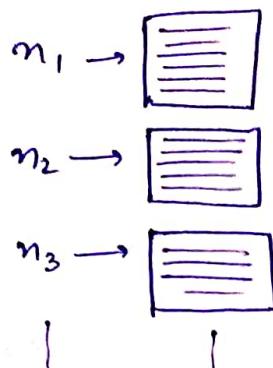


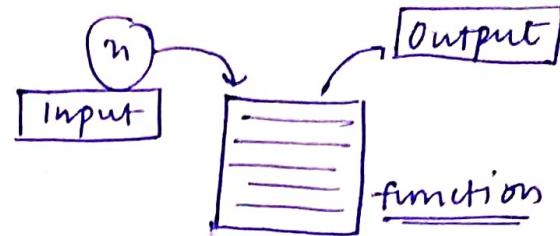
INTRODUCTION TO FUNCTION

$$f(x) = x^2 \rightarrow \text{Maths}$$



change har n ke liye karna hoga
 Delete krne hai toh n (every) ke liye karna hoga.

D R Y
 DONOT REPEAT YOURSELF



$$f(x) = x^2 + 3x + 4,$$

$$f(2) = 2^2 + 3(2) + 4 = 14$$

$$f(3) = 3^2 + 3(3) + 4 = 22$$

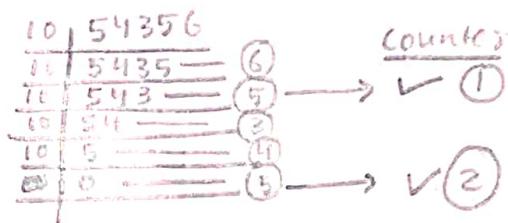
BENEFITS OF FUNCTION

- * CODE CLEAR
- * CODE MODULAR (MODULARITY)
- * MAINTAINABILITY

COUNT DIGIT FREQUENCY

$n = 54356$
 $d = 5$

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



FUNCTION NAME
 $\text{getDigitFrequency (int n, int d)}$

```
int counter = 0;
while (n > 0)
{
    int temp = n % 10; gDF ②
    n = n / 10;
    if (temp == d)
        counter++;
}
return counter;
```

FUNCTION NAME
 $f(x, y)$

ARGUMENTS

main ①	temp	33333330
	counter	0X23
	d	6
	n	4X88888
	f	3
	d	6
	n	465663
	scn	

After return counter, all the variables will be destroyed from memory and value of counter=3 will be assigned to f

ORDER OF FLOW $\Rightarrow ① \rightarrow ② \rightarrow ③$

System.out;

FUNCTION CALL

freq(n, d);

;

FUNCTION NAME

it frequency (int n, int d) {

f(x,y)
ARGUMENTS

gDF

②

temp	38688840
counter	0.123
d	6
n	4888883
	██████████
③ f	3
	██████████
d	6
n	465663
schn	

main

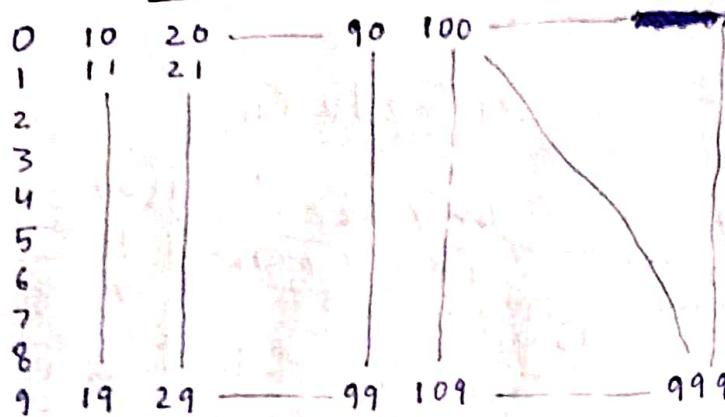
①

After return
counter, all the
variable will be
destroyed from
memory and value
of counter = 3 will be
→ assigned to f.

ORDER OF FLOW \Rightarrow ① \rightarrow ② \rightarrow ③

NUMBER SYSTEM

DECIMAL NUMBER SYSTEM : BASE $\rightarrow 10 \therefore$ DIGIT : $[0, 1, 2, 3, \dots, 9]$

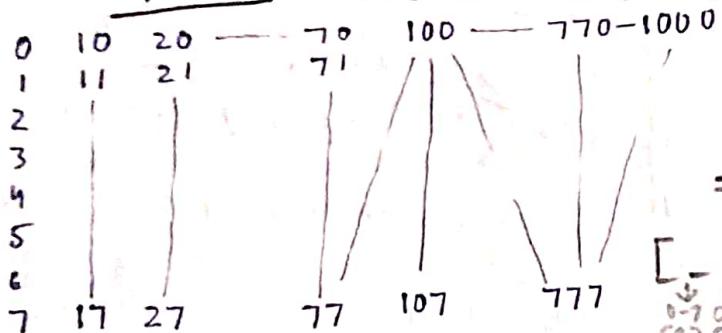


$$[9 \ 5 \ 6 \ 3]_{10} = 9 \times 10^3 + 5 \times 10^2 + 6 \times 10^1 + 3 \times 10^0$$

$$[1 \ 1 \ 1 \ 1]_{10} = 10^4 (10 \cdot 10 \cdot 10 \cdot 10) \\ (0 \cdot 9 \ 0 \cdot 9 \ 0 \cdot 9 \ 0 \cdot 9)_{(10) \ (10) \ (10) \ (10)} \\ (0000 - 9999)$$

SMALLEST LARGEST

OCTAL NUMBER SYSTEM : BASE $\rightarrow 8$



$$(3 \ 4 \ 6) \\ \downarrow \quad \downarrow \quad \downarrow \\ 8^2 \ 8^1 \ 8^0$$

$$= 3 \times 8^2 + 4 \times 8^1 + 6 \times 8^0$$

$$[1 \ 1 \ 1]_8 = 8^4 (8 \cdot 8 \cdot 8 \cdot 8) \\ (0 \cdot 7 \ 0 \cdot 7 \ 0 \cdot 7 \ 0 \cdot 7)_{(8) \ (8) \ (8) \ (8)} \\ (0000 - 7777)$$

SMALLEST LARGEST

BINARY NUMBER SYSTEM:

BASE $\rightarrow 2 \therefore$ DIGIT : $[0, 1]$

$$0-10-100-1000$$

$$\begin{array}{r} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{array}$$

$$[1 \ 1 \ 0 \ 1 \ 1 \ 1]_2 \\ \downarrow \quad \downarrow \quad \downarrow \quad \downarrow \quad \downarrow \quad \downarrow \\ 2^5 \ 2^4 \ 2^3 \ 2^2 \ 2^1 \ 2^0$$

$$= 1 \times 2^5 + 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 2^0 \times 1$$

$$[1 \ 1 \ 1 \ 1] = 2^4 (2 \cdot 2 \cdot 2 \cdot 2)$$

$$(2) \ (2) \ (2) \ (2) \quad (0000 - 1111) \\ \downarrow \quad \downarrow \quad \downarrow \quad \downarrow \\ \text{SMALLEST} \quad \text{LARGEST}$$

CONVERSION

DECIMAL TO OCTAL

$$(765)_{10} \rightarrow [1375]_8$$

= OCTAL ME CONVERT KARE HAI TOH ③ SE
DIVIDE KRENNY.
CROSS-CHECKING

8 765	<u>rem</u>
8 95	5
8 11	7
8 1	3
0 0	1

$$x 10^0 = 5$$

$$x 10^1 = 70$$

$$x 10^2 = 300$$

$$x 10^3 = 1000$$

$$\underline{1375}$$

$$7 \times 10^2 = 700$$

$$6 \times 10^1 = 60$$

$$5 \times 10^0 = 5$$

$$\underline{\underline{765}}$$

$$\boxed{1} = \boxed{2}$$

$$1375 \\ (6, 6, 6, 8) \rightarrow 8^0 \\ 8^3 8^2 8^1$$

$$= 1 \times 8^3 = 512$$

$$= 3 \times 8^2 = 192$$

$$= 7 \times 8^1 = 56$$

$$= 5 \times 8^0 = 5$$

$$\underline{\underline{765}} - 2$$

OCTAL TO DECIMAL

$$(1375)_8 \rightarrow (765)_{10}$$

10 1375	<u>rem</u>
10 137	5
10 13	7
10 1	3
0 0	1

$$x 8^0 = 5$$

$$x 8^1 = 56$$

$$x 8^2 = 192$$

$$x 8^3 = 512$$

$$\underline{\underline{765}}$$

DECIMAL TO BINARY

$$[57]_{10} = [111001]_2$$

2 57	<u>rem</u>
2 28	1
2 14	0
2 7	0
2 3	1
2 1	1
0 0	1

$$x 10^0 = 1$$

$$x 10^1 = 0$$

$$x 10^2 = 0$$

$$x 10^3 = 1000$$

$$x 10^4 = 10000$$

$$x 10^5 = 100000$$

$$\underline{\underline{111001}}$$

DECIMAL TO ANY BASE

$$[634]_{10} = [\quad]_8 ?$$

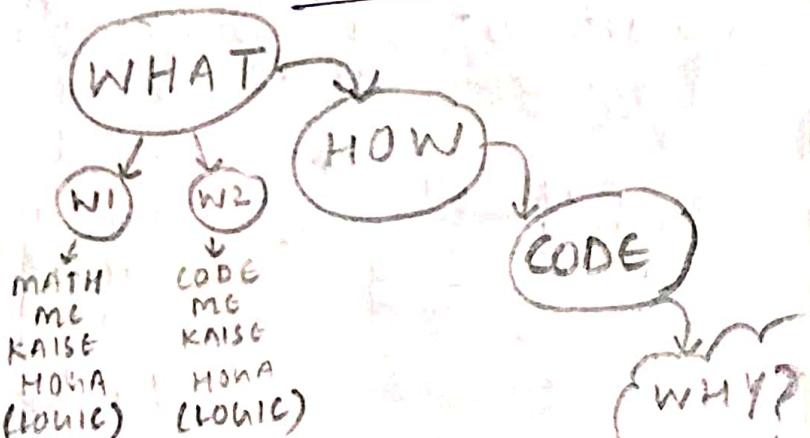
8 634	<u>rem</u>
8 79	2
8 9	7
8 1	1
0 0	1

$$x 10^0 = 2$$

$$x 10^1 = 70$$

$$x 10^2 = 100$$

$$x 10^3 = 1000$$

$$\underline{\underline{1172}}$$


JISME ME JATEY HAI
USSEY DIVIDE KRTY HAI

JISSE JAATEY HAI
MULTIPLY KARTEY HAI

USSE
POWER
ST

NOW WHY?

$$6 \times 10^2 + 3 \times 10^1 + 4 \times 10^0 = 1 \times 8^3 + 1 \times 8^2 + 7 \times 8^1 + 2$$

$$600 + 30 + 4 = 512 + 64 + 56 + 2$$

$$634 = 634$$

10 KE GUTCHY

8 KE GUTCHY

8 634	<u>rem</u>
8 79	2
8 9	7
1 1	1

$$\therefore 634 = 8 \cdot 79 + 2 = 8^3 \cdot 1 + 8^2 \cdot 1 + 8^1 \cdot 7 + 2$$

$$\therefore 79 = 8 \cdot 9 + 7 = 8^2 \cdot 1 + 8 \cdot 1 + 7$$

$$\therefore 9 = 8 \cdot 1 + 1$$

```

public static void main(String[] args)
{
    Scanner s = new Scanner(System.in);
    int n = s.nextInt(); // n = decimal
    int b = s.nextInt(); // b = base in which
    // we have to convert
    int dn = getValueInBase(n, b);
    System.out.println(dn);
}

```

```

public static int getValueInBase(int n, int b)
{
    int tp = 1; //  $10^0 = 1$ 
    int xv = 0;
    while (n > 0)
    {
        int dig = n % b; // rem. niklega
        xv += (dig * tp); // xv = xv + dig * p
        tp *= 10; // p = p * 10;
        n = n / b; // no. chota hogा
    }
    return xv; // FINAL NUMBER
}

```

ANY BASE TO DECIMAL

```

public static int getValueInDecimal(int n, int b)
{
    int bp = 1; // base power
    int xv = 0; // remain value
    while (n > 0)
    {
        int dig = n % 10; // decimal me jana hai toh
        // 10% aur / karna hai remainder ke liye
        xv = xv + dig * bp;
        bp = bp * b;
        n = n / 10; // no. chota hogा
    }
    return xv;
}

```

$$1 \times 8^3 + 1 \times 8^2 + 7 \times 8^1 + 2 \times 8^0 = \underbrace{6 \times 10^2 + 3 \times 10^1 + 4 \times 10^0}_{10 \text{ के गुणात्मक}} = 634$$

8 के गुणात्मक

$$634 = 634$$

DRY RUN

$$(784)_{10} = (?)_8$$

$$\begin{array}{r} n=784 \\ b=8 \end{array}$$

b) n	p	→ 8V
3 784	rem	$\xrightarrow{+10^0} 0$
8 98	0	$\xrightarrow{+10^1} 20$
3 12	2	$\xrightarrow{+10^2} 400$
3 1	4	$\xrightarrow{+10^3} 1000$
	0	$+1420$

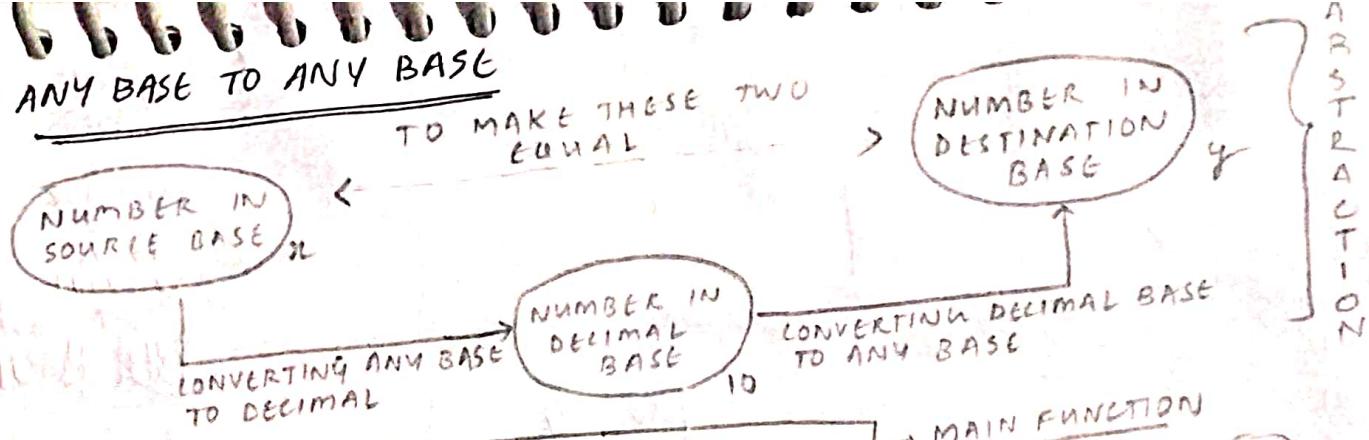
DRY RUN

$$[1172]_3 = [634]_{10}$$

n = 1172	b = 10
10 1172	rem * p → 2
10 117	$2 \times 10^0 \xrightarrow{+} 2$
10 11	$7 \times 10^1 \xrightarrow{+} 70$
10 1	$1 \times 10^2 \xrightarrow{+} 100$
0	$1 \times 10^3 \xrightarrow{+} 512$

$$512 + 70 + 2 = 584$$

ANY BASE TO ANY BASE



```
public static void main(String []args)
{ Scanner s = new Scanner(System.in);
  int n = s.nextInt(); → n = 784
  int sourceBase = s.nextInt(); → x = 8
  int destBase = s.nextInt(); → y = 2
  int value = getValue(n, sourceBase, destBase);
  System.out.println(value); → 1000000100
```

GETVALUE
FUNCTION
FOR STORING VALUES
FROM OTHER FUNCTION'S
FOR FINAL OUTPUT

```
000100  
public static int getValue (int n, int src, int dest)  
{ int decimalValue = anyBaseToDecimal (n, src);  
    int finalBaseValue = decimalToAnyBase (decimalValue, dest);  
    return finalBaseValue;  
}
```

```

public static int anyBaseToDecimal (int n, int b)
{
    int rv = 0; // return value
    int bp = 1; // base power ( $b^0 = 1$ )
    while (n > 0)
    {
        int dig = n % 10; // remainder ayega
        n = n / 10; // no. chata hogi
        rv = rv + dig * bp;
        bp = bp * b; // base power + hogi
    }
    return rv;
}

```

ANY BASE TO DECIMAL FUNCTION

DECIMAL TO ANY BASE FUNCTION

<u>2</u>	<u>516</u>	$2 \text{cm} \times p$	\rightarrow	<u>21</u>
<u>2</u>	<u>258</u>	0×10^0	\rightarrow	<u>0</u>
<u>2</u>	<u>129</u>	0×10^1	\rightarrow	<u>0</u>
<u>2</u>	<u>64</u>	1×10^2	\rightarrow	<u>100</u>
<u>2</u>	<u>32</u>	0×10^3	\rightarrow	<u>0</u>
<u>2</u>	<u>16</u>	0×10^4	\rightarrow	<u>0</u>
<u>2</u>	<u>8</u>	0×10^5	\rightarrow	<u>0</u>
<u>2</u>	<u>4</u>	0×10^6	\rightarrow	<u>0</u>
<u>2</u>	<u>2</u>	0×10^7	\rightarrow	<u>0</u>
<u>2</u>	<u>1</u>	0×10^8	\rightarrow	<u>0</u>
<u>2</u>	<u>0</u>	0×10^9	\rightarrow	<u>1×10^9</u>

```

public static int decimalToAnyBase(int n, int b)
{
    int rv = 0; // return value
    int tp = 1; // 10 base power => (100 = 1)
    while (n > 0)
    {
        int dig = n % b; // remainder ayega
        n = n / b; // no. chota hogा
        rv = rv + dig * tp;
        tp = tp * 10; // 10 power + hogi
    }
    return rv;
}

```

(iv) \Rightarrow 1000000100

ANY BASE ADDITION

$$\begin{array}{r} \textcircled{1} \textcircled{1} \textcircled{1} \\ (7 \ 5 \ 4)_9 \\ + (3 \ 4 \ 7)_9 \\ \hline 1 \ 3 \ 2 \ 3 \end{array}$$

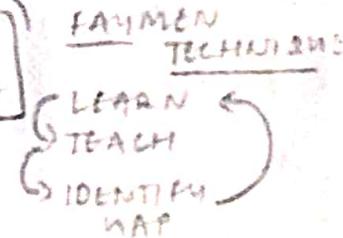
$$\begin{array}{r} \textcircled{1} \textcircled{1} \textcircled{1} \\ (6 \ 9 \ 5)_{10} \\ + (4 \ 8 \ 9)_{10} \\ \hline 1 \ 1 \ 8 \ 4 \end{array}$$

$$\left\{ \begin{array}{l} 14 < \frac{14}{10} = 1 \\ \therefore 14 = 10 \cdot 1 + 4 \\ 13 < \frac{13}{10} = 1 \\ \therefore 13 = 10 \cdot 1 + 3 \\ 11 < \frac{11}{10} = 1 \\ \therefore 11 = 10 \cdot 1 + 1 \end{array} \right.$$

$$11 = 8 + 3 \\ = 8 \cdot 1 + 3$$

$$10 = 8 + 2 \\ = 8 \cdot 1 + 2$$

$$11 = 8 + 3 \\ = 8 \cdot 1 + 3$$



```
public static int getSum(int b, int n1, int n2), PASSED IN MAIN FUNC
{ int sum = 0;
  int c = 0; // carry
  int tp = 1; // 10^i
  while(n1 > 0 || n2 > 0 || c > 0)
  {
    int d1 = n1 % 10; // rem
    int d2 = n2 % 10; // rem
    n1 = n1 / 10; // n1
    n2 = n2 / 10; // n2
    int d = d1 + d2 + c;
    c = d / b; // carry
    d = d % b;
    sum = sum + d * tp;
    tp = tp * 10;
  }
  return sum;
}
```

$$\begin{aligned} n_1 &= 567 \\ n_2 &= 435 \\ b &= 8 \end{aligned}$$

$$77(d) \rightarrow 1001101$$

$$77(10)$$

$$777(0) \rightarrow 11111$$

$$63(10) \leftarrow 1111$$

$$d = d_1 + d_2 + c \quad d \% b$$

	n ₁	d ₁	n ₂	d ₂	c	d
1	10 567	① 0	10 435	② 3	③	7 + 543 = 12, 1, 4 × 10 ⁱ
2	10 56	① 7	10 43	② 4	③	6 + 3 + 1 = 10, 1, 2 × 10 ⁱ
3	10 5	① 6	10 4	② 0	④	5 + 4 + 1 = 10, 1, 2 × 10 ⁱ
4	10 0	① 5	10 0	② 0	⑤	0 + 0 + 1 = 1, 0, 1 × 10 ⁱ
		0	0	0	⑥	

$$\text{sum} = 1224$$

$$\begin{aligned} n_1 &< \frac{n_1}{10} = n_1 \\ n_2 &< \frac{n_2}{10} = n_2 \end{aligned}$$

$$\begin{aligned} d &= b \cdot D + \text{rem} \\ &\text{carry} \end{aligned}$$

$$\begin{aligned} tp &\Rightarrow 10^0 \\ &10^1 \\ &10^2 \\ &10^3 \end{aligned}$$

I/P EQUATION
O/P
TREAT AS OCTAL
ADD THEM AS OUNI

ALGORITHMIC DESIGN OF FUNCTION

"int getSum(int b, int n1, int n2)"

declare integers sum=0;
done=1;

ALGORITHMIC DESIGN OF FUNCTION

"int getSum(int b, int n1, int n2)"

```
declare integers sum=0;  
power=1,  
carry=0;
```

if $n1 > 0$ or $n2 > 0$ or carry $\neq 0$

FALSE

```
return  
sum
```

TRUE

```
declare integers digit1 = n1%10;  
digit2 = n2%10;  
digit = digit1 + digit2 + carry;  
update n1 = n1/10;  
n2 = n2/10;  
carry = digit/b;  
digit = digit % b;  
sum = sum + digit * power;  
power = power * 10;
```

ANY BASE SUBTRACTION

$$\begin{array}{r}
 1212 \\
 -256 \\
 \hline
 734
 \end{array}$$

$$m_1 = 1212 \\ m_2 = 256$$

		Quotient - Remainder		
10	1212	256	, carry, difference	
10	121	-256	, -1, $10-6=4 \rightarrow \times 10^1$	
10	12	-0	, $10-5=5 \rightarrow \times 10^2$	+ 734
10	1	-0	, $10-3=7 \rightarrow \times 10^3$	
10	0	-0	, $10-2=8 \rightarrow \times 10^4$	
10	0	-0	, $0-0=0$	

$m_1 \rightarrow$ Subtrahend
 $m_2 \rightarrow$ Minuend

public static int getDifference (int b, int n1, int n2)

{ int diff = 0;

int p = 1;

int c = 0;

while ($n_2 > 0$) \rightarrow Tab Tak chalane hai jis tak minuend zero na bane!

{ int d1 = $n_1 \% 10$; } // Remainder Extraction

int d2 = $n_2 \% 10$;

$n_1 = n_1 / 10$;

$n_2 = n_2 / 10$;

int d = 0;

\rightarrow To bhi carry/borrow hai use sethle kروا!

\rightarrow $d_2 = d_2 + c$; // Agar minuend chota hai substrahend se!

if ($d_2 \geq d_1$) // Agar minuend bada hai substrahend se!

{ $c = 0$;
 $d = d_2 - d_1$;} \leftarrow next iteration sethle hoga

else // Agar minuend chota hai substrahend se!

{ $c = -1$;
 $d = d_2 + b - d_1$;} \leftarrow next iteration sethle hoga

} \rightarrow no. banega yaha

diff = diff + d * p; // power badlegi yaha!

p = p + 10; // power badlegi yaha!

} return diff;

}

ALGORITHMIC DESIGN OF FUNCTION

"int getDifference (int b, int n1, int n2)"

declare integer diff = 0,
power = 1;
carry = 0

\rightarrow diff = diff + (digit * power);

NON

Auswertung - Lösungsweg

$$\begin{array}{r} 10 \\ 10 \end{array} \quad | \quad 12 \text{ } 1 \text{ } 2 \quad , \quad 2 \text{ } 5 \text{ } 6 \quad , \quad \text{carry}, \text{ difference}$$

$$\begin{array}{r} 10 \\ 10 \end{array} \quad | \quad 12 \text{ } 1 \text{ } -\textcircled{2}, \quad 2 \text{ } 5 -\textcircled{6}, \quad -1, \quad 10 - 6 = 4 \rightarrow \times 10^1 \\ \textcircled{10} \quad + 8 \text{ (über)} \end{math>$$

$$\begin{array}{r} 10 \\ 10 \end{array} \quad | \quad 12 \text{ } -\textcircled{4}, \quad 2 \text{ } -\textcircled{5}, \quad -1, \quad 8 - 5 = 3 \rightarrow \times 10^2 \\ \textcircled{8} \text{ (über)} \end{math>$$

$$\begin{array}{r} 10 \\ 10 \end{array} \quad | \quad 1 \text{ } -\textcircled{2}, \quad 0 \text{ } -\textcircled{2}, \quad -1, \quad 9 - 2 = 7 \rightarrow \times 10^3 \\ \textcircled{9} \text{ (über)} \end{math>$$

$$0 - \textcircled{0} \quad , \quad 0 - \textcircled{0} \quad , \quad 0 - 0 = 0$$

hence (int b, int n1, int n2)

$n_1 \Rightarrow$ Subtrahend
 $n_2 \Rightarrow$ Minuend

(n_2)

Tab Tak Chatana Mai ist tick minder
 aber es funktioniert!

ANY-BASE MULTIPLICATION

$$\begin{array}{r}
 & 3 & 3 & \leftarrow \\
 2 & 2 & 3 & 3 \\
 & 2 & 3 & 4 \\
 \times & 7 & 6 & \\
 \hline
 14/11 & 21/24 & 24/28 \\
 1 & 6\textcircled{1} & 5 & 0\leftarrow \\
 & 0 & 4 & \times \\
 \hline
 227 & 1 & 0 &
 \end{array}$$

BASE $\rightarrow 8$

$$6 \times 4 \Rightarrow 24 = 8^1 \cdot 3 + 0$$

QUOTIENT WILL GO AS CARRY.

REMAINDER WOULD BE LEAVE BELOW

$$(6 \times 3) + 3 \Rightarrow 21 = 8^1 \cdot 2 + 5$$

$$(6 \times 2) + 2 \Rightarrow 14 = 8^1 \cdot 1 + 6$$

$$(7 \times 4) \Rightarrow 28 = 8^1 \cdot 3 + 4$$

$$(7 \times 3) + 3 \Rightarrow 24 = 8^1 \cdot 3 + 0$$

$$(7 \times 2) + 3 \Rightarrow 17 = 8^1 \cdot 2 + 1$$

$$(5 + 4) \Rightarrow 9 = 8^1 \cdot 1 + 1$$

PIECE BY PIECE

SOLN

n_2 ki pheli digit ko n_1 ki sabhi digits se multiply karayey.

$$\begin{array}{r}
 234 \rightarrow n_1 \\
 \times 76 \rightarrow n_2
 \end{array}$$

$$\begin{array}{r}
 234 \rightarrow n_1 \\
 \times 76 \rightarrow n_2
 \end{array}$$

then, n_2 ki dusri digit ko n_1 ki sabhi digits se multiply karayey.

\therefore Fir, Dene Ko ADD Kardo!

Multiply n_1 with digit one of n_2

Multiply n_1 with digit two of n_2

Multiply n_1 with digit three of n_2

Multiplying with n_2

+

NET PRODUCT WITH A
SINGLE DIGIT

ANYBASE
MULTIPLICATION

NET PRODUCT

①

GET SUM

②

```

public static int getProductWithASingleDigit(int b, int n1, int d2)
{
    int rv = 0;
    int p = 1;
    int c = 0;
    while (n1 > 0 || c > 0)
    {
        int d1 = n1 % 10;
        n1 = n1 / 10;
        int d = d1 * d2 + c;
        c = d / b;
        d = d % b;
        rv = rv + d * p;
        p = p * 10;
    }
    return rv;
}

```

```

graph TD
    A[declare  
rv=0;  
p=1;  
c=0;] --> B{while  
(n1>0 || c>0)}
    B --> C[d1=n1%10;  
n1=n1/10;]
    C --> D[d=d1*d2+c;]
    D --> E[c=d/b;]
    E --> F[d=d%b;]
    F --> G[rv=rv+d*p;]
    G --> H[p=p*10;]
    H --> I[set n1=n1/10;]
    I --> B
    B --> J{return rv;}

```

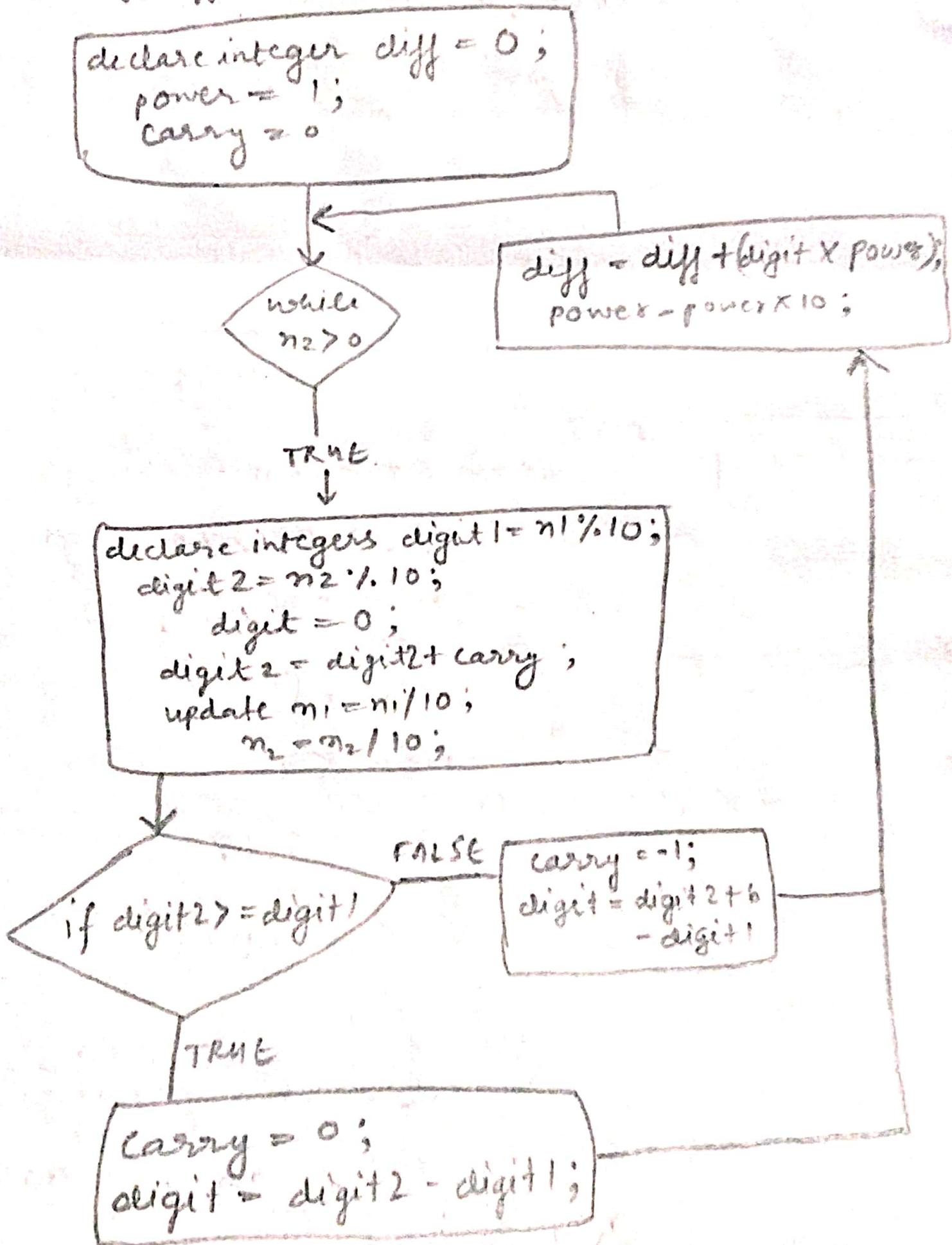
```

public static int getSum (int b, int n1, int n2)
{
    int rv = 0;
    int p = 1;
    int c = 0;
    while (n1 > 0 || n2 > 0 || c > 0)
    {
        int d1 = n1 % 10;
        int d2 = n2 % 10;
        n1 = n1 / 10;
        n2 = n2 / 10;
        int d = d1 + d2 + c;
        c = d / b;
        d = d % b;
        rv = rv + d * p;
        p = p * 10;
    }
    return rv;
}

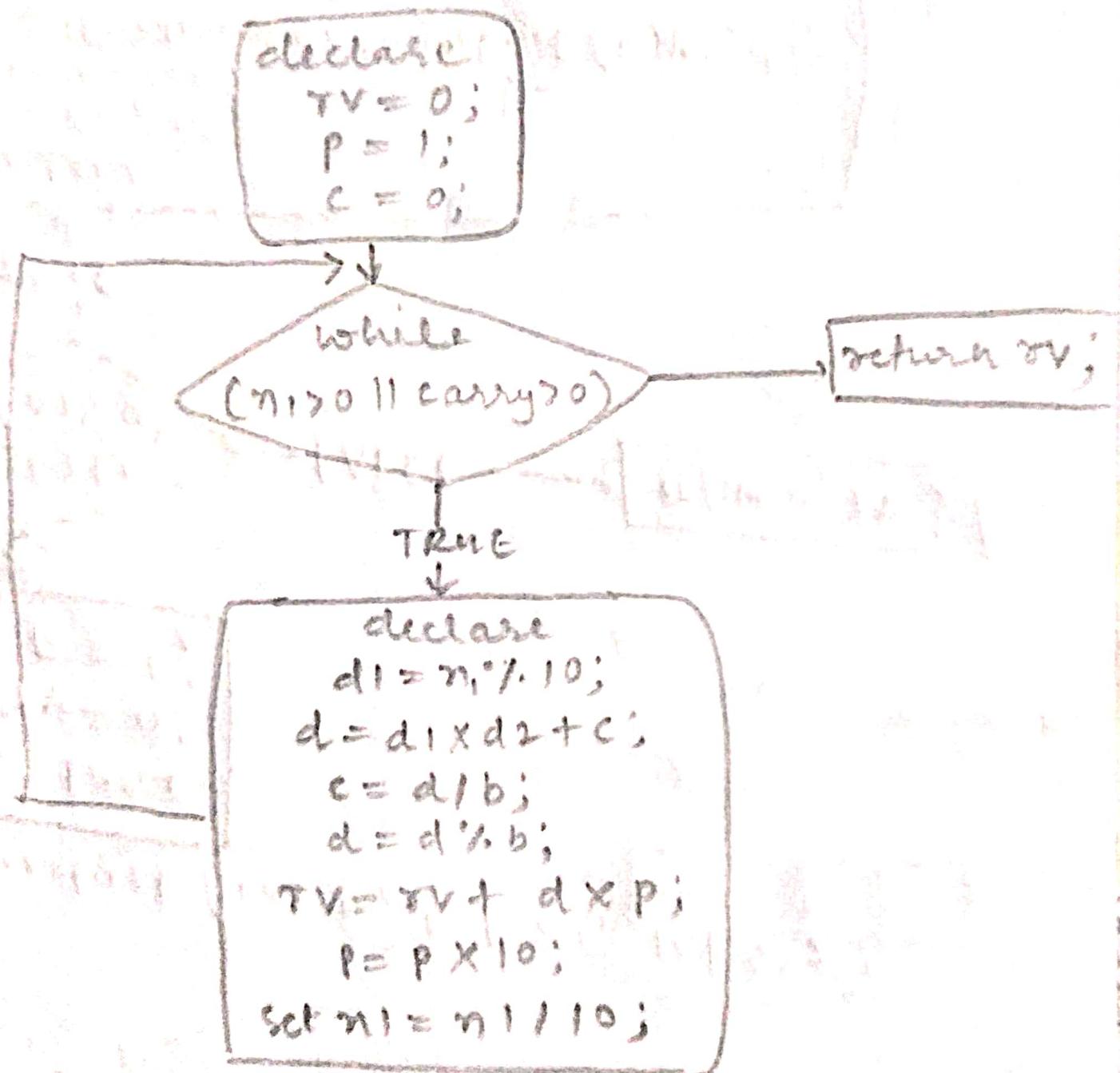
```

ALGORITHMIC DESIGN OF FUNCTION

"int getDifference(int b, int m, int n2)"



getProductWithASingleDigit(int b, int n1, int d2)



```

public static int getProdud (int b, int n1, int n2) {
    int rv = 0;
    int p = 1;
    while (n2 > 0)
    {
        int d2 = n2 % 10;
        n2 = n2 / 10;
        int sprd = getProductwithASingleDigit (b, n1, d2);
        sprd = sprd * p;
        p = p * 10;
        rv = getSum (b, rv, sprd);
    }
    return rv;
}

```

```

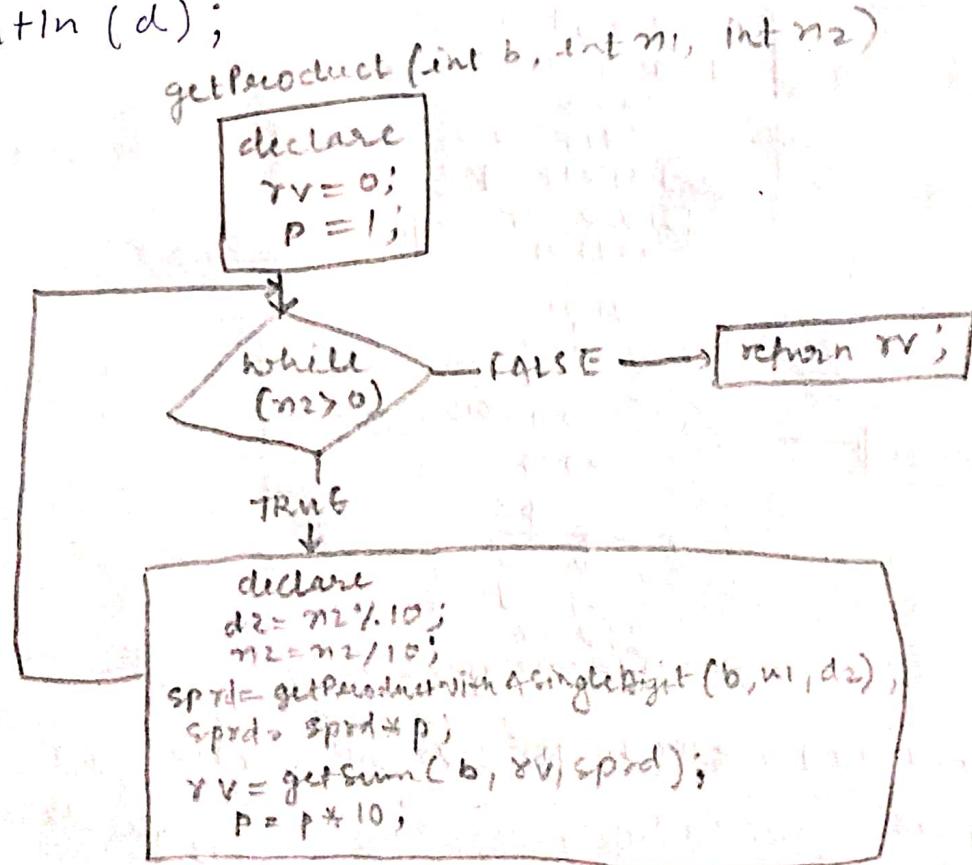
public static void main (String [] args)
{

```

```

    Scanner s = new Scanner (System.in);
    int b = s.nextInt();
    int n1 = s.nextInt();
    int n2 = s.nextInt();
    int d = getProdud (b, n1, n2);
    System.out.println (d);
}

```



ARRAYS (INTRODUCTION)

```

int m = 50;
int m1;
int m2;
int m3;
.
.
.
int marks[5];
    
```

→ int [] marks ;
marks = new int [5];
marks [0] = 51
————— [1] = 33
————— [2] = 21
————— [3] = 67
————— [4] = 89

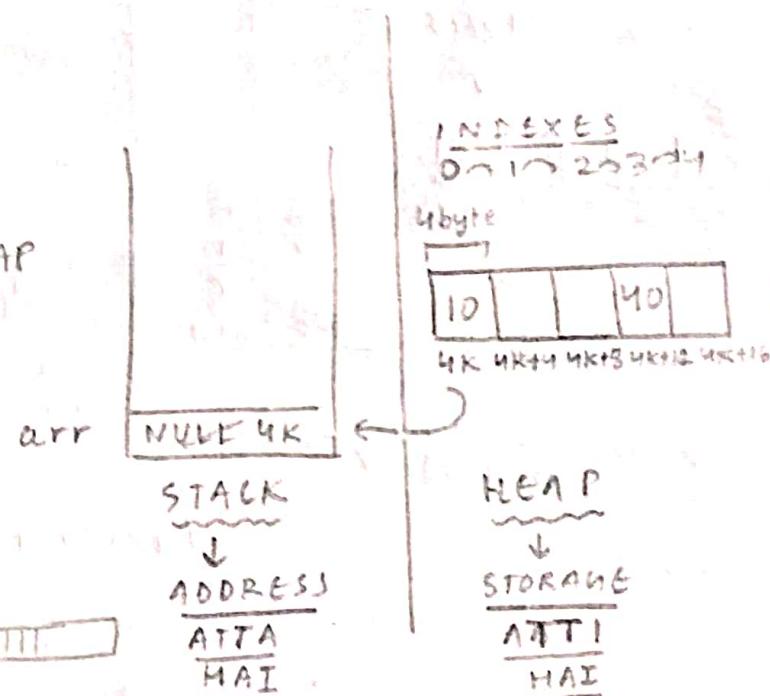
for(int i=0; i<marks.length; i++)
{
 System.out.println(marks[i]);
}

int [] arr; → DECLARATION
arr = new int [2]; → ASSIGNMENT
arr [0] = 33;
arr [1] = 47;

int [] arr = new int [2];

ARRAY MEMORY

int [] arr; → ALLOCATED IN STACK
arr = new int [5]; → ALLOCATED IN HEAP
arr [0] = 10; [4K[0]]
arr [3] = 40; [4K[3]]



IMPLICATIONS

① PERFORMANCE

int [] arr = new int [1000];
arr [500] = 10; 4K[500]
arr [700] = 20; 4K[700]

$$4K[500] = 4K + 2K = 6K$$

X → 2000 →

$$4K[700] = 4K + 2800 = 6800$$

X → 2800 →

* ARRAY KA SIZE JITNA BHI HO,
DONO KE LIYE SAME TIME LAHEGA!
KU?

↳ KUKI → AS FAR AS JAVA IS CONCERN HUMANE
USKO TOH ADDRESS BATA DIA HAI!

② ASSIGNMENT

```
int [] one = new int [3];
```

```
one [0] = 10;
```

```
one [1] = 20;
```

```
one [2] = 30;
```

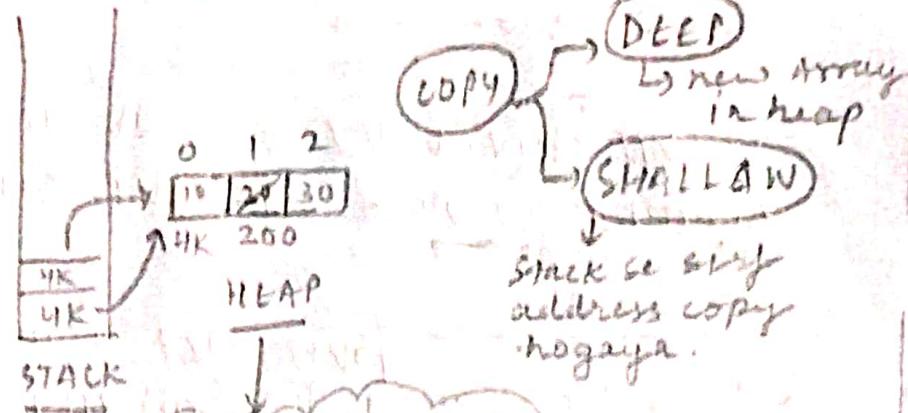
```
int [] two = one;
```

```
two [1] = 200;
```

QUESTION

```
① for (i=0; i<one.length; i++)
{ System.out.println(one[i]);
}
```

```
② for (i=0; i<two.length; i++)
{ System.out.println(two[i]);
}
```



Help me jab bhi
naya kuch banega
to hamisha **new**
word se start hogega

③ FUNCTIONS

```
public static void swap (int [] arr, int i, int j)
```

```
{ int temp = arr [i];
arr [i] = arr [j];
arr [j] = temp;
}
```

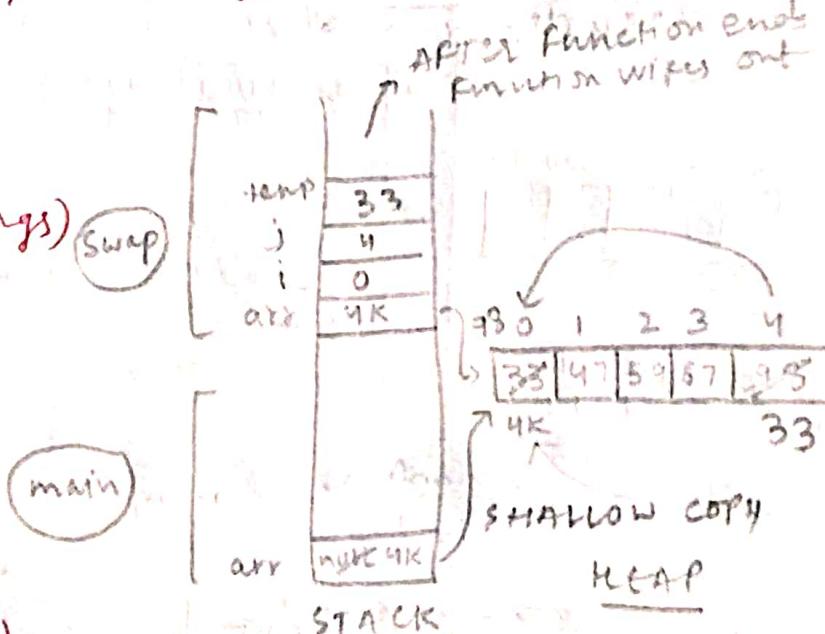
```
public static void main (String [] args)
```

```
{ int arr = new int [5];
arr [0] = 33;
arr [1] = 47;
arr [2] = 59;
arr [3] = 67;
arr [4] = 98;
}
```

```
Swap (arr, 0, 4);
```

```
for (int i=0; i<arr.length; i++)
{ System.out.println(arr [i]);
}
```

```
}
```



SPAN OF ARRAY

```

public static void main (String [] args) {
    Scanner s = new Scanner (System.in);
    int n = s.nextInt();
    int [] arr = new int [n];
    for (int i = 0; i < arr.length; i++)
        arr[i] = s.nextInt();
    int max = arr[0], min = arr[0];
    for (int i = 1; i < arr.length; i++)
        if (arr[i] > max) { // CONDITION
            max = arr[i];    FOR
            MAX. NO.          |
        }
        if (arr[i] < min) { // CONDITION
            min = arr[i]; FOR
            MIN. NO.          |
        }
    }
    int span = max - min; // SPAN
    System.out.println(span);
}

```

ARRAY
INPUT

$n = 6$

0	1	2	3	4	5
2	11	19	90	3	1

$\max = 2 \cancel{11} \cancel{19} 90$
 $\min = 2 \cancel{11} \cancel{19}$
 $\text{span} = 90 - 1$
 $= \underline{\underline{89}}$

FIND ELEMENT IN ARRAY

```

public static void main (String [] args) {
    Scanner s = new Scanner (System.in);
    int n = s.nextInt();
    int [] arr = new int [n];
    for (int i = 0; i < arr.length; i++)
        arr[i] = s.nextInt();
    int data = s.nextInt(); int idx = -1;
    for (int i = 0; i < arr.length; i++)
        if (arr[i] == data) {
            idx = i;
            break;
        }
    System.out.println (idx);
}

```

ARRAY
INPUT

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

$\underline{\underline{\text{id}}} \underline{\underline{x}} = -1$
 $\underline{\underline{\text{data}}} = 40$

loop $i = 0$

i	$\text{arr}[i] == \text{data}$	$\underline{\underline{\text{id}}} \underline{\underline{x}}$
0	false ($15 == 40$)	-1
1	false ($30 == 40$)	-1
2	true ($40 == 40$)	2 set $\underline{\underline{\text{id}}} \underline{\underline{x}} = i = 2$ break;

EXIT LOOP

BAR CHART

FLOOR	1	2	3	4	5
7	-	-	*	-	
6	-	-	*	-	
5	-	-	*	*	
4	-	-	*	*	
3	*	-	*	*	
2	*	-	*	*	*
1	*	*	*	*	*

i → 1 2 3 4 5

arr [3] [1] [0] [7] [5]

```

public static void main (String [] args) {
    Scanner s = new Scanner (System.in);
    int n = s.nextInt();
    int [] arr = new int [n];
    for (int i=0; i<arr.length; i++) {
        arr[i] = s.nextInt();
    }
    int max = arr[0];
    for (int i=1; i<arr.length; i++) {
        if (arr[i] > max) {
            max = arr[i];
        }
    }
}
    
```

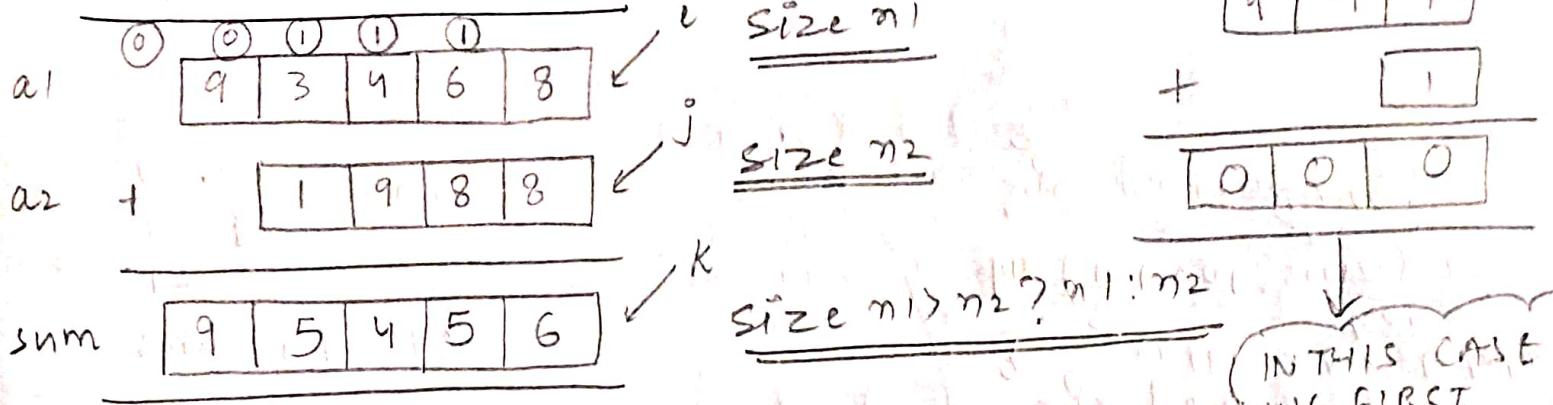
] ARRAY INPUT
] MAX FOUND
] KRNA

YEH LOOP FLOOR KE LIYE CHALEGA!
 FLOOR KO MAX KE FORMAAL RAKHENGY!
 YEH LOOP STARS AUR SPACE PRINT KARNE KE LIYE CHALEGA

```

for (int floor = max; floor >= 1; floor--) {
    for (int i=0; i<arr.length; i++) {
        if (arr[i] >= floor) {
            System.out.print ("*");
        } else {
            System.out.print ("H");
        }
        System.out.println();
    }
}
    
```

② SUM OF TWO ARRAYS

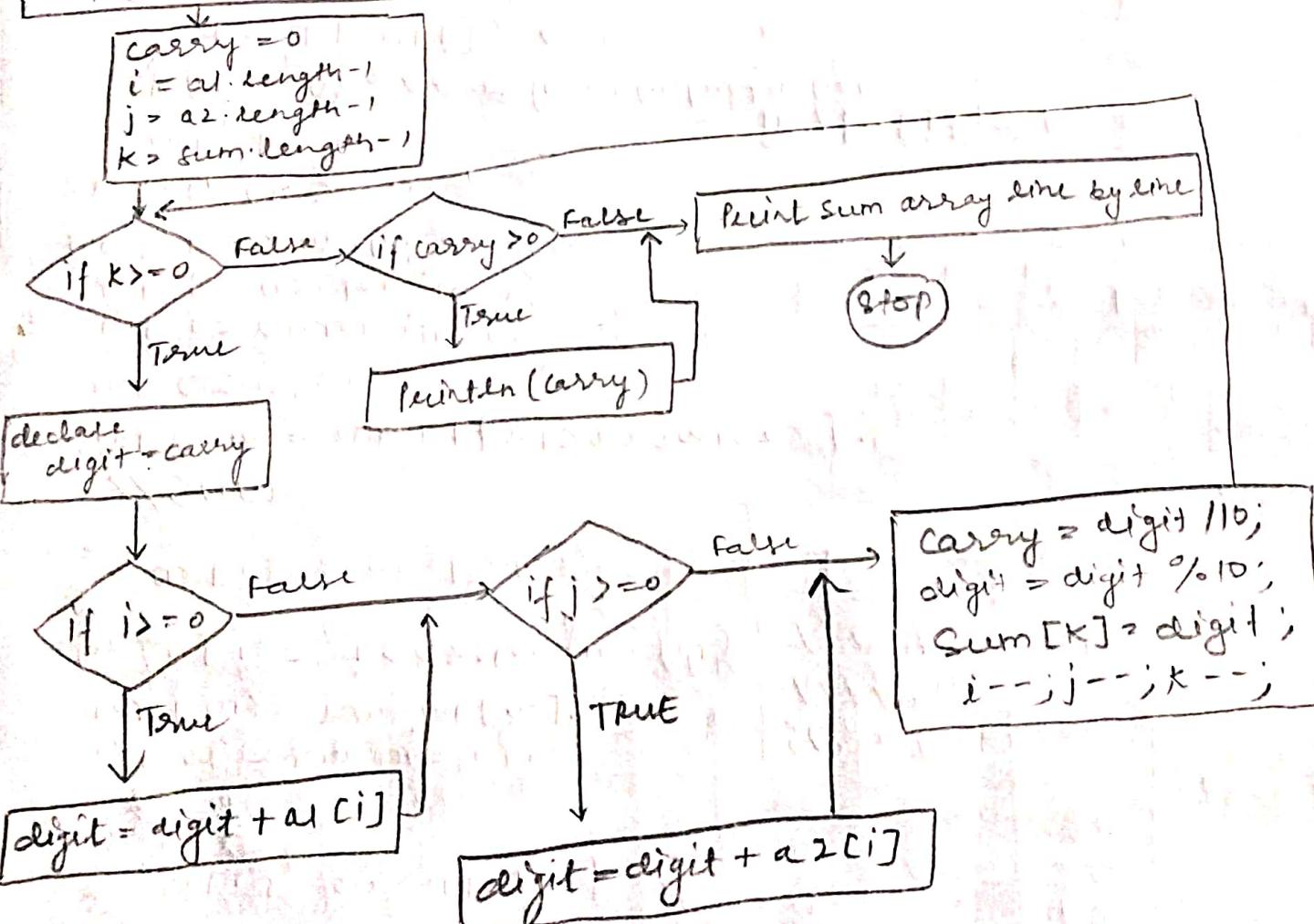


Start

declare and input two arrays required:
let array 1 be a_1 with size n_1
let array 2 be a_2 with size n_2

declare array sum with size greater among
 n_1, n_2

IN THIS CASE
WE FIRST
PRINT THE
LEFTOVER
CARRY AND
THEN SUM OF
ARRAY CONTENT



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

```
{ Scanner s = new Scanner (System.in);
```

```
int n1 = s.nextInt();
```

```
int [] a1 = new int [n1];
```

```
for (int i = 0; i < a1.length; i++)
```

```
{ a1[i] = s.nextInt();
```

```
}
```

FIRST

ARRAY

READY

SECOND

ARRAY

READY

```
int n2 = s.nextInt();
```

```
int [] a2 = new int [n2];
```

```
for (int i = 0; i < a2.length; i++)
```

```
{ a2[i] = s.nextInt();
```

```
}
```

// Sum

```
int [] sum = new int [n1 > n2 ? n1 : n2];
```

```
int c = 0;
```

```
int i = a1.length - 1;
```

```
int j = a2.length - 1;
```

```
int k = sum.length - 1;
```

```
while (k >= 0) {
```

```
    int d = c;
```

```
    if (i >= 0) { // Agar (i) abhi bacha hai
```

```
        d = d + a1[i]; ←
```

```
}
```

```
    if (j >= 0) { // Agar (j) abhi bacha hai
```

```
        d = d + a2[j]; ←
```

```
}
```

```
    c = d / 10; Quotient → carry
```

```
    d = d % 10; remainder → digit
```

```
sum[k] = d;
```

```
i--;
```

```
j--;
```

```
k--;
```

```
}
```

Agar (c) is non-zero
+ek carry thi
P print hoga

if (c > 0) {

System.out.println(c);

} *sikhा value dedeta
hai*

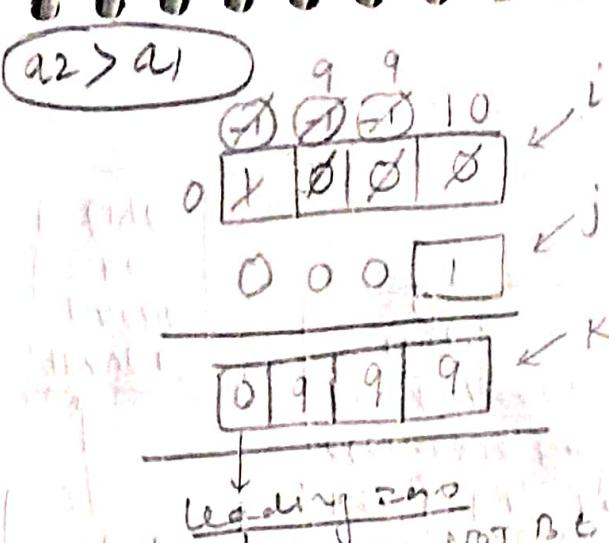
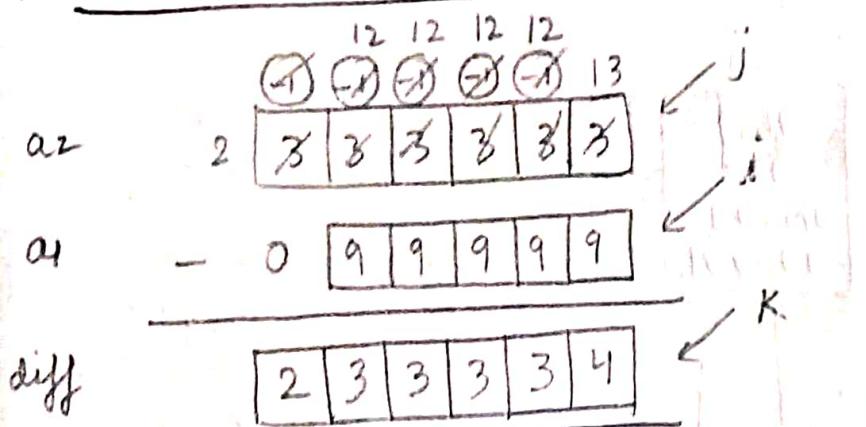
for (int val : sum) {

System.out.println(val);

}

}

DIFFERENCE OF TWO ARRAY



```

public static void main (String [] args) {
    Scanner s = new Scanner (System.in);
    int n1 = s.nextInt ();
    int [] a1 = new int [n1];
    for (int i = 0; i < a1.length; i++) {
        a1[i] = s.nextInt ();
    }
    int n2 = s.nextInt ();
    int [] a2 = new int [n2];
    for (int i = 0; i < a2.length; i++) {
        a2[i] = s.nextInt ();
    }
    int [] diff = new int [n2];
    int c = 0;
    int i = a1.length - 1;
    int j = a2.length - 1;
    int k = diff.length - 1;
    while (k >= 0) {
        if (a2[j] + c >= a1[i]) {
            d = a2[j] + c - a1[i];
            c = 0;
        } else {
            d = a2[j] + c + 10 - a1[i];
            c = -1;
        }
        diff[k] = d;
        i--;
        j--;
        k--;
    }
}

```

leading zero removal code

```

while (k >= 0) {
    int d = 0;
    int alv = i >= 0 ? a1[i] : 0;
    if (a2[j] + c >= alv) {
        d = a2[j] + c - alv;
        c = 0;
    } else {
        d = a2[j] + c + 10 - alv;
        c = -1;
    }
    diff[k] = d;
    i--;
    j--;
    k--;
}
int idx = 0;
while (idx < diff.length) {
    if (diff[idx] == 0) {
        idx++;
    } else {
        break;
    }
}
while (idx < diff.length) {
    System.out.println (diff[idx]);
    idx++;
}

```

③ REVERSE AN ARRAY

public static void reverse(int [] a)

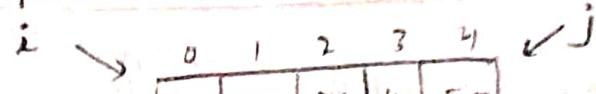
```
{ int i=0;
  int j=a.length-1;
  while(i < j) {
    int temp = a[i];
    a[i] = a[j];
    a[j] = temp;
  }
}
```

i++;

j--;

}

}



i	j	a[i]	a[j]	temp
0	4	50	10	10
1	3	40	20	20
2	2	30	30	30
3	1			

LOOP ENDS
as $i > j$

AFTER SWAPPING

50	40	30	20	10
----	----	----	----	----

④ INVERSE OF AN ARRAY

public static int [] inverse (int [] a)

```
{ int [] inv = new int [a.length];
  for (int i=0; i < a.length; i++)
    { int v = a[i];
      inv[v] = i;
    }
  return inv;
}
```

0	1	2	3	4
3	4	1	2	0

0	1	2	3	4
4	2	3	0	1

values

⑤ SUBARRAY PROBLEM

I/P: a b c d

O/P: a

 a b

 a b c

 a b c d

 b

 b c

 b c d

 c

 c d

 d

i ka loop

j ka loop

k ka loop

TRAVEL KARNE KO LIYE

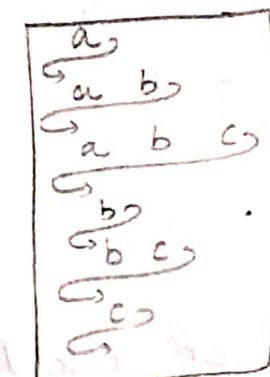
TRAVEL KARNE KO LIYE

i se j tak ke elements ko print karne ke liye.

```

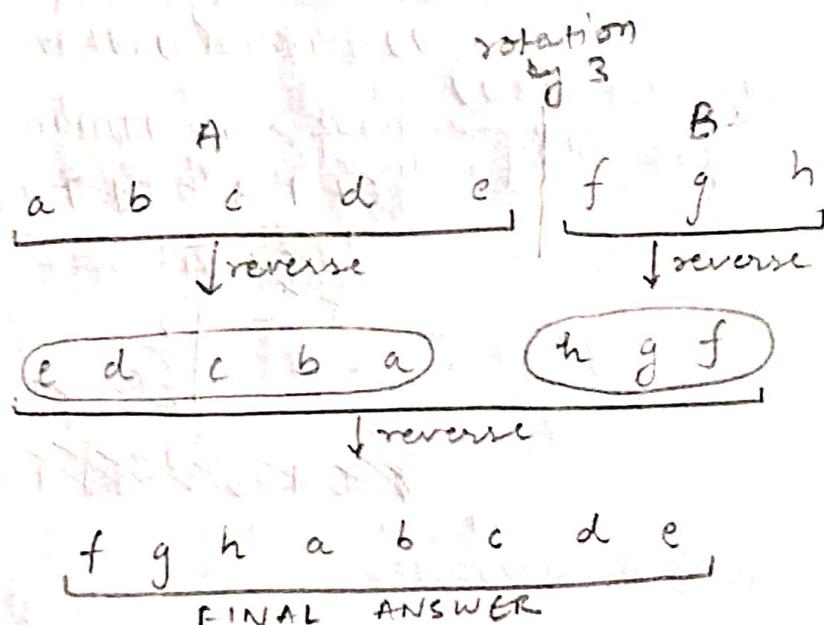
public static void main (String [] args) throws Exception {
    Scanner s = new Scanner (System.in);
    int n = s.nextInt();
    int [] arr = new int [n];
    for (int i=0; i<arr.length; i++)
        arr[i] = s.nextInt();
    for (int i=0; i<arr.length; i++)
        for (int j=i; j<arr.length; j++)
            for (int k=i; k<=j; k++)
                System.out.print (arr[k] + " ");
            System.out.println();
}

```



ROTATE AN ARRAY

	-1	b	c	d	e	a
K	0	a	b	c	d	e
	1	c	a	b	c	d
	2	d	e	a	b	c
	3	c	d	e	a	b
	4	b	c	d	e	a
	5	same as 0				
	6	c	a	b	c	d



MULTIPLE OF K ROTATIONS

Ex: 101

WE WILL USE

$$K = K \% .a.length$$

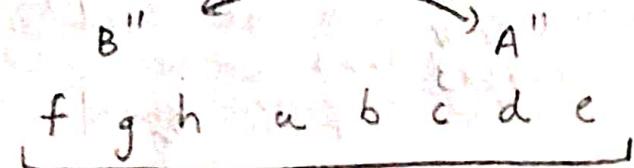
NEGATIVE ROTATIONS

If ($K < 0$)

$$K = K + a.length$$

(-1)

SAME



a₀ b₁ c₂ d₃ e₄ f₅ g₆ h₇

A = 0-4 [0 to a.length - k-1]

B = 5-7 [a.length - k to a.length - 1]

public static void reverse (int [] a, int i, int j) {

```
    int li = i;
    int ri = j;
    while (li < ri) {
        int temp = a[li];
        a[li] = a[ri];
        a[ri] = temp;
        li++;
        ri--;
    }
}
```

REVERSE
FUNCTION

ROTATION : +3
ARRAY
Length : 8

~~public static void rotate (int [] a, int k) {~~

~~// Part 1~~

~~reverse (a, 0, a.length - k - 1);~~

~~// Part 2~~

~~reverse (a, a.length - k, a.length - 1);~~

public static void ~~rotate~~ (int [] a, int k) {

$k = k \% a.length$; → multiple of k rotation code

if ($k < 0$) {

$k = k + a.length$; } → NEGATIVE ROTATION CODE

}

// Part 1

reverse (a, 0, a.length - k - 1);

// Part 2

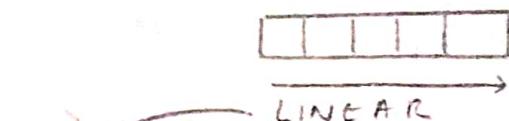
reverse (0, a.length - k, a.length - 1);

// All

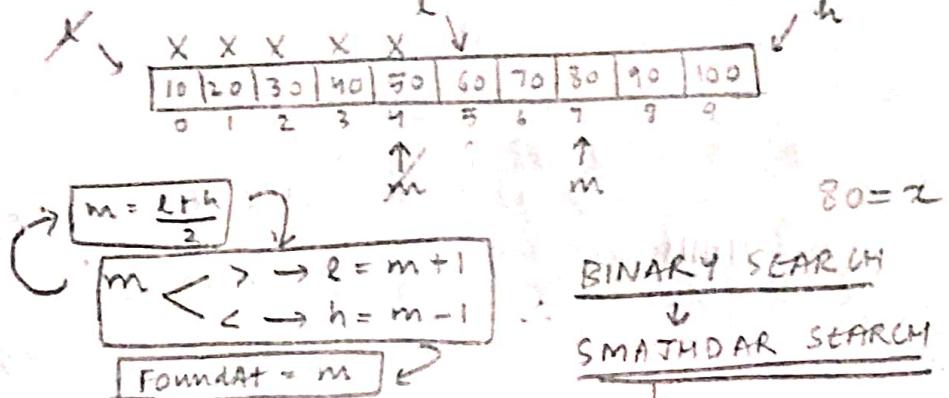
reverse (a, 0, a.length - 1);

}

BINARY SEARCH



HAR ELEMENT KO SEARCH KARENA.
AGAR $n = 10^3$ ELEMENTS HUAGY TDH 10^3 COMPARISONS KRENA.



BINARY SEARCH

SIMPLIFIED SEARCH

ONLY WORKS WHEN SORTED ARRAY IS GIVEN.

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

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

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

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

```
for (int i=0; i<arr.length; i++)
```

```
{ arr [i] = s.nextInt(); }
```

```
int data = s.nextInt();
```

```
int lo = 0;
```

```
int hi = arr.length - 1;
```

```
int foundAt = -1;
```

```
while (lo <= hi) {
```

```
    int mid = (lo + hi) / 2;
```

```
    if (data > arr [mid]) {
```

```
        lo = mid + 1;
```

```
    } else if (data < arr [mid]) {
```

```
        hi = mid - 1;
```

```
    } else {
```

```
        foundAt = mid;
```

```
        break;
```

```
    }
```

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

ARRAY

READY

$$80 = ?$$

BROKEN ECONOMY

IN A COUNTRY OF NOVICE COUNTRY, THE ~~NO~~ ECONOMIC SYSTEM IS CHANGED WHERE ONLY COINS ARE USED THAT TOO OF VARIOUS DENOMINATIONS. WHENEVER A FOREIGNER VISITS THIS COUNTRY, THEY VISIT A MONEY EXCHANGER TO GET THE PREFER'S TO TELL THEM THE DENOMINATION WHICH IS THE NEAREST MAXIMUM AND MINIMUM TO THE DENOMINATION MENTIONED BY FOREIGNER.

IN CASE THEY GET THE CORRECT GUESS OF THE DENOMINATION, THEY ARE TOLD THE SAME DENOMINATION. THE DENOMINATIONS ARE ALWAYS QUOTED IN ASCENDING ORDER.

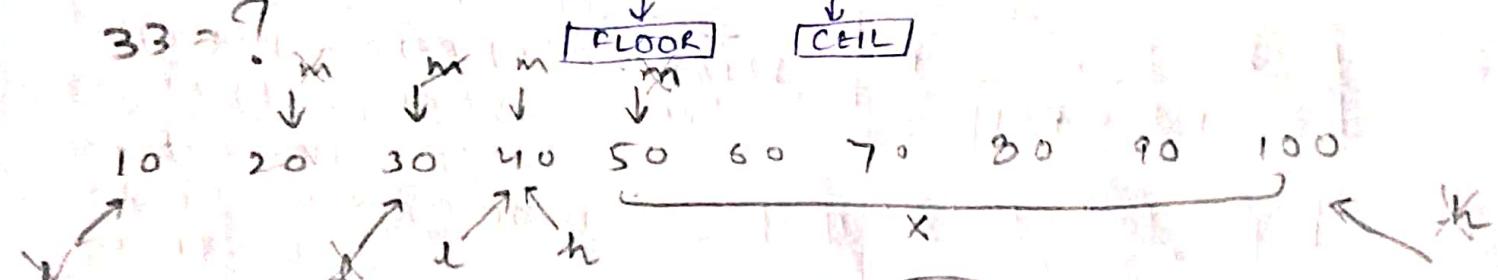
⇒ IN COUNTRY, 5 GIVEN DENOMINATIONS ARE AS FOLLOWS :-

[7, 14, 18, 25, 30]

FOREIGNER ASK 18 → MONEY EXCHANGE WILL TELL SAME DENOMINATION IS AVAILABLE

FOR $2^6 \rightarrow$ MONEY EXCHANGE WILL TELL DENOMINATIONS OF (25) AND (30) ARE AVAILABLE

33 = ?



$$m = \frac{a+b}{2} = 50, 20, 30, 40$$

$$1011 = \frac{2}{7} \times 501 - 40$$

$$F100R = -\varphi 20^{\circ}30'$$

psvm(s[])a){

Scanner S+ new Scanner (System.in);

```
int r = s.resetInt();
```

```
int [7] arr = new int [n];
```

```
for (int i=0; i< arr.length; i++)
```

```
{ arr[i] = s.nextInt();
```

• *www.* •

```
int data = s.nextInt();
```

- JAB high | mid-1 RE ATTA

hai joh hum CEIL ko set
kratety hai!

= JAB low mid + 1 pe atta hai

toh chum FLOOR ko set krde ty
hai,

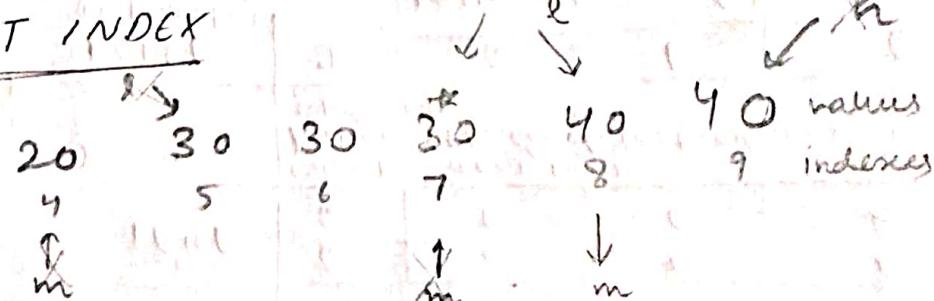
```

int lo = 0;
int hi = arr.length;
int ceil = 0;
int floor = 0;
while (lo <= hi) {
    int mid = (lo + hi) / 2;
    if (data < arr[mid]) {
        hi = mid - 1;
        ceil = arr[mid];
    } else if (data > arr[mid]) {
        lo = mid + 1;
        floor = arr[mid];
    } else {
        ceil = arr[mid];
        floor = arr[mid];
        break;
    }
}
Sys.out.println(ceil);
Sys.out.println(floor);

```

FIRST INDEX AND LAST INDEX

10 10 10 20
data 1 2 3



$$30 = ?$$

LI
FI

NORMAL
BS

$$d < m$$

$$h = m - 1$$

$$d > m$$

$$l = m + 1$$

$$d = m$$

break;

THIS OUTSTN
BS

$$d < m$$

$$h = m - 1$$

$$d > m$$

$$l = m + 1$$

$$d = m$$

li = m → potential Answer

l = m + 1 max krdega

$$d < m$$

$$h = m - 1$$

$$d > m$$

$$l = m + 1$$

$$d = m$$

$$ci = m$$

$$l = m + 1$$

```

public static void main (String [] args) {
    Scanner s = new Scanner (System.in);
    int n = s.nextInt();
    int [] arr = new int [n];
    for (int i = 0; i < arr.length; i++) {
        arr[i] = s.nextInt();
    }
    // ENTER DATA TO BE SEARCHED
    int data = s.nextInt();
    int low = 0; // FIRST INDEX
    int high = arr.length - 1;
    int lastIndex = -1;
    int firstIndex = -1;
    while (low <= high) {
        int mid = (low + high) / 2;
        if (data < arr[mid]) {
            high = mid - 1;
        } else if (data > arr[mid]) {
            low = mid + 1;
        } else { // potential answer
            lastIndex = mid;
            low = mid + 1;
        }
    }
}

```

```

3 // BINARY SEARCH FOR
low = 0; // FIRST INDEX
high = arr.length - 1;
while (low <= high) {
    int mid = (low + high) / 2;
    if (data < arr[mid]) {
        high = mid - 1;
    } else if (data > arr[mid]) {
        low = mid + 1;
    } else { // potential answer
        firstIndex = mid;
        high = mid - 1;
    }
}

```

```

3 System.out.println (firstIndex);
3 System.out.println (lastIndex);
3

```

loop condition	low	high	mid	lastIndex
TRUE	0	9	-	-1
	0	9	4	-1
	5	9	4	4
	5	9	7	4
	5	6	7	4
	5	6	5	4
	6	6	5	5
	6	6	6	5
FALSE as low>high	7	6	6	6
				lastIndex = 6

15 10 10 20 20 20 20 30 40 40 → values
 ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ → indices

Data = 20

Last index = 6
 First index = 3

→ WE DO NOT BREAK ON GETTING THE MATCH AND FURTHER UPDATE THE LOW POINTER AS THERE COULD BE ANOTHER INDEX HOLDING THE REQUIRED VALUE ON THE RHS.

→ WE DON'T BREAK ON GETTING THE MATCH AND FURTHER UPDATE THE HIGH POINTER AS THERE COULD BE ANOTHER INDEX HOLDING THE REQUIRED VALUE ON THE LEFT SIDE.

loop condition	low	high	mid	last index
TRUE	0	9	-	-1
	0	9	4	-1
	condition array[mid] == data is satisfy, i.e. low = mid + 1 and last index = mid are updated.			
TRUE	5	9	4	4
	5	9	7	4
	condition array[mid] > data is satisfy, high = m - 1 is updated			
TRUE	5	6	7	4
	5	6	5	4
	condition array[mid] == data is satisfy, low = mid + 1 and last index = mid are updated			
TRUE	6	6	5	5
	6	6	6	5
	condition array[mid] == data is satisfy, low = mid + 1 and last index > mid is updated.			
FALSE as low > high	7	6	6	6
	last_index = 6			

10	10	10	20	20	20	20	30	40	40	→ values
0	1	2	3	4	5	6	7	8	9	→ indexes

$$\text{Data} = 20$$

$$\text{last index} = 6$$

$$\text{First index} = 3$$

Q) SUBSET OF ARRAY

SUBARRAY

a b c

a	b	c
a	b	c
a	b	c
b	c	
b	c	
c		

SUBARRAY

$\frac{n(n+1)}{2}$

GIVEN

10, 20

\downarrow

$n=2$

0 → 0 0	→ - -
1 → 0 1	→ - 20
2 → 1 0	→ 10 -
3 → 1 1	→ 10 20

BINARY

2 6
2 3 - 0
2 1 - 1
0 - 1 - 0

ab c
↓
6 → 110

HUMAN

OUTPUT

- ba

```

for(int i=0; i<limit; i++) {
    String set = "";
    int temp = i;
    for(int j=0; j<arr.length; j++) {
        int r = temp % 2;
        temp = temp / 2;
        if(r == 0) {
            set = "-\\" + set;
        } else {
            set = arr[j] + "\\" + set;
        }
    }
    System.out.println(set);
}

```

WRONG

SUBSET

a	b	c
-	-	-
-	-	c
-	b	-
-	b	c
a	-	-
a	-	c
a	b	-
a	b	c

$n = \text{arr.length}$

$\therefore 2^n = \text{NO. OF SUBSETS}$

Psvm (scra)

scanners = new Scanner (System.in);

int n = scanner.nextInt();

int []arr = new int [n];

for(int i=0; i<arr.length; i++)

{ arr[i] = scanner.nextInt(); }

}

int limit = (int) Math.pow(2, arr.length);

for(int i=0; i<limit; i++)

{ String set = "";

int temp = i;

for(int j=arr.length; j>=0; j--)

{ int r = temp % 2;

temp = temp / 2;

if(r == 0) {

set = "-\\\" + set;

}

else {

set = arr[j] + "\\\" + set;

}

System.out.println(set);

}

CORRECT