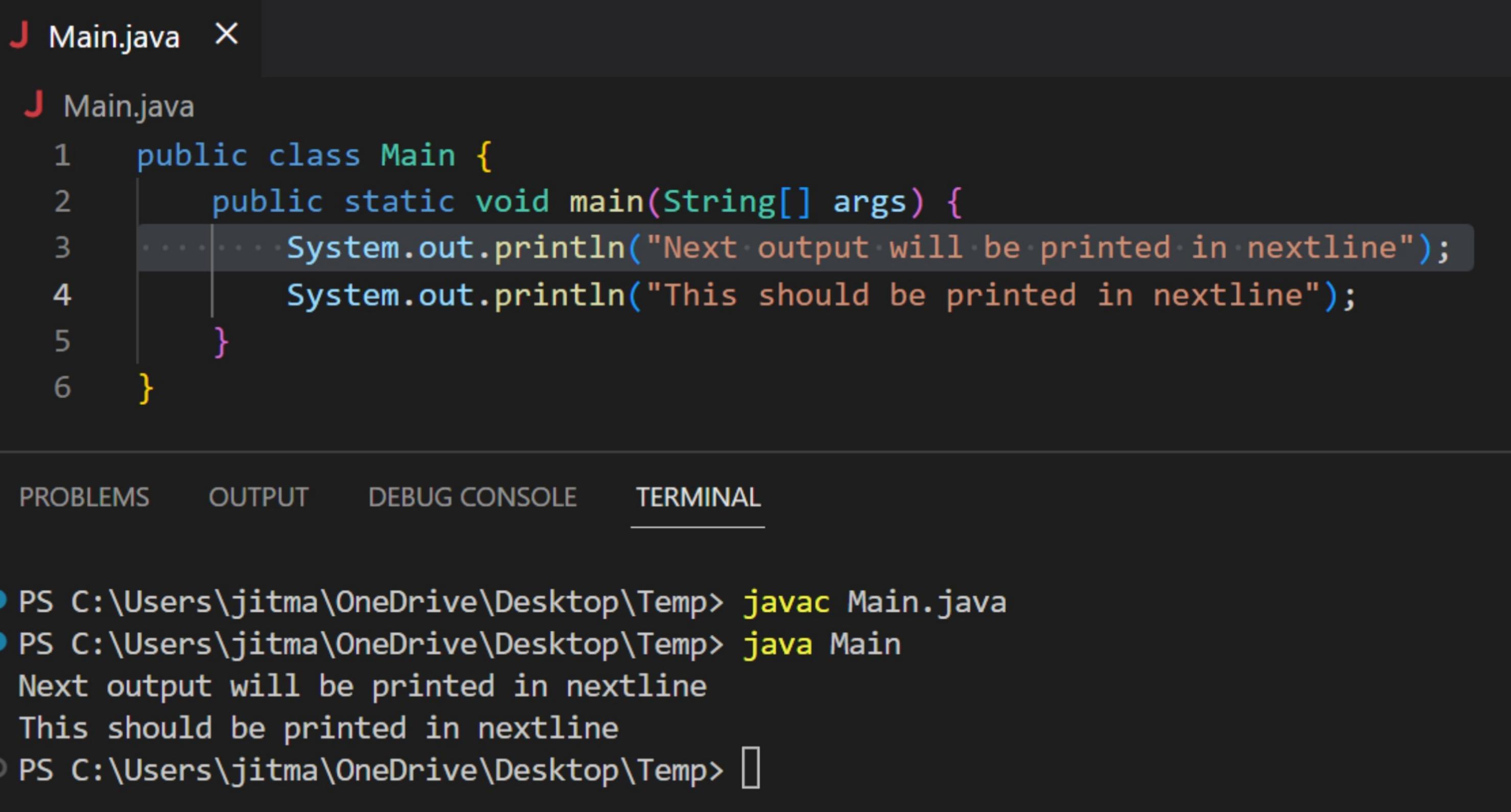
System.out.println ("One");
System.out.println ("2+3");
System.out.println ("5 < 6");

Output:
One
2+3
5 < 6

println vs print

`println` → Prints the output and leaves the cursor in the next line. Next output hence will be printed from the next line.

`print` → Will leave the cursor in the same line.



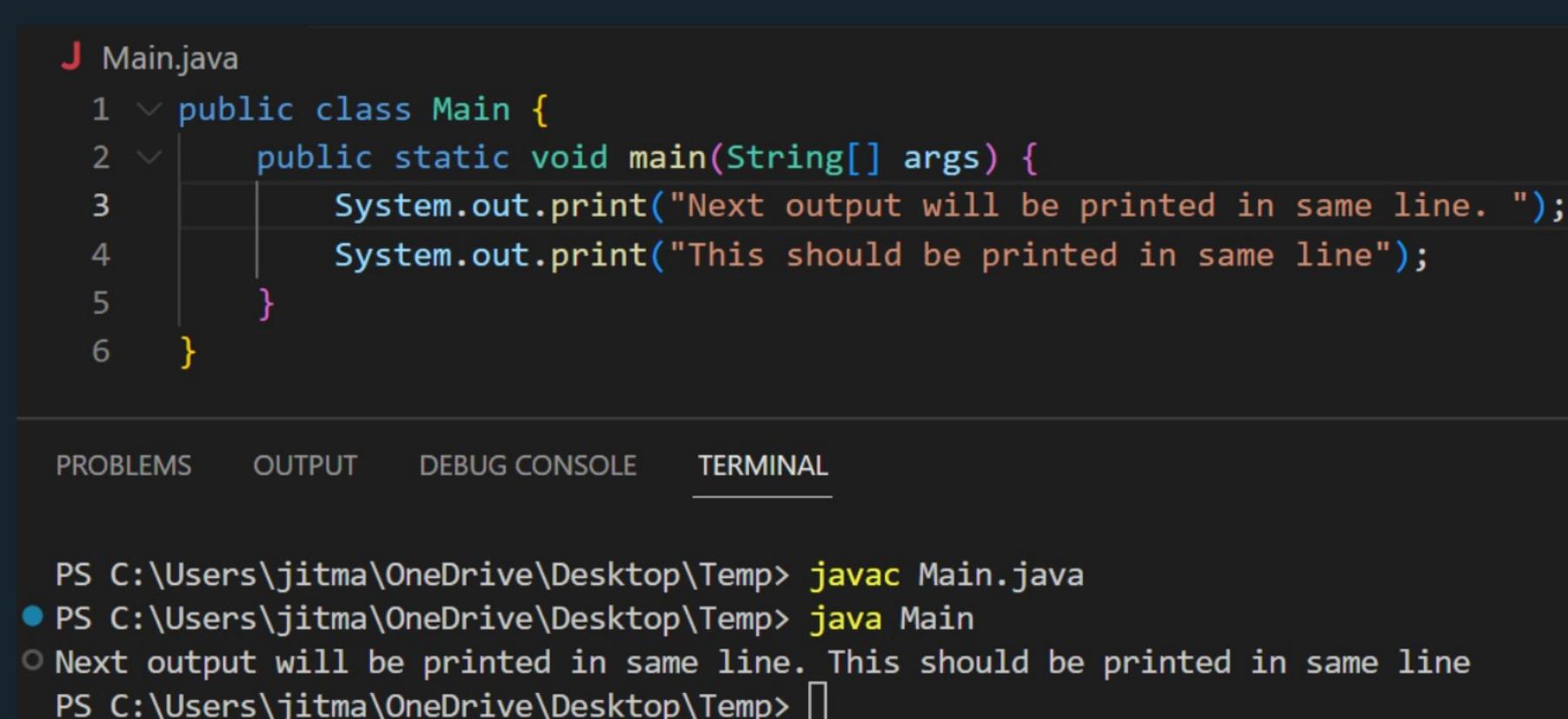
```

J Main.java X

J Main.java
1 public class Main {
2     public static void main(String[] args) {
3         System.out.println("Next output will be printed in newline");
4         System.out.println("This should be printed in newline");
5     }
6 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

- PS C:\Users\jitma\OneDrive\Desktop\Temp> `javac Main.java`
- PS C:\Users\jitma\OneDrive\Desktop\Temp> `java Main`
Next output will be printed in newline
This should be printed in newline
- PS C:\Users\jitma\OneDrive\Desktop\Temp> []



```

J Main.java
1 public class Main {
2     public static void main(String[] args) {
3         System.out.print("Next output will be printed in same line. ");
4         System.out.print("This should be printed in same line");
5     }
6 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

- PS C:\Users\jitma\OneDrive\Desktop\Temp> `javac Main.java`
- PS C:\Users\jitma\OneDrive\Desktop\Temp> `java Main`
Next output will be printed in same line. This should be printed in same line
- PS C:\Users\jitma\OneDrive\Desktop\Temp> []

Operators



Arithmetic Operators

Arithmetic operators are used to perform common mathematical operations.

Operator	Name	Description	Example
+	Addition	Adds together two values	$x + y$
-	Subtraction	Subtracts one value from another	$x - y$
*	Multiplication	Multiplies two values	$x * y$
/	Division	Divides one value by another	x / y
%	Modulus	Returns the division remainder	$x \% y$

$$\begin{aligned} 2 + 5 &= 7 \\ 8 - 3 &= 5 \\ 3 * 5 &= 15 \end{aligned} \quad \left. \begin{array}{l} \text{Same as common maths} \\ \text{part) } \end{array} \right\}$$

$5/3$ [integer division will result in loss of decimal part)
 $5/3 = 1$
Modulo operator \rightarrow remainder when a is devideed by b

Usage:

$a \% b \rightarrow$ returns remainder

$$10 \% 3 = 1$$

$$3 \% 3 = 0$$

$$5 \% 3 = 2$$

$$46 \% 100 = 46$$

$$452 \% 10 = 2$$

$$-28 \% 3 = -1$$

Comparison Operator



Operator	Name	Example
<code>==</code>	Equal to	<code>x == y</code>
<code>!=</code>	Not equal	<code>x != y</code>
<code>></code>	Greater than	<code>x > y</code>
<code><</code>	Less than	<code>x < y</code>
<code>>=</code>	Greater than or equal to	<code>x >= y</code>
<code><=</code>	Less than or equal to	<code>x <= y</code>

a operator b → Produces binary output, true or false based on condn.

J Main.java

```
1 public class Main {  
2     public static void main(String[] args) {  
3         System.out.println(3 < 4);  
4         System.out.println(5 <= 5);  
5         System.out.println((3 * 2) == 9);  
6         System.out.println(4 != (2 * 2));  
7         System.out.println(56 >= 12);  
8         System.out.println(45 != (45 % 100));  
9     }  
10 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

PS C:\Users\jitma\OneDrive\Desktop\Temp> javac Main.java
● PS C:\Users\jitma\OneDrive\Desktop\Temp> java Main
○ true
true
false
false
true
false
PS C:\Users\jitma\OneDrive\Desktop\Temp> □

Variables



Why can't we just hardcode values?

↳ VARIABLES ARE CONTROLLABLE

Data types

byte }
short } 1, 3, 45, -72, 42976
int
long }

char → 'a', '7',
'*', '!', '&'

boolean → true/false

float } 32.7 64.0
double } -9.8

↳ Different size (you don't have to memorize that
as of now)

Range is calculated using size of a particular datatype

$$2^{\text{bits-1}} - 1$$

Note: Concept of range will be covered again in
depth later

Variables

`int a = 5;` } Declaration and assignment
`int b = -4;` } -ment in the same line

`int c;` → Declaration

`c = 72;` → Assignment

`double x = 72.0;`

`x = 73.4;` ← value will be update

Main.java

```

1  public class Main {
2      public static void main(String[] args) {
3          int a = 6;
4          int b;
5          b = 32;
6          System.out.println(a);
7          System.out.println(b);
8      }
9 }
```

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

```

PS C:\Users\jitma\OneDrive\Desktop\Temp> javac Main.java
● PS C:\Users\jitma\OneDrive\Desktop\Temp> java Main
6
32
○ PS C:\Users\jitma\OneDrive\Desktop\Temp> 
```

Inputs

```
Scanner scn = new Scanner(System.in);
```

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

```
double d = scn.nextDouble();
```

↳ assignment operator

↳ right hand side will be resolved first

then it'll be saved in the variable on
the left hand side

J Main.java X

J Main.java

```
1 import java.util.*;
2 public class Main {
3     public static void main(String[] args) {
4         Scanner scn = new Scanner(System.in);
5         int a = scn.nextInt();
6         double b = scn.nextDouble();
7         System.out.println(a + b);
8     }
9 }
```

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

PS C:\Users\jitma\OneDrive\Desktop\Temp> javac Main.java

● PS C:\Users\jitma\OneDrive\Desktop\Temp> java Main

Conditionals



Conditionals

G

marks →

45
50
78
34
96
12
27
42
61
05

Score
 $\{ \text{Score} \geq 40 \rightarrow \text{passed}$
 failed otherwise



37

if (condition) {
 do this
} else {
 do something else
}

⇒

if (marks ≥ 40) {

} else {

}

if (condition) {

• // true → this will be exec

} else {

// condition resulted false

}

if (marks < 40) {

 System.out.println("fail");

} else {

 System.out.println("pass");

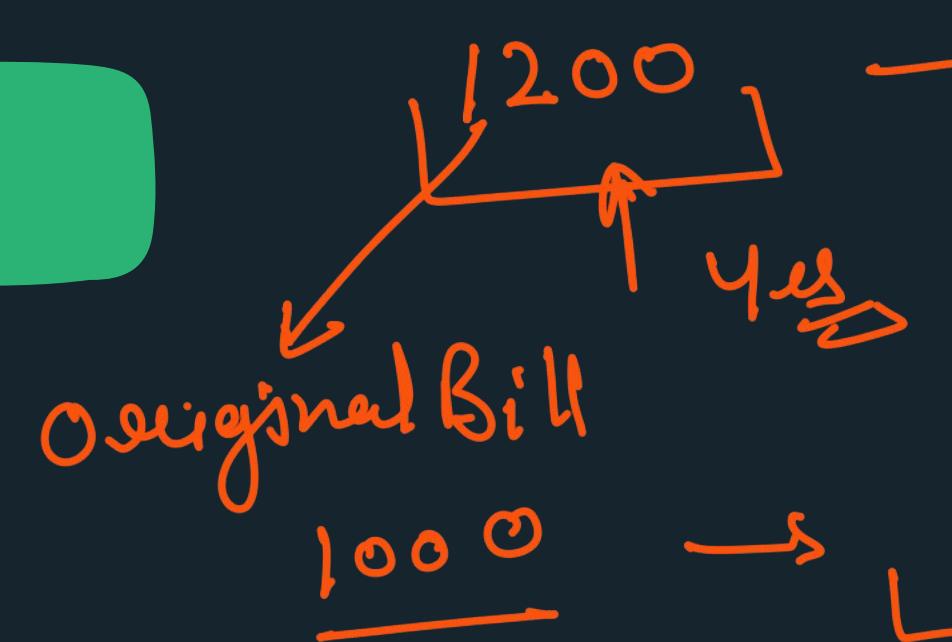
}

J Main.java X

```
1 import java.util.*;
2 public class Main {
3     public static void main(String[] args) {
4         Scanner scn = new Scanner(System.in);
5         int marks = scn.nextInt();
6         if(marks < 40) {
7             System.out.println("Fail");
8         } else {
9             System.out.println("Pass");
10        }
11    }
12 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
PS C:\Users\jitma\OneDrive\Desktop\Temp> java Main
68
Pass
● PS C:\Users\jitma\OneDrive\Desktop\Temp> java Main
45
Pass
● PS C:\Users\jitma\OneDrive\Desktop\Temp> java Main
12
Fail
○ PS C:\Users\jitma\OneDrive\Desktop\Temp>
```



$x\%$ of y
 10% of 1200

Formula

$$\frac{x}{100} \times xy$$

$$\frac{10}{100} \times \frac{1200}{120}$$

$$\frac{10}{100} \times 1200 = 120 \quad \hookrightarrow \text{Discount Value}$$

$$\text{Final Val} = 1200 - 120 = \underline{\underline{1080}}$$

$$\text{User} \rightarrow \frac{\text{no of units}}{\text{ }} \rightarrow 74$$

$$\boxed{\text{Bill}} = \text{no of units} * 100 \rightarrow 7400$$

\downarrow
before discount

$$\text{Discount} = \frac{10}{100} \times 7400 = \underline{\underline{740}}$$

$$\boxed{\text{Bill}} = 7400 - 740 = \underline{\underline{6660}}$$

After discount

Variable Names

↳ small case character

↳ camel casing ↴

noOfUnitsPurchasedByUser

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

public class Solution {

    public static void main(String[] args) {
        Scanner scn = new Scanner(System.in);
        int noOfUnits = scn.nextInt();
        int billBeforeDiscount = noOfUnits * 100;
        if(billBeforeDiscount > 1000) {
            int discount = (billBeforeDiscount * 10) / 100;
            int finalBill = billBeforeDiscount - discount;
            System.out.println(finalBill);
        } else {
            System.out.println(billBeforeDiscount);
        }
    }
}
```

$[-1000, 1000]$

~~8 bytes~~

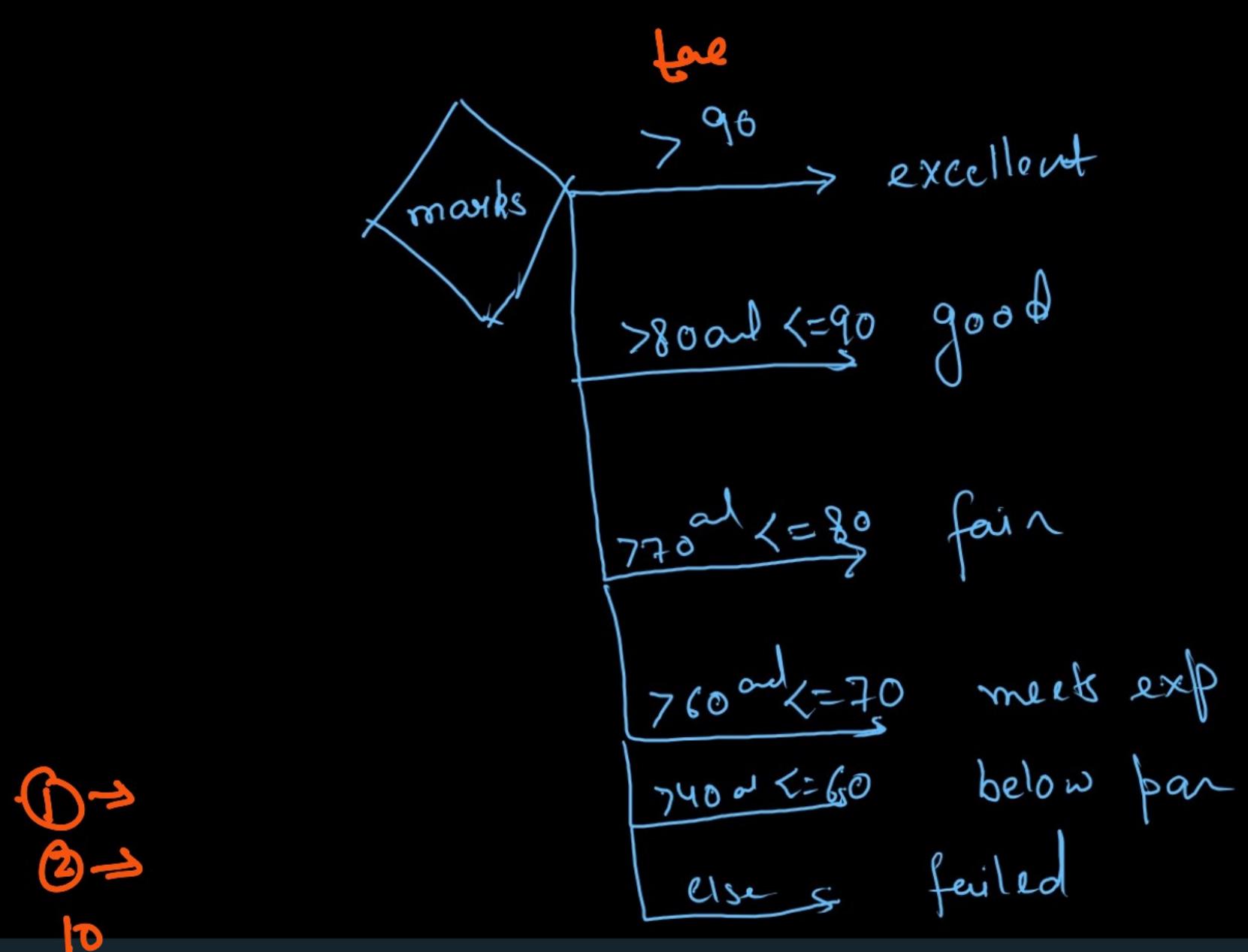
64

int a = 1000
int b = 900

int c = a+b;
1900

You are given marks of a student as an integer input. You need to print according to the following rules: 1 for marks above 90, print excellent. 2 for marks above 80 and less than or equal to 90, print good. 3 for marks above 70 and less than or equal to 80, print fair. 4 for marks above 60 and less than or equal to 70, print meets expectations. 5 for marks above 40 and less than or equal to 60, print below par. 6 print failed if none of the above conditions follow.

94



→ if (Condition 1) {
→ if (Condition 2) {
→ if (Condition 3) {
→ else {
→ }
→ }
→ }
→ }

if else

```

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

public class Solution {

    public static void main(String[] args) {
        Scanner scn = new Scanner(System.in);
        int marks = scn.nextInt();
        if(marks > 90) { //96
            System.out.println("excellent");
        } else if(marks > 80){
            System.out.println("good");
        } else if(marks > 70) {
            System.out.println("fair");
        } else if(marks > 60) {
            System.out.println("meets expectations");
        } else if(marks > 40){
            System.out.println("below par");
        } else {
            System.out.println("failed");
        }
    }
}
  
```

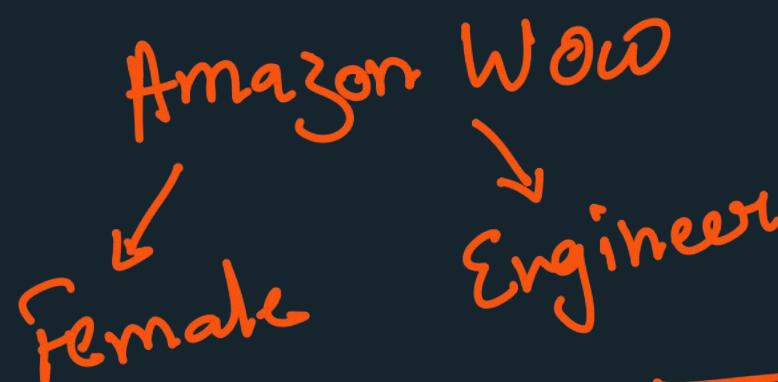
Logical Operators

And

{ → && → And operator

1

1



\equiv

\equiv

Cond 1

↑ ↑

$\equiv \text{gender} == \text{female}$

\equiv

\equiv

Cond 2

↑

$\equiv \text{if} > \text{engineer}$

&&

$\&\&$ → true \rightarrow ✓ ✓
false \rightarrow otherwise

$\wedge\wedge$	{	false false true true	true → false false → false false → false true → true
----------------	---	--------------------------------	---

J Main.java

```
1 import java.util.*;
2 public class Main{
3     public static void main(String[] args) {
4         System.out.println((5 < 7) && (6 > 3)); t && t → t
5         System.out.println(true && (7 <= (3 / 4))); t && f → f
6         System.out.println(false && false); f
7         System.out.println((0 == (3 / 4)) && (0 == (3 / 4.0)))); t && f
8         System.out.println("true && true"); true && true
9     }
10 }
```

$$\frac{3}{4} \rightarrow 0$$

import java.util.*;
public class Main {
 public static void main(String[] args) {
 System.out.println(0.75);
 }
}

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

Σ powershell + × ⊞ 🗑 ... ^ >

- PS C:\Users\jitma\OneDrive\Desktop\FS-14> javac Main.java
 - PS C:\Users\jitma\OneDrive\Desktop\FS-14> java Main
true
false
false
false
true && true
 - PS C:\Users\jitma\OneDrive\Desktop\FS-14> □

OR → ||

≤
Cond 1

≤
Cond 2

↳ Backed

↳ fronted

get the
job

false || false = false

t

|| f

t

false || true = true

f

|| t

t

true || false = true

t

|| t

t

true || true = true

f

|| f

f

J Main.java

```

1 import java.util.*;
2 public class Main{ ()
3     public static void main(String[] args) {
4         System.out.println((5 < 7) || (6 > 3)); // t || t => t
5         System.out.println(true || (7 <= (3 / 4))); // t
6         System.out.println(false || false); // f
7         System.out.println((0 == (3 / 4) || (0 == (3 / 4.0)))); //t
8         System.out.println("true || true"); // true || true
9     }
10 }
```

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

pc

● PS C:\Users\jitma\OneDrive\Desktop\FS-14> java Main

● true

true

false

true

true || true

○ PS C:\Users\jitma\OneDrive\Desktop\FS-14> []

NOT (!)

& & || !
and or not



! (Condⁿ)

Condⁿ

true → false
false → true

J Main.java

```
1 import java.util.*;
2 public class Main{
3     public static void main(String[] args) {
4         System.out.println(!(5 < 7)); //f
5         System.out.println(!(7 <= (3 / 4))); //t
6         System.out.println(!false); //t
7         System.out.println((!(0 == (3 / 4.0)))); //t
8         System.out.println("!false"); // !false
9     }
10 }
```

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

```
PS C:\Users\jitma\OneDrive\Desktop\FS-14> javac Main.java
```

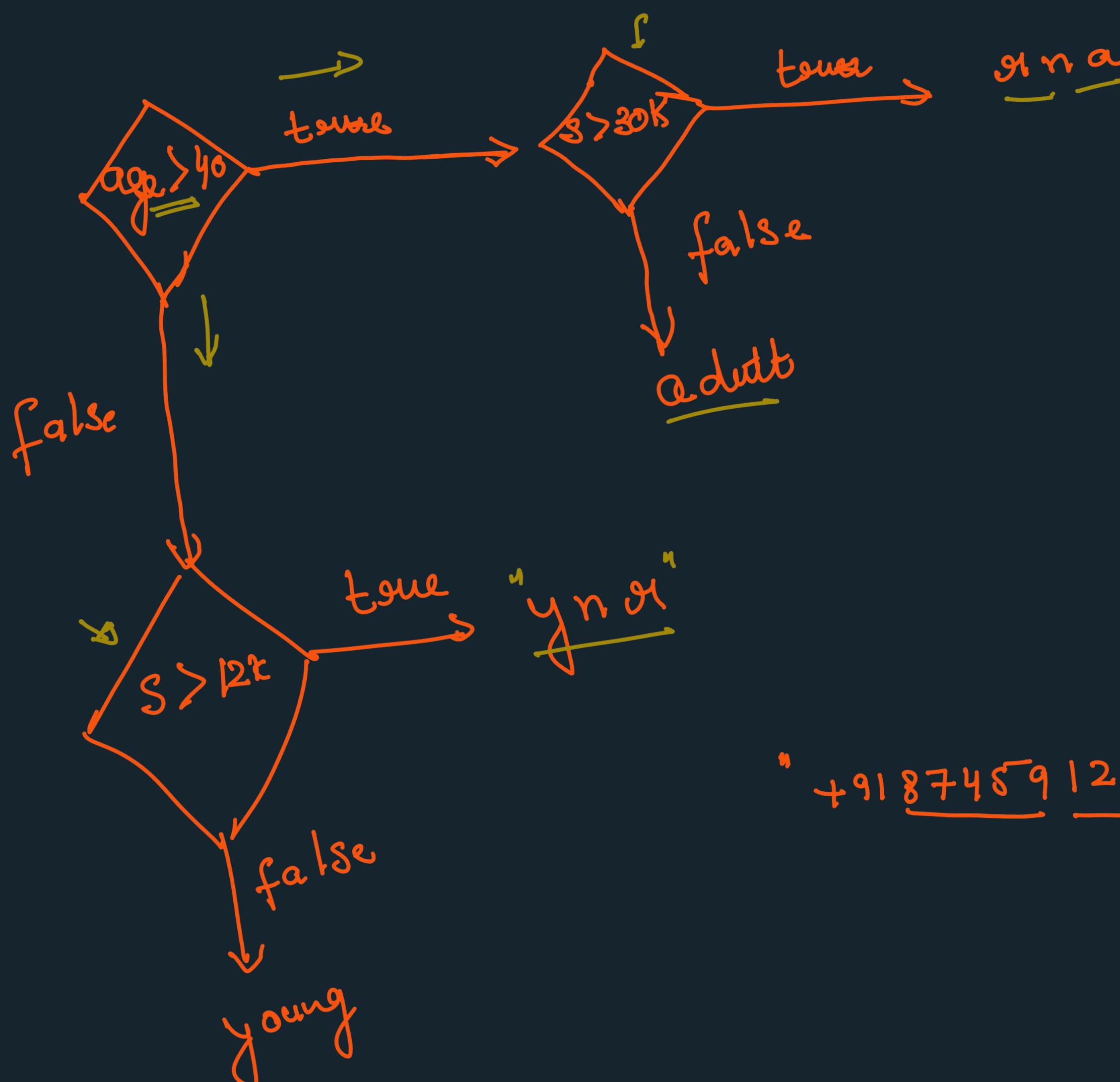
● PS C:\Users\jitma\OneDrive\Desktop\FS-14> java Main

false

true

true

$\rightarrow \frac{35}{\geq 40}$
 $\downarrow \frac{15K}{15K}$
 $\rightarrow \frac{12}{0}$



" +91 8745912135 "

Java 8

```

1 import java.io.*;
2 import java.util.*;
3
4 public class Solution {
5
6     public static void main(String[] args) {
7         Scanner scn = new Scanner(System.in);
8         int age = scn.nextInt(), salary = scn.nextInt();
9         if(age > 40) {
10             if(salary >= 30000) {
11                 System.out.println("You are rich and adult");
12             } else {
13                 System.out.println("You are an adult");
14             }
15         } else {
16             if(salary >= 12000) {
17                 System.out.println("You are rich and young");
18             } else {
19                 System.out.println("You are young");
20             }
21         }
22     }
23 }
24 
```

The screenshot shows the Java code above. A hand-drawn flowchart is overlaid on the right side, corresponding to the logic in the code. The flowchart uses blue lines and circles to mark specific points of interest:

- Point 1: After the first if statement (age > 40).
- Point 2: After the first if-else block (salary >= 30000).
- Point 3: After the second if-else block (salary >= 12000).
- Point 4: After the final else block.

C characters

ASCII

'a' to 'z'

'A' to 'Z'

'0' to '9'

? → space !

@ * . - #

→ char c = '\u20ac'

unical

Char c = '*' ;

Dec	Char	Dec	Char	Dec	Char	Dec	Char
0	NUL (null)	32	SPACE	64	@	96	`
1	SOH (start of heading)	33	!	65	A	97	a
2	STX (start of text)	34	"	66	B	98	b
3	ETX (end of text)	35	#	67	C	99	c
4	EOT (end of transmission)	36	\$	68	D	100	d
5	ENQ (enquiry)	37	%	69	E	101	e
6	ACK (acknowledge)	38	&	70	F	102	f
7	BEL (bell)	39	'	71	G	103	g
8	BS (backspace)	40	(72	H	104	h
9	TAB (horizontal tab)	41)	73	I	105	i
10	LF (NL line feed, new line)	42	*	74	J	106	j
11	VT (vertical tab)	43	+	75	K	107	k
12	FF (NP form feed, new page)	44	,	76	L	108	l
13	CR (carriage return)	45	-	77	M	109	m
14	SO (shift out)	46	.	78	N	110	n
15	SI (shift in)	47	/	79	O	111	o
16	DLE (data link escape)	48	0	80	P	112	p
17	DC1 (device control 1)	49	1	81	Q	113	q
18	DC2 (device control 2)	50	2	82	R	114	r
19	DC3 (device control 3)	51	3	83	S	115	s
20	DC4 (device control 4)	52	4	84	T	116	t
21	NAK (negative acknowledge)	53	5	85	U	117	u
22	SYN (synchronous idle)	54	6	86	V	118	v
23	ETB (end of trans. block)	55	7	87	W	119	w
24	CAN (cancel)	56	8	88	X	120	x
25	EM (end of medium)	57	9	89	Y	121	y
26	SUB (substitute)	58	:	90	Z	122	z
27	ESC (escape)	59	;	91	[123	{
28	FS (file separator)	60	<	92	\	124	
29	GS (group separator)	61	=	93]	125	}
30	RS (record separator)	62	>	94	^	126	~
31	US (unit separator)	63	?	95	_	127	DEL

$\rightarrow x \neq 3$



true

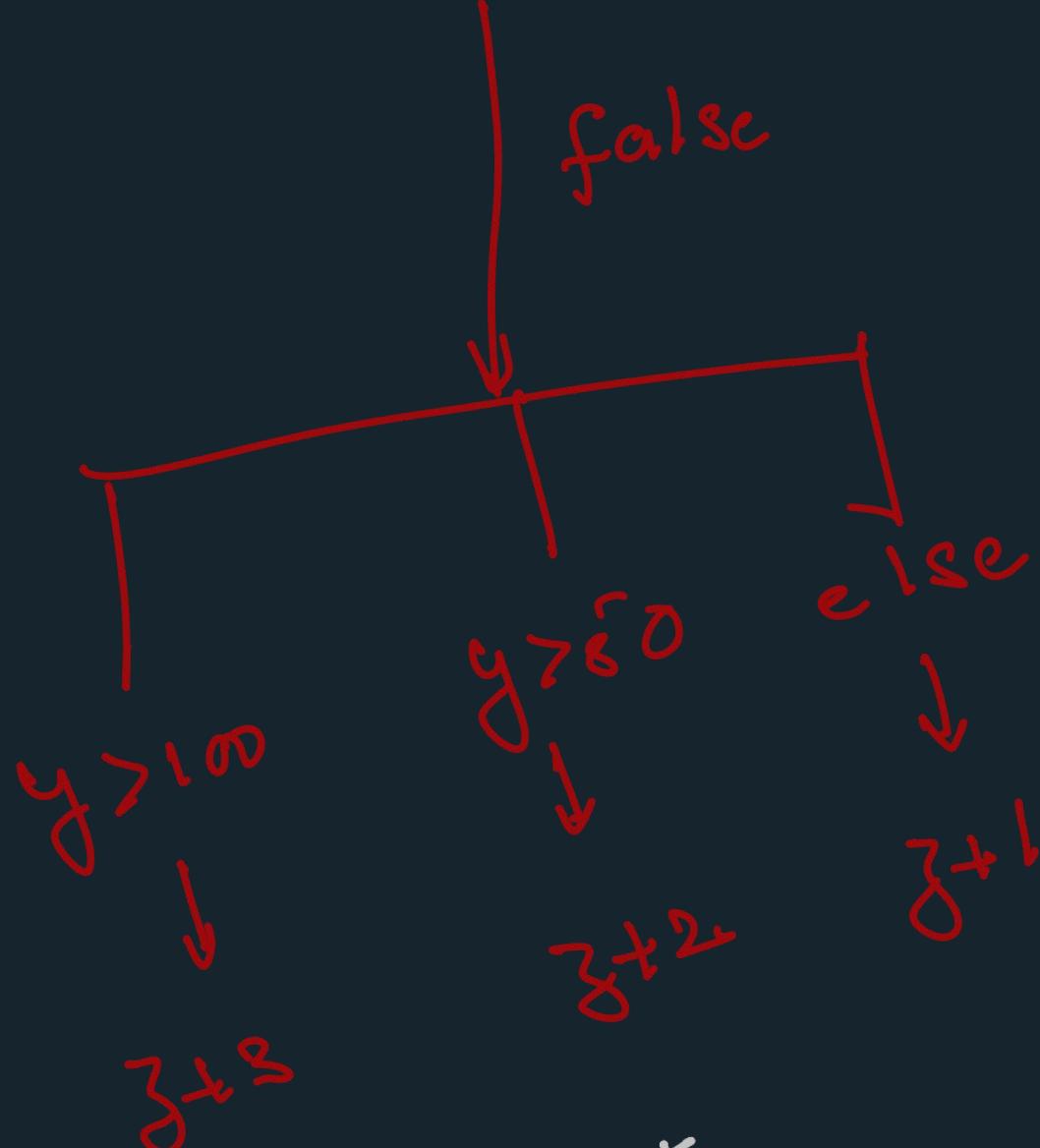
$y > 100 \rightarrow z \leftarrow 100$
 $y > 50 \rightarrow z \leftarrow 50$
 $else \rightarrow z \leftarrow 10$



112

57

$x \leftarrow \begin{cases} 36 & \\ +4 & 45 \\ 12 & \end{cases}$



$$\begin{cases} a = 32 \\ b = 46 \\ c = 38 \end{cases}$$

Finding $\overset{c=38}{\cancel{2^{nd}}}$ largest

$$\textcircled{1} \quad \begin{bmatrix} 44 \\ 42 \\ 43 \end{bmatrix}$$

$\rightarrow a > b$ and $a > c$
 $\rightarrow a$ cannot be your ans

$$\begin{bmatrix} b \\ c \end{bmatrix}$$

whichever is greater

$\rightarrow b > a$ and $b > c$

$$\begin{bmatrix} a \\ c \end{bmatrix}$$

greater one

$c = \cancel{a}$

$$\begin{bmatrix} a > 100 \rightarrow b \\ \cancel{a} > c \end{bmatrix}$$

a b
equal one

Ternary op.

$$\begin{matrix} c \\ a & b \\ & c \end{matrix}$$

if (cond) {

① $\cancel{a == b}$

} else {

② $\cancel{a == b}$

}

if ($a > 100$)

 sys.out.println($a > 100 ? b : c$);

else

 sys.out.println($a > 100 ? b : c$);

$\underset{\text{Cond}}{\cancel{a == b}}$: $\underset{\text{true}}{\cancel{a > 100}}$: $\underset{\text{false}}{\cancel{a < 100}}$;

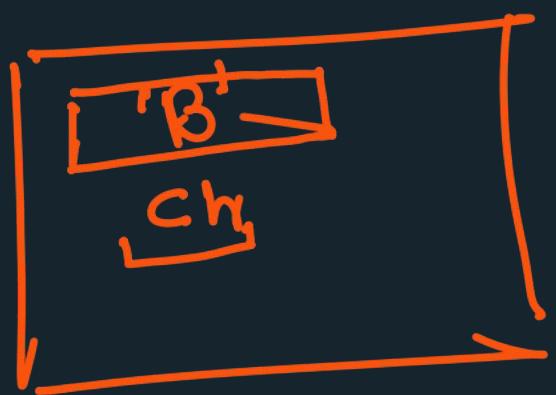
78

\cancel{t}

\cancel{s}

char ch = scn.next() . charAt(0);

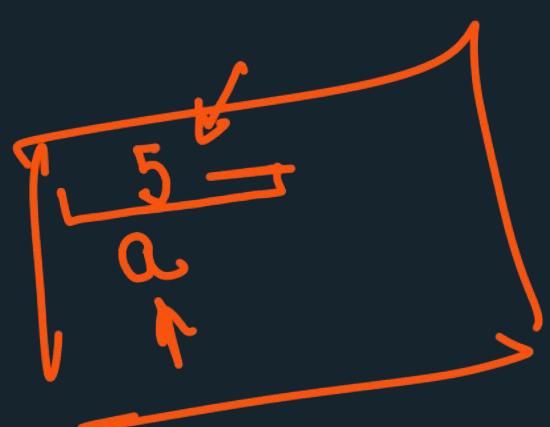
'B' ←
= 'example'



2 bytes

16 bits

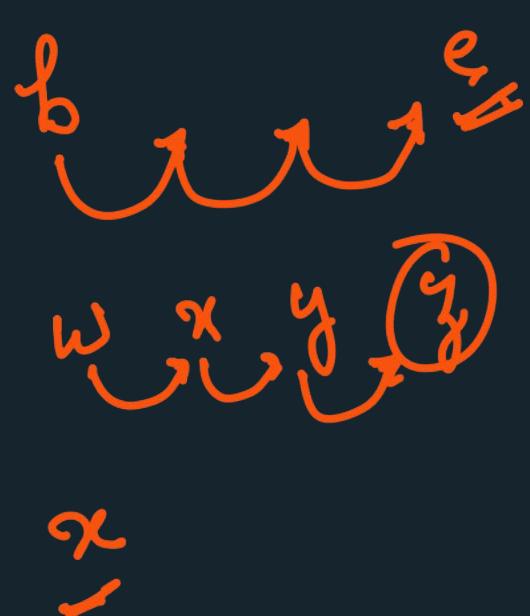
ch
↳ 'B'



5

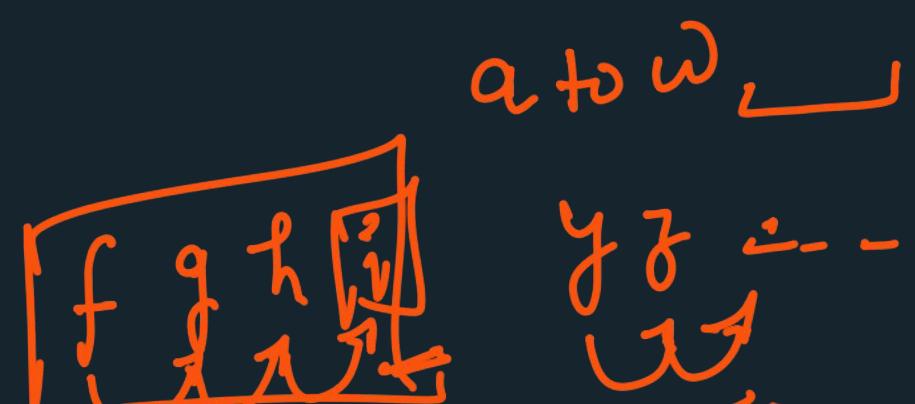
Explain char input again

char input → [a to ω]
↳ +3

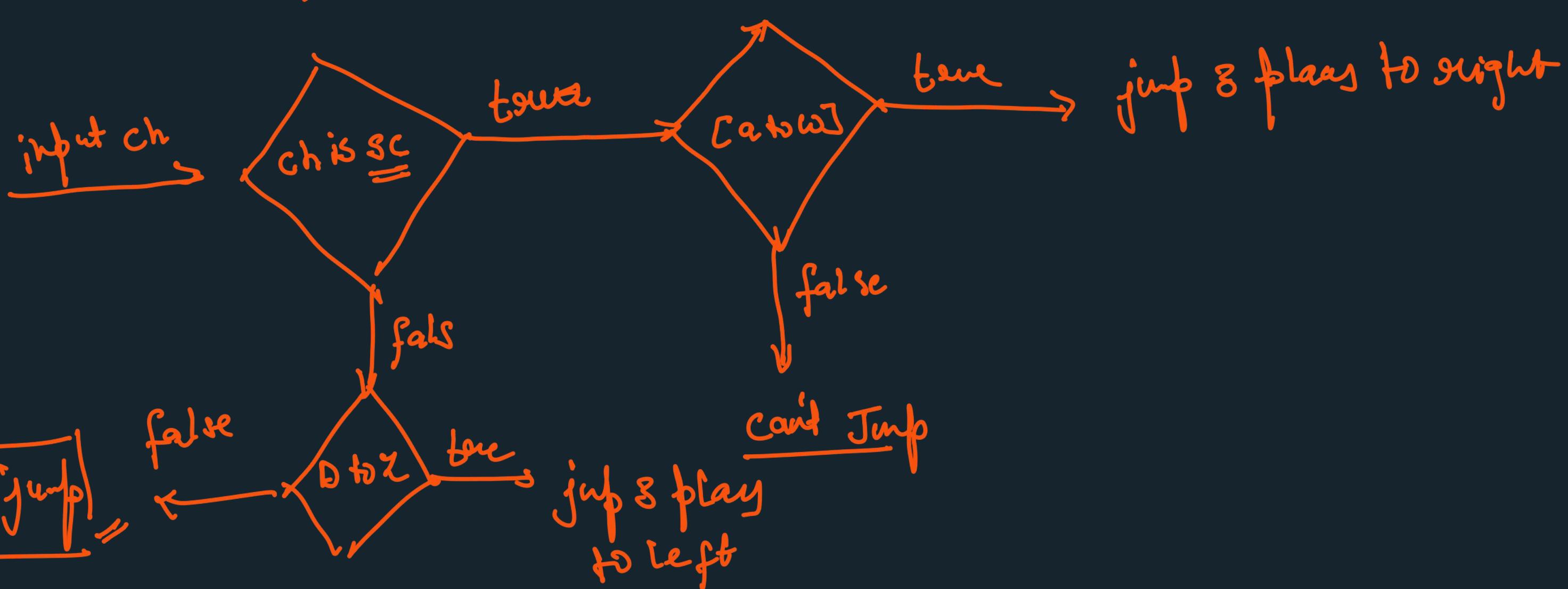
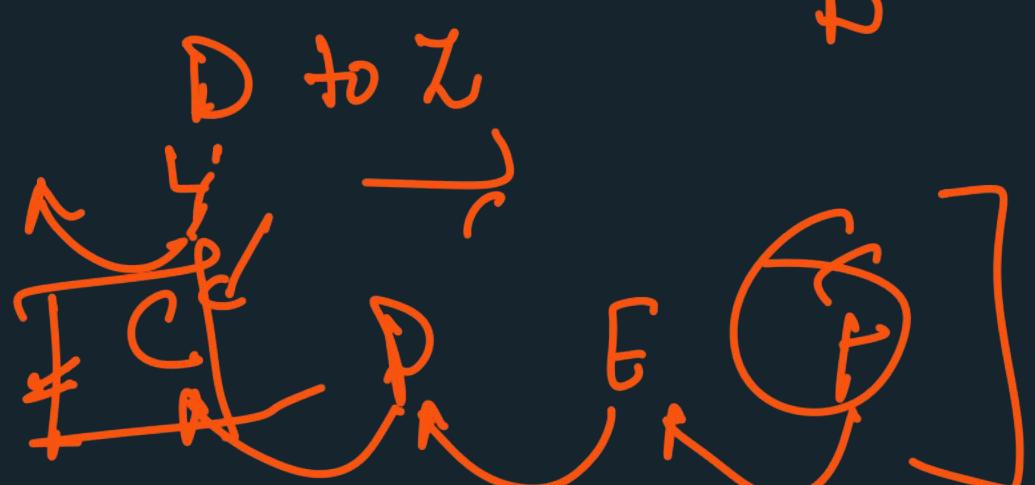


a b c d
↳

[a to ω]



a to ω ↳
y z ↳



```

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

public class Solution {

    public static void main(String[] args) {
        Scanner scn = new Scanner(System.in);
        char ch = scn.next().charAt(0);
        if(ch >= 'a' && ch <= 'w') System.out.println((char)(ch + 3));
        else if(ch >= 'D' && ch <= 'Z') System.out.println((char)(ch - 3));
        else System.out.println("Can't jump");
    }
}

```

$$51 - 48 \quad \{ \text{jump} \\ 3 - 0$$



$$17 - 10 =$$

$$10 - 10 = \\ 48 - 48 = 0$$

$$11 - 10 = 1 \\ 49 - 48 = 1$$

8 units

$$13 - 10 \rightarrow 3$$

$$\text{char } c = '7';$$

$$13 - 10 \\ 51 - 48$$

Character: getNumericValue (≤)

class ↗

↖ ↗ 7

String



"Jitmāneऽत्यागि" → 14

$$n = 12$$

index → A b e s h - Sa a k a M $n-1$
 0 1 2 3 4 5 6 7 8 9 10 11 ↑

[0, n-1]

String str = "Hello" ;

str.length() → 5

charAt
str.charAt(i) = "Deepak Kumar" ;
↓ ↓ ↓ ↓ ↓ ↓
0 1 2 3 4 5 6 7 8 9 10 11 ith
Deepak Kumar ;
↓ ↓
D K

str.charAt(i) = "K" ;

str.charAt(i) → character present at ith index

String Input

next() → stops taking input when space is encountered

nextLine() → a when it encounters newline character

scn.next() → word

next() =

Deepak - Kumar

Loops

(for (int i = s ; i < end ; i ++) {
 do something to be repeated

{ for (int i = 0; i < 10; i++)
 // things that are to be repeated

$$1 \leq n \leq 10$$

进 \rightarrow 进 \Rightarrow 进

```
    } }  
    Start           cond           update  
    for (int i = 1; i <= 10; i += 1) {  
        System.out.println(i);  
    }  
    f-lse
```

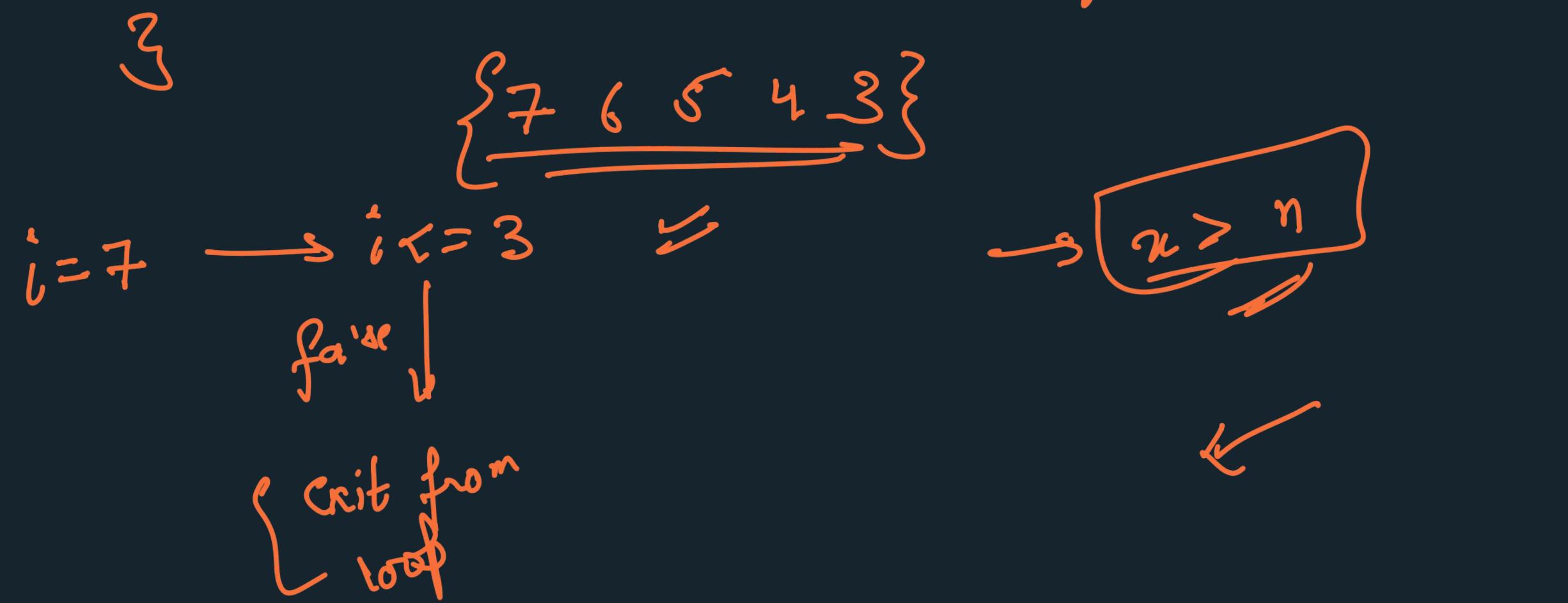
$\rightarrow \text{System.out.print}(n, c);$

A hand-drawn diagram illustrating a while loop iteration. The process starts with $i = 1$. The condition $i \leq 10$ is checked. If **false**, the loop ends. If **true**, **work** is performed, and then i is updated to $i + 1$. This leads to the next iteration where $i = 2$.

3

$$\underline{x} = \underline{z}^T \quad n = 7, 3$$

for (int $i = x$; $i <= n$; $i++$) {
 $x[i] = n$
 $i = i + 1$
 $i = i - 1$



$$4 \times 1 = 4$$

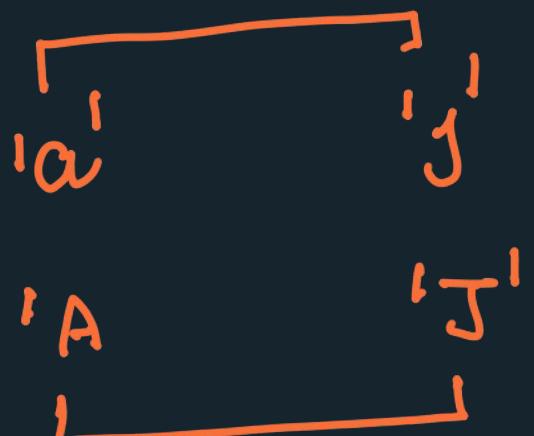
$$4 \times 2 = 8$$

$$4 \times 3 = 12$$

$$\vdots \vdots \vdots$$

$$4 \times 10 = 40$$

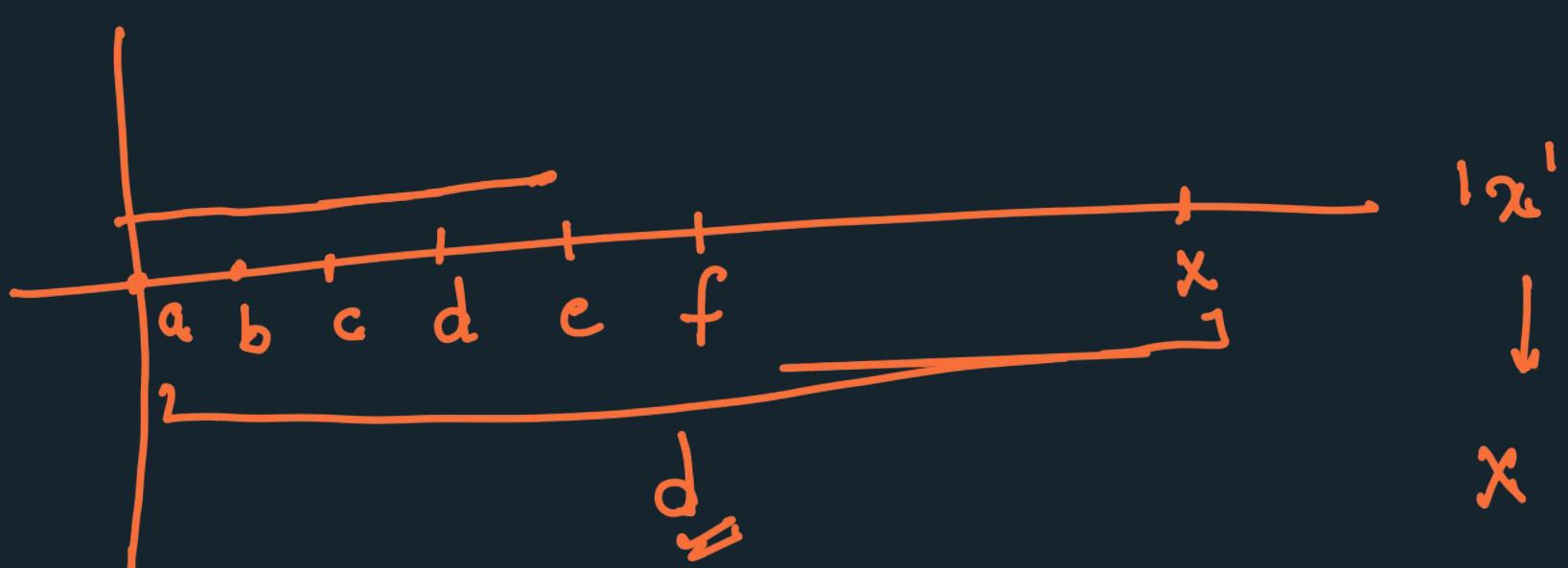
$$\underline{4x^i}$$



algo

'k'
 'x'
 'A' + x → 'K'

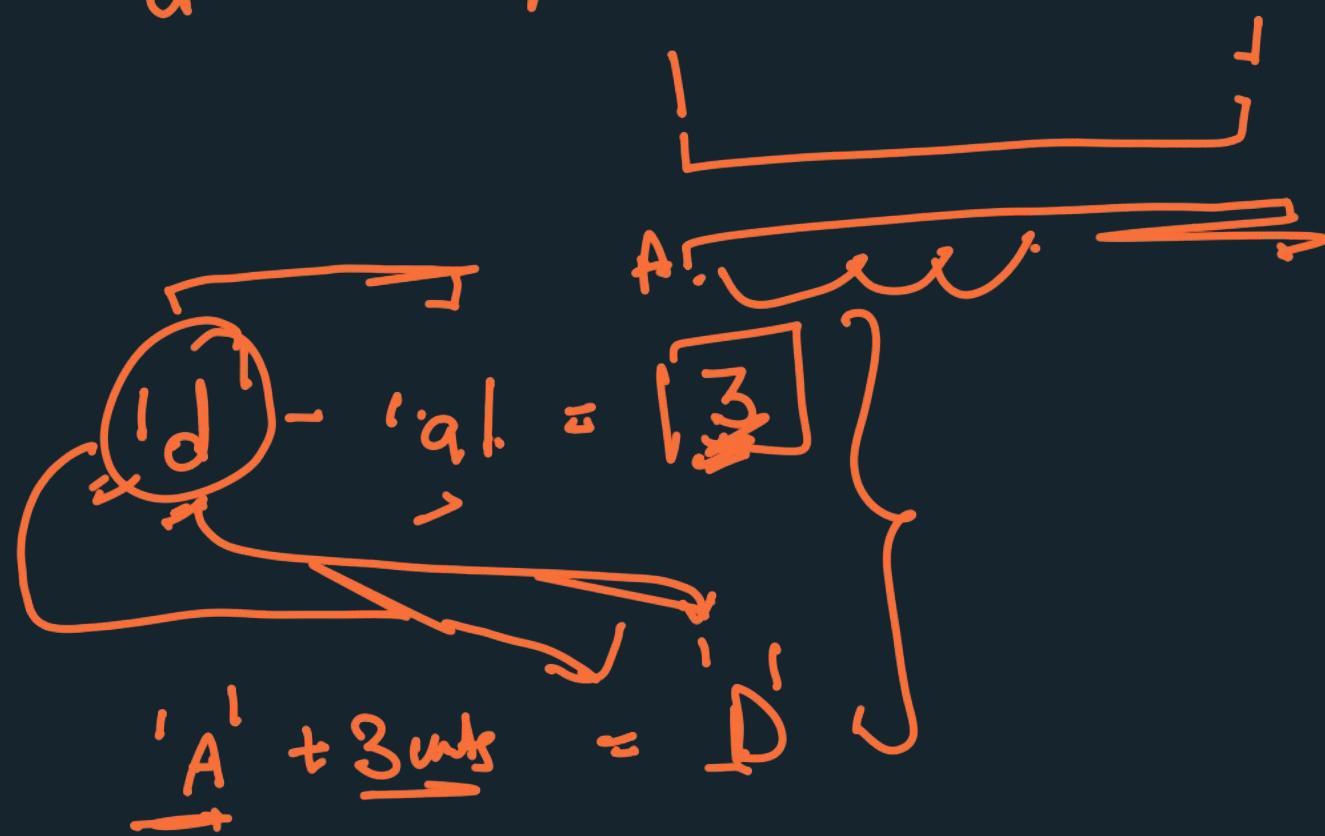
$$'x' - 'a' = d$$



$$\begin{aligned} 'a' - 'a' &= 0 \\ 'b' - 'a' &= 1 \\ 'c' - 'a' &= 2 \end{aligned}$$

⋮

$$'d' \rightarrow 'p'$$



$$'a' \xrightarrow{\text{dist}} f$$

$$'A' \xrightarrow{\text{dist}} f$$

Indexing in String

→ 0 1 2 3 4 5 ←
Kalyan

Input → [] to 2^n

$\begin{bmatrix} 5 \\ 7 \\ 9 \\ 40 \\ 3 \\ 4 \end{bmatrix} = 32$

$\begin{bmatrix} 7 \\ 2 \\ 9 \\ 2 \\ 40 \end{bmatrix} = 128$

$\begin{bmatrix} 2 \\ 2 \\ 2 \end{bmatrix} = 612$

$\frac{8}{2^4 = 16}$

$$[0, len-1]$$

→ Input → [] → Output
process

$\sqrt[5]{2^5} = 2$

5 → n

3
10

2^n

2^n

$2^3 = 8$

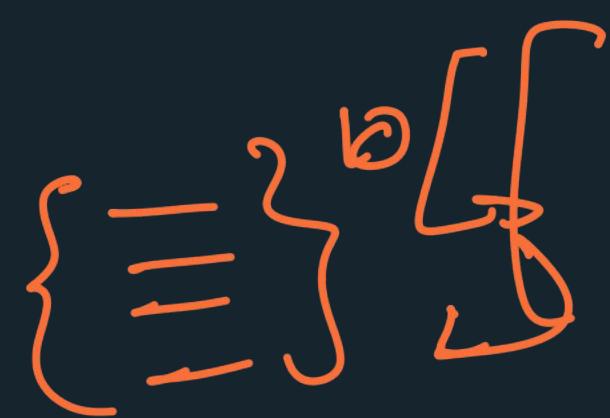
$2^{10} = 1024$

$$n \approx 35$$

1 2 4 8 16 32

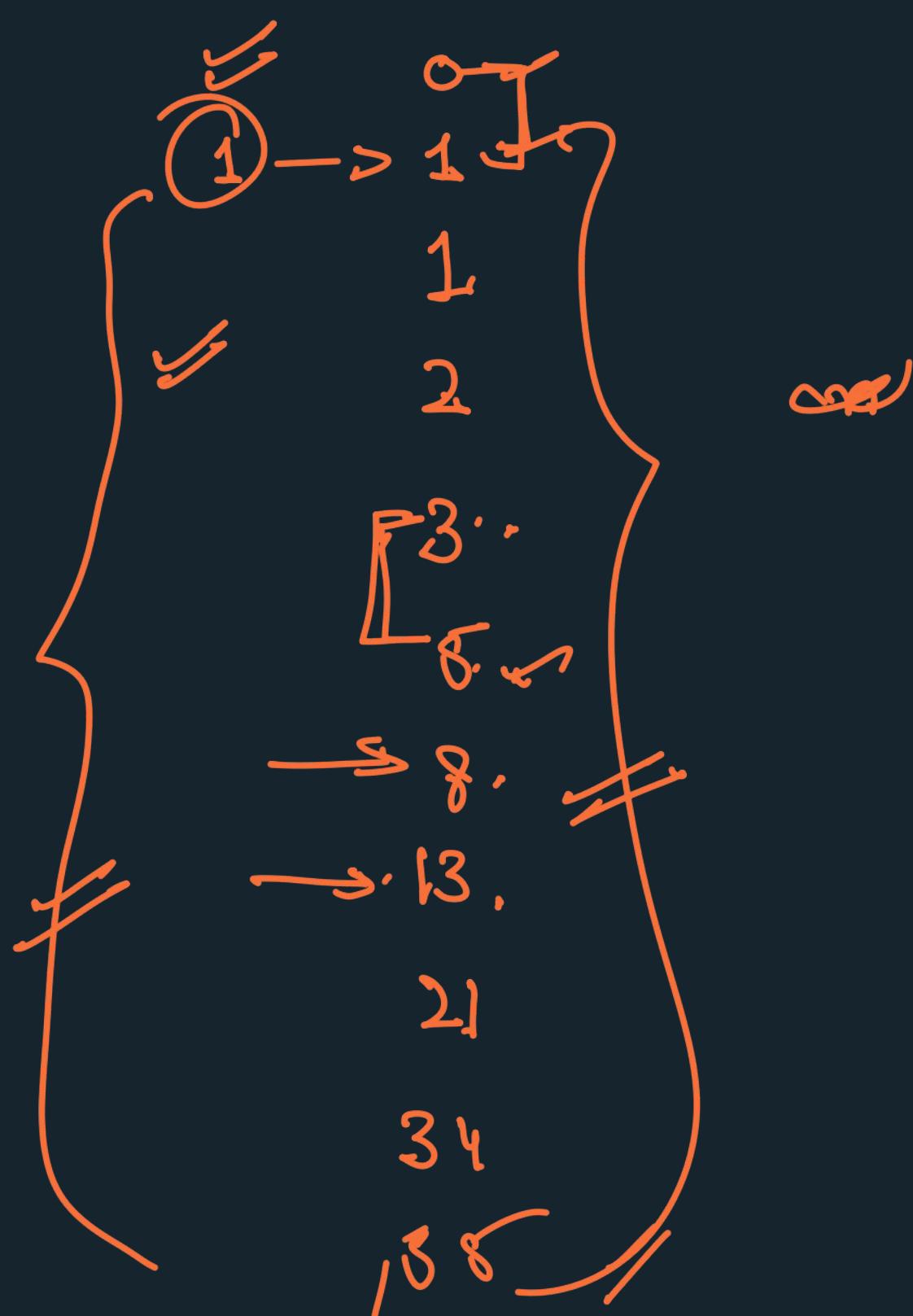
$11^k \rightarrow c++$

2-3 mas



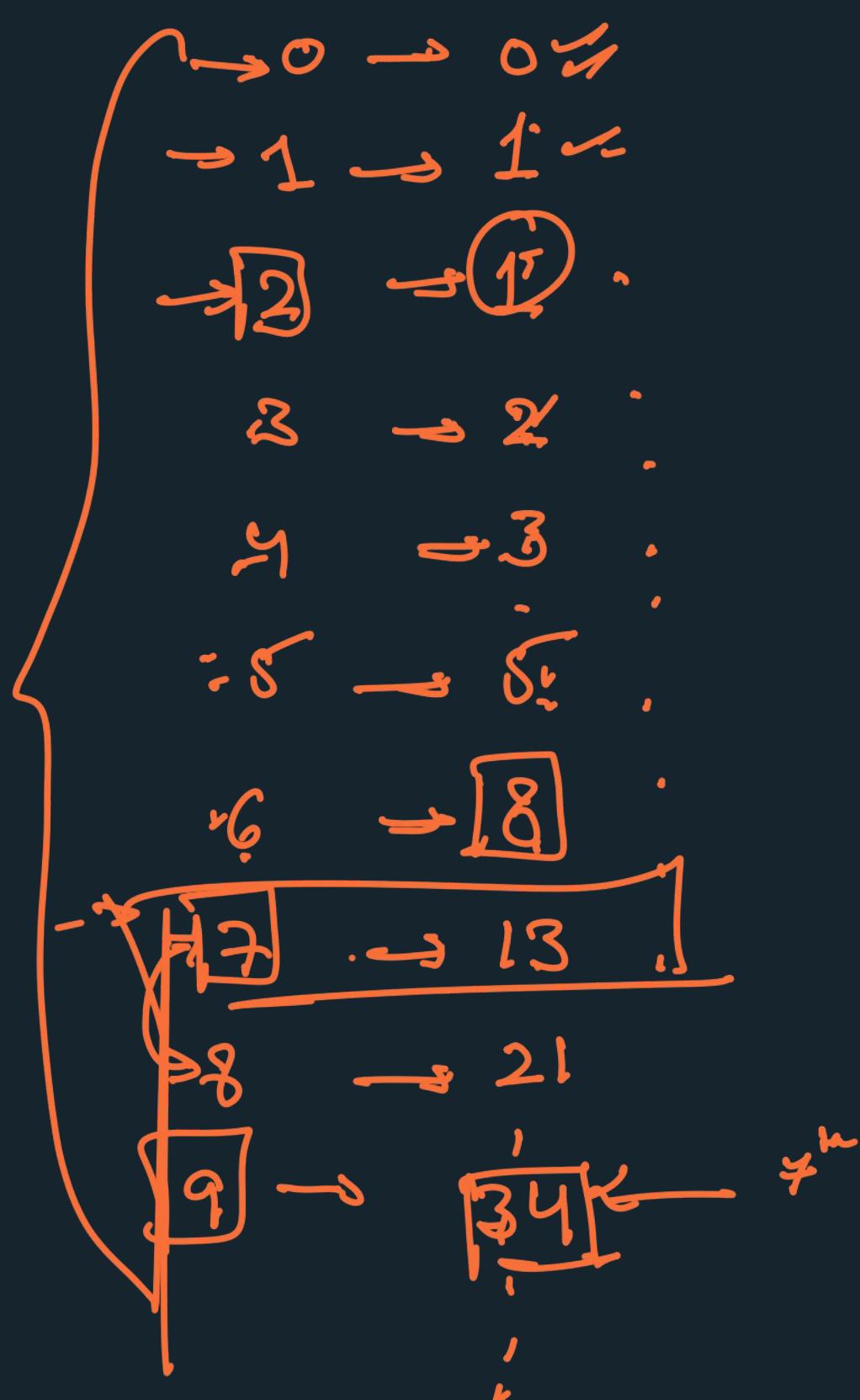
\Rightarrow

$$\underline{f(n)} = f(n-1) + f(n-2)$$



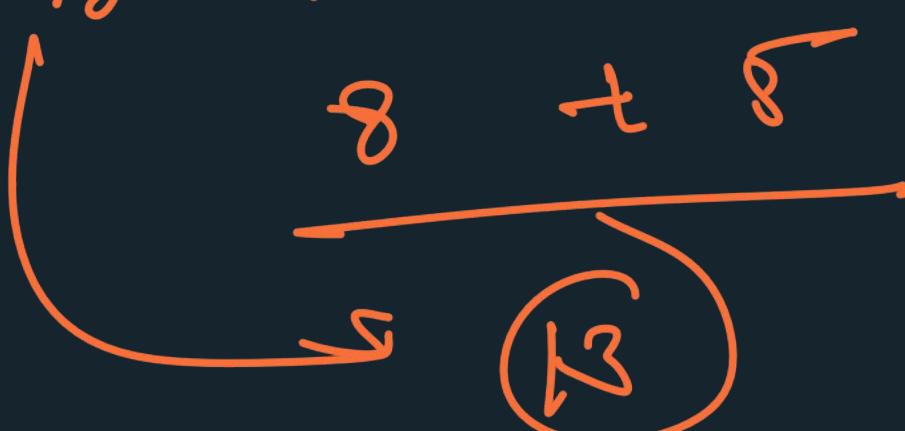
Fibonacci

$$= 0 \underset{2}{\overset{2}{\dots}} 2 \underset{2}{\overset{2}{\dots}} 4 \underset{2}{\overset{2}{\dots}} 8 \underset{2}{\overset{2}{\dots}} 16 \underset{2}{\overset{2}{\dots}} 32 \underset{2}{\overset{2}{\dots}} 64 \dots$$



$$f(n) = f(n-1) + f(n-2)$$

$$f(7) = f(6) + f(5)$$



$$a=0 \\ b=1$$

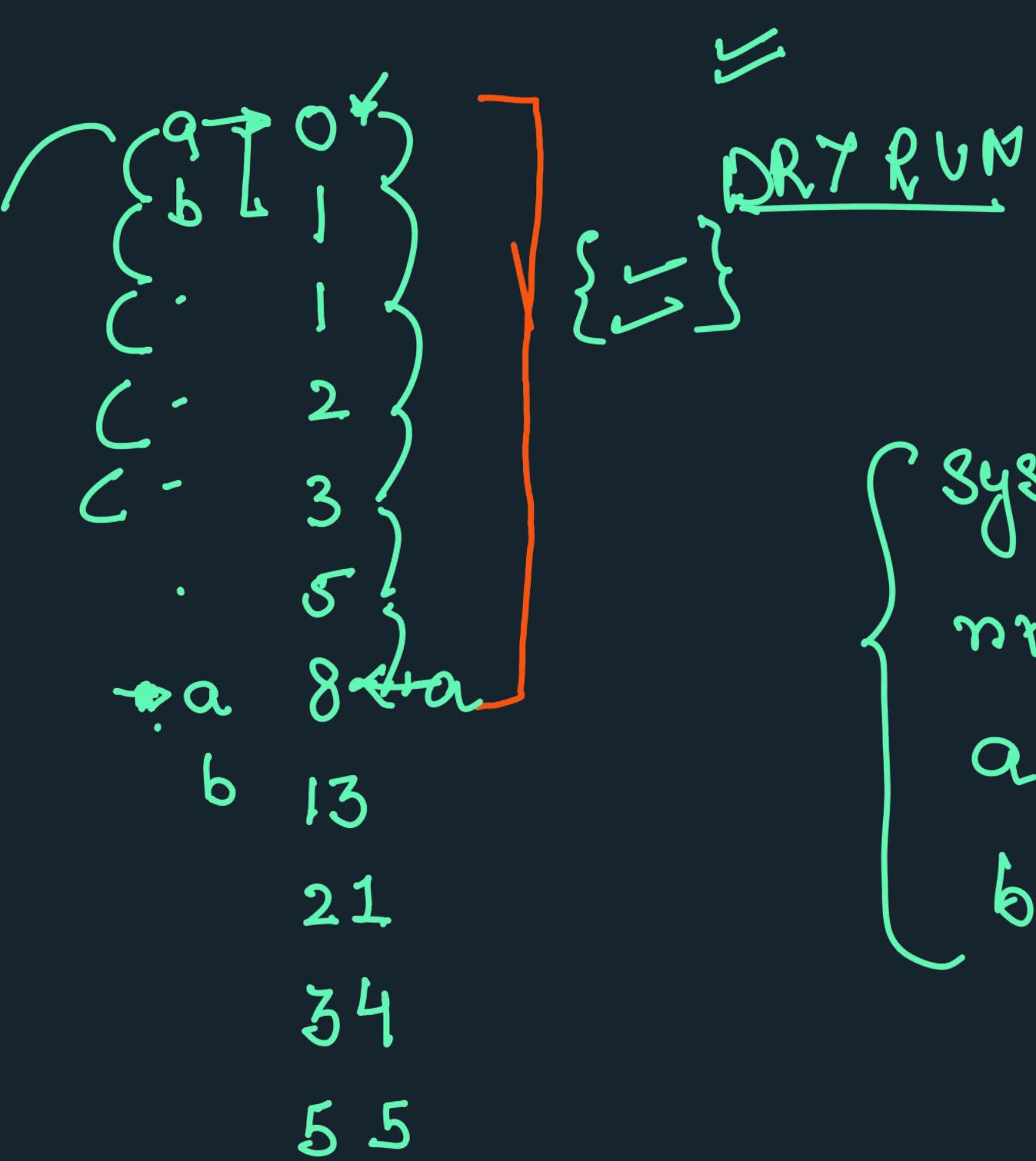
Fibonacci

G

$$\begin{cases} a = 0 \\ b = 1 \end{cases}$$

$$\text{nextTerm} = a + b$$

$$n \rightarrow 7$$



0 1 1 2 3 5 8
→

$\left\{ \begin{array}{l} \text{sys0(a);} \\ \text{nt = a + b} \\ a = b \\ b = nt \end{array} \right.$

no of ite = $\lfloor \frac{n}{2} \rfloor + 1$

$\left[\begin{array}{l} a = 0 \xrightarrow{} 1 \xrightarrow{} 2 \xrightarrow{} 3 \xrightarrow{} 5 \xrightarrow{} 8 \xrightarrow{} 13 \\ b = 1 \xrightarrow{} 1 \xrightarrow{} 2 \xrightarrow{} 3 \xrightarrow{} 5 \xrightarrow{} 8 \xrightarrow{} 13 \end{array} \right]$
 nextTerm = $2 \xrightarrow{} 3 \xrightarrow{} 5 \xrightarrow{} 8 \xrightarrow{} 13 \xrightarrow{} 21$

→ 7

While Loop

for C_{int} i ; cond ; update

init → cond → update

↳ iterator

[init
 ↑ time]

while (cond) {

P.S. ↓
↓ [i++] use first chang/update
↓ [i--] the val later

Pre ↓
↓ [++i ; - -i] update first
↓ [- -i] use later

inside the loop → update;

}

N = 50

K = 5

L = 4 ←

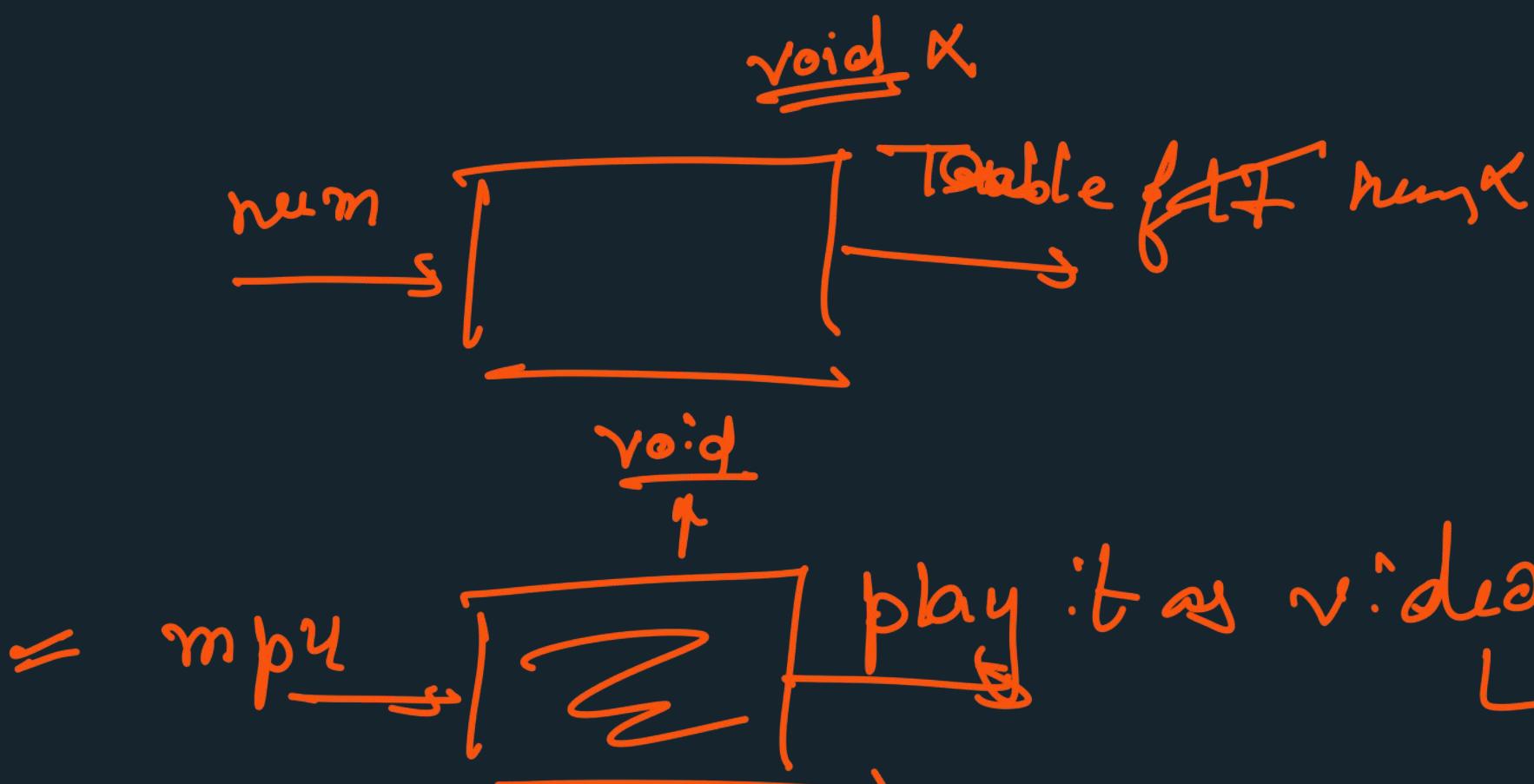
50 45 40 35 30 5



Functions → reusable pieces of code

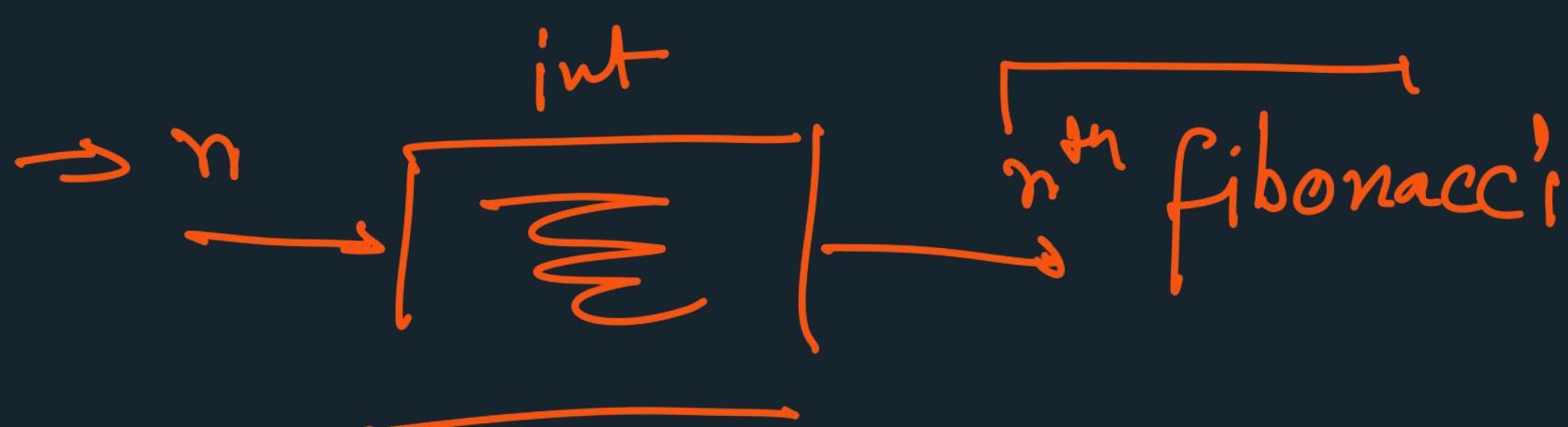
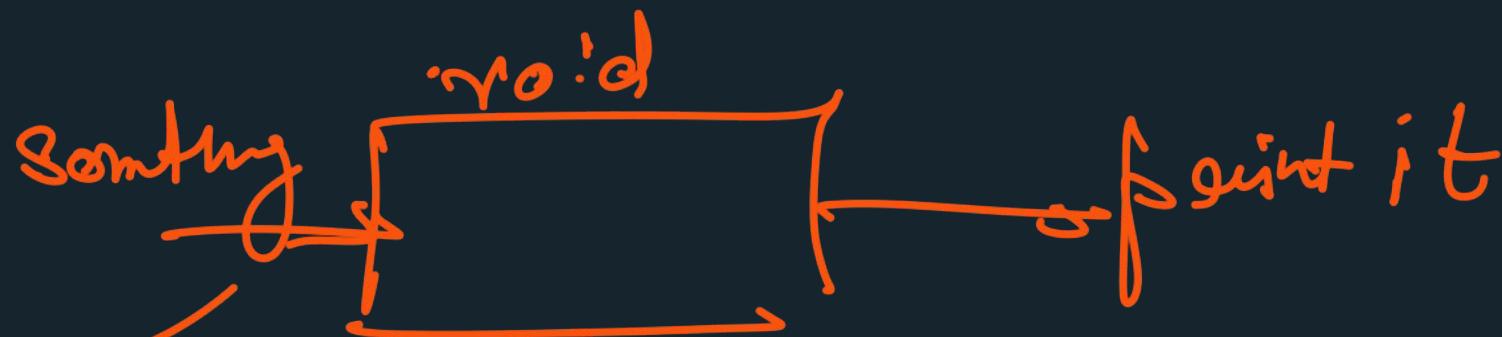
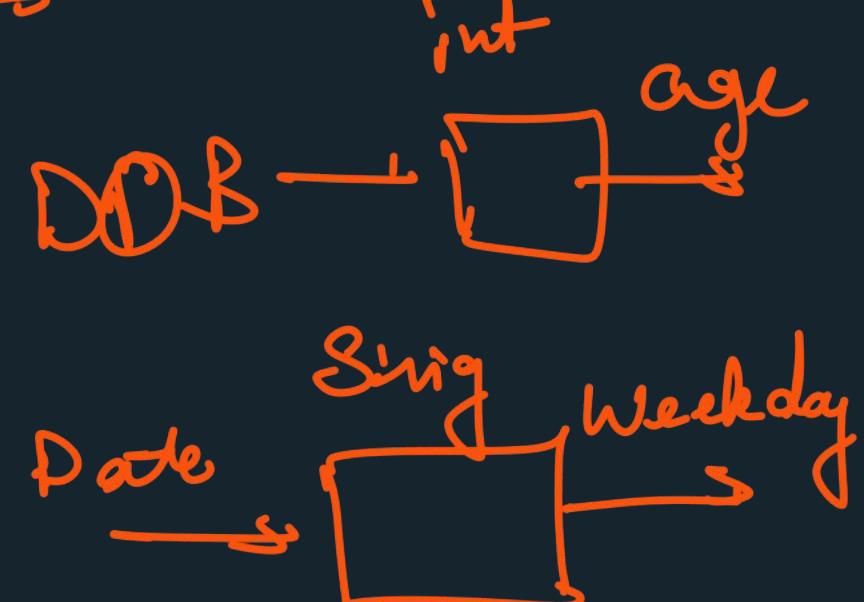


int =
String
char



$$2 \times 1 = 2$$

$$2 \times 2 = 4$$



P S , return type, func name (arguments) {

// BODY

}

Patterns



→ 12 rows → 12 times
n stars = $\frac{n}{2}$ (n)
 $n \rightarrow * \rightarrow \{ \text{int} \}$



```
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 nst = 1, nsp = n - 1; // nst - no of stars
        //nsp - no of spaces
        for(int r = 1; r <= n; r++) {
            //print perfect number of spaces
            for(int sp = 1; sp <= nsp; sp++)
                System.out.print(" ");

            //print perfect number of stars
            for(int st = 1; st <= nst; st++)
                System.out.print("*");
            //move the cursor to the next line
            System.out.println();
        }

        // adjust nst and nsp for next row
        nst++;
        nsp--;
    }
}
```