

Anybase Multiplication

$$\begin{array}{r}
 & 1 & 0 & j & 1 & j & j \\
 & 7 & 3 & 9 & 5 & 1 & 5 \\
 \hline
 1 & 0 & x & 2 & & & \\
 \hline
 3 & 6 & 9 & 5 & 2 & & \\
 & 1 & 4 & 7 & 8 & & \\
 \hline
 & 1 & 8 & 4 & 7 & 5 & 0 \\
 \hline
 \end{array}$$

Decimal
Adelut

reg' →

$$739 \times (20 + 5)$$

$$\text{ceny} = 0$$

$7 \times 3 \times 9 = 63$

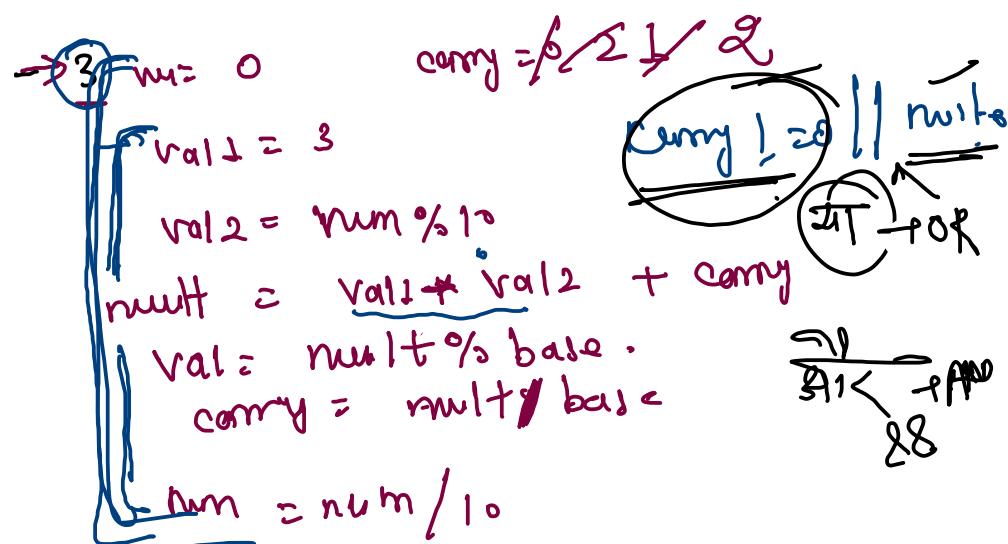
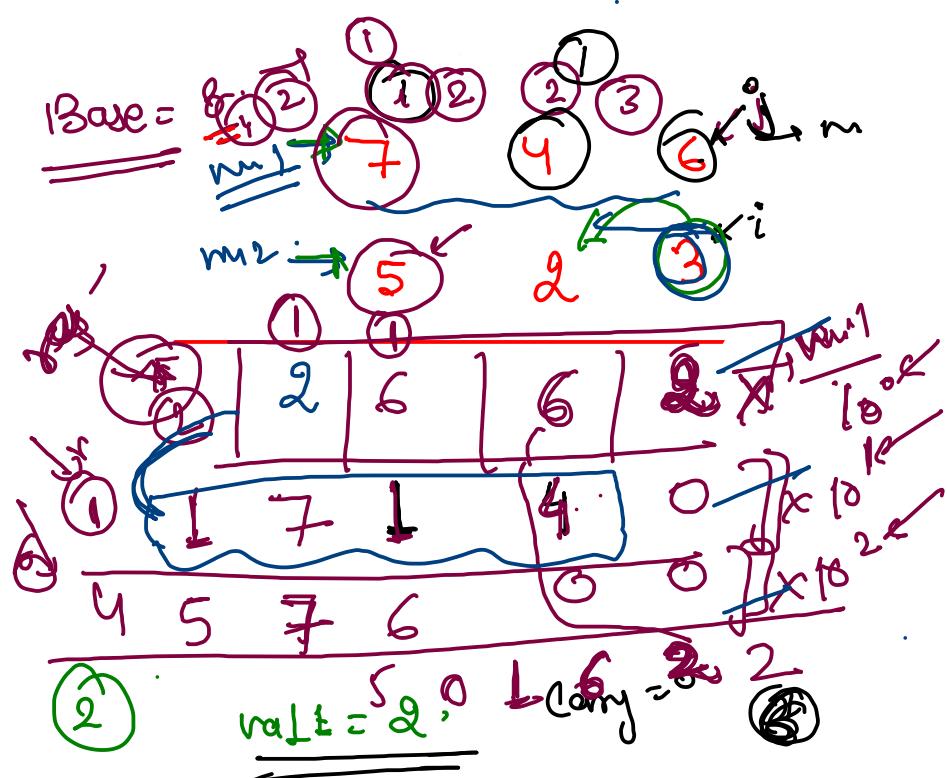
$45\% \times 10 \rightarrow \text{val}$

Count digit

$$\text{carry} = 45 / 10$$

7 → 7
19 % 10 → Val = 9
 $19 / 10 \rightarrow \text{com}$

$$\begin{array}{r}
 867.10 \rightarrow \text{Va} \\
 86 / 10 \rightarrow \text{carry} \\
 187.10 \rightarrow 8 \rightarrow \text{Va} \\
 18 / 10 \rightarrow \text{carry} \\
 14 / 10 = 1
 \end{array}$$



$$\text{quot} = 2662$$

5	0	1	6	2	2
---	---	---	---	---	---

$$\begin{array}{c}
 (2) \rightarrow \\
 \xrightarrow{2 \times 10^0} \rightarrow 100 \\
 (1) \rightarrow \\
 \xrightarrow{6 \times 10^1} \rightarrow \\
 (6) \rightarrow \\
 \xrightarrow{6 \times 10^2} \rightarrow \\
 (2) \rightarrow \\
 \xrightarrow{2 \times 10^3} \rightarrow
 \end{array}$$

power = 1 $\rightarrow 10^0$

$res := val * power$; }
 $power \leftarrow 10^1$

overall res -

power = ~~1~~ ~~2~~ ~~6~~ ~~1000~~ $\rightarrow 1000$

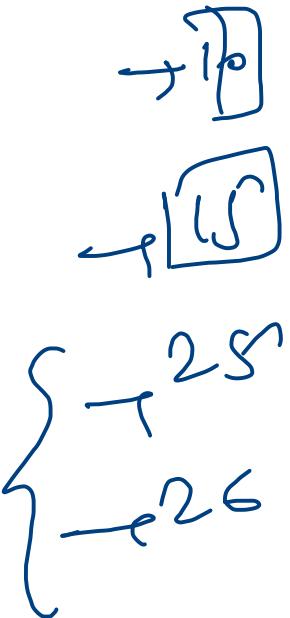
val = ~~2~~ ~~6~~ ~~6~~ $\rightarrow 2$

$$\begin{aligned}
 res &= \underbrace{2 * 1}_{\text{1}} + \underbrace{6 * 10}_{\text{60}} + \underbrace{6 * 100}_{\text{600}} + \\
 &\quad + \underbrace{2 * 1000}_{\text{2000}}
 \end{aligned}$$

work = 2 662, By -

r_{i0}

- (P) Any base-
- (C) Cmrc
- (S) Quat



$$res = \cancel{0} \cancel{10} \cancel{28} \cancel{80} \quad \textcircled{7c}$$

$$res = \overset{0}{\cancel{0}} + 2662 \xrightarrow[B_{9e}]{+} (17140)$$

Diagram showing a sequence of four boxes: T, A, G, and C. The first three are grouped by a brace under them, labeled 'T 25'. The fourth box is labeled 'C 26'. An oval labeled 'res' is connected to the sequence by arrows pointing to each box. Below the sequence, a wavy line connects to another oval labeled 'res-'.

r_{i0}

- (P) Any base-
- (C) Amine
- (A) Quaternary

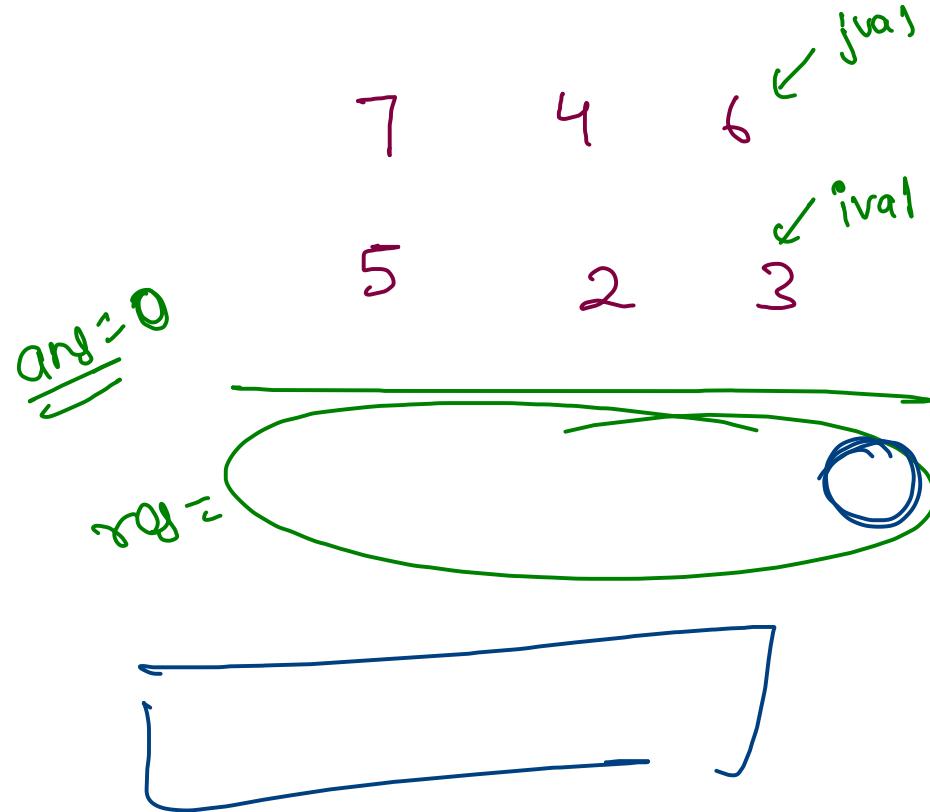
$\rightarrow 10$

$\leftarrow 15$

$\left\{ \begin{array}{l} \rightarrow 25 \\ \rightarrow 26 \end{array} \right.$

$r_{es} = \cancel{0.10} \cancel{28}$ $\cancel{80} \circled{75}$

$r_{es} = 0$ $(0 + 2662) \xrightarrow[B_{9e}]{+} (17140)$



$\text{mult} := \text{ival} * \text{jval} + \text{carry};$
 ↘ val = mult % base;
 ↘ carry = mult / base;

18
 $\text{res} += \text{val} * \text{powerOne}$
 $\text{val} = 2$
 $\text{carry} = 2$



Primitive data type-

Default

- (1) byte = 0
- (2) short = 0
- (3) int = 0
- (4) long = 0
- (5) float = 0
- (6) double = 0
- (7) char = 'n'
- (8) boolean = false

Non primitive D.T.

Stack

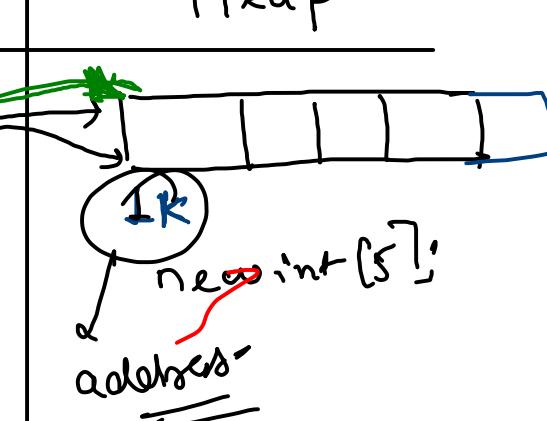
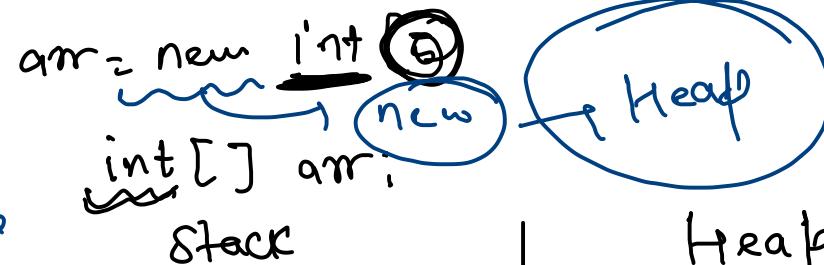
By default null

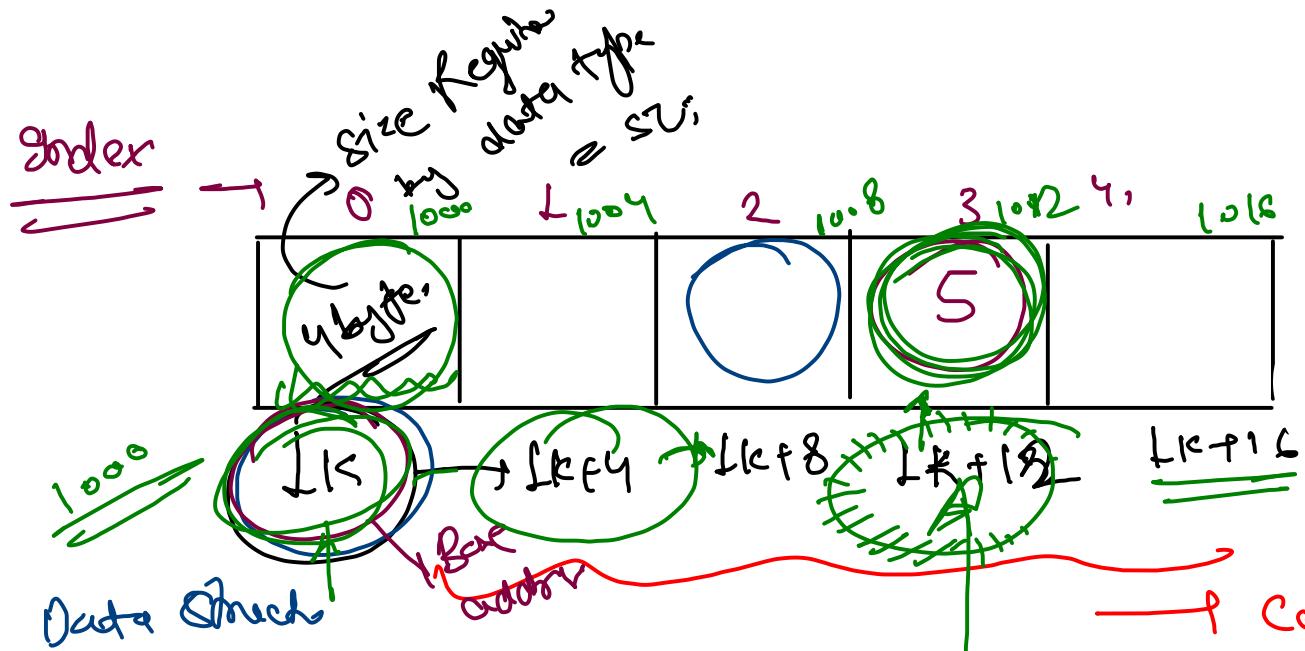
Return is available on stack

Memory (RAM)

Stack { Primitive Data types,

Heap { Non primitive data types,





- ① get { Retrieval }
- ② set → Add / Update
- ③ Remove =

`int var1`

`var1 = 5;`

ans = 10

$Lc + 4 \times 3$
 $Lc + 12$

Memory allocation

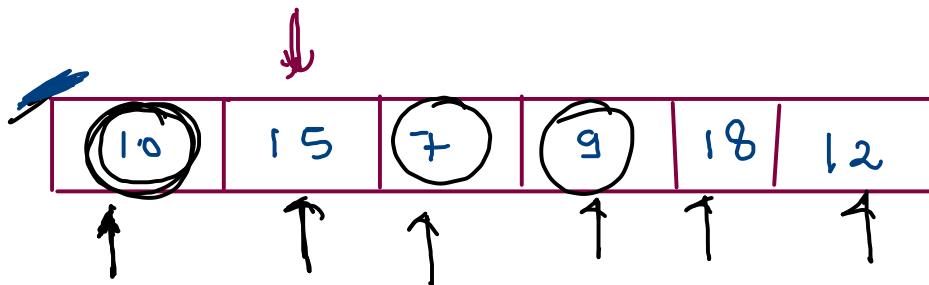
Span of Array -

~~arr.size = 20~~

Span = Max - Min

~~min = +∞~~

~~7~~



Identify
of

if (arr[i] < min) {

min = arr[i];

n

Min

→ ~~+∞~~ ⇒

}

n diff elemts

~~nmin = arr[0],~~

if arr[i] < min > 0
out of
bound

Span

Identity

$$a \text{ op. } b = a$$

b is identity element for operator op.

$$a * e = a$$

\uparrow
operator

e is identity for *

$$\begin{matrix} * \\ + \\ - \\ / \\ X \end{matrix} \rightarrow \begin{matrix} e \\ \cancel{e} \\ \cancel{e} \\ \cancel{e} \\ \cancel{e} \end{matrix}$$

$$* \rightarrow \min, \quad \cancel{*} \rightarrow \max$$

$$\min(a, e) = a \quad \left. \begin{array}{l} e = +\infty \\ e = -\infty \end{array} \right\}$$

$$\max(a, e) = a \quad \left. \begin{array}{l} e = +\infty \\ e = -\infty \end{array} \right\}$$

$$\begin{aligned} a + e &= a, \Rightarrow e = 0 \\ a - e &= a, \Rightarrow e = 0 \\ a/e &= a, \Rightarrow e = 1 \\ a \times e &= a \Rightarrow e = 1 \end{aligned}$$

```

public static Scanner scn = new Scanner(System.in);

public static void takeInput(int[] arr, int n) {
    for(int i = 0; i < n; i++) {
        arr[i] = scn.nextInt();
    }
}

```

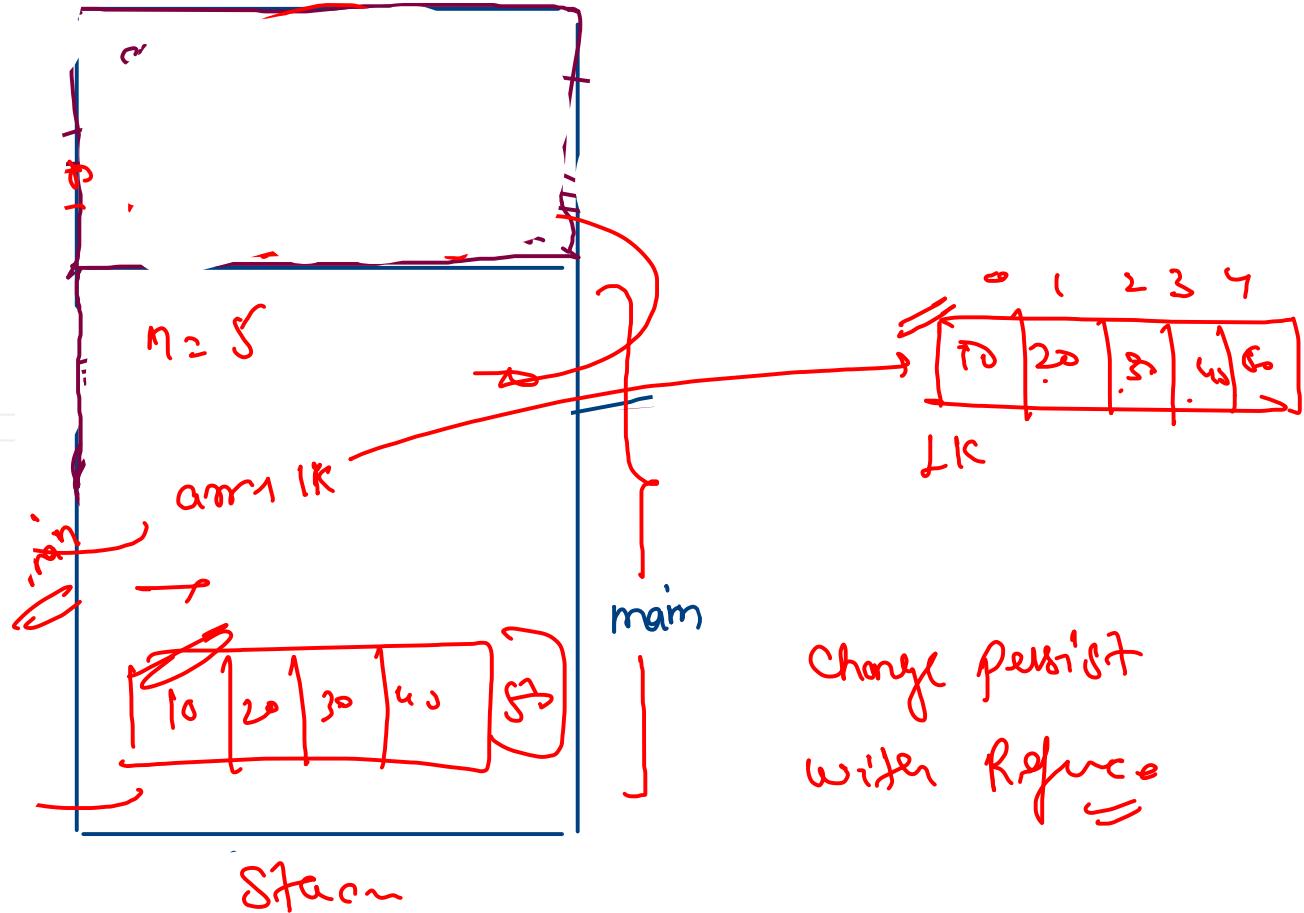
Run | Debug

```

public static void main(String[] args) {
    // take size from user
    int n = scn.nextInt();
    // create array of size n
    int[] arr = new int[n];

    takeInput(arr, n);
}

```



Find an Element in array -

$$Snp \leq n$$

n diff. elm

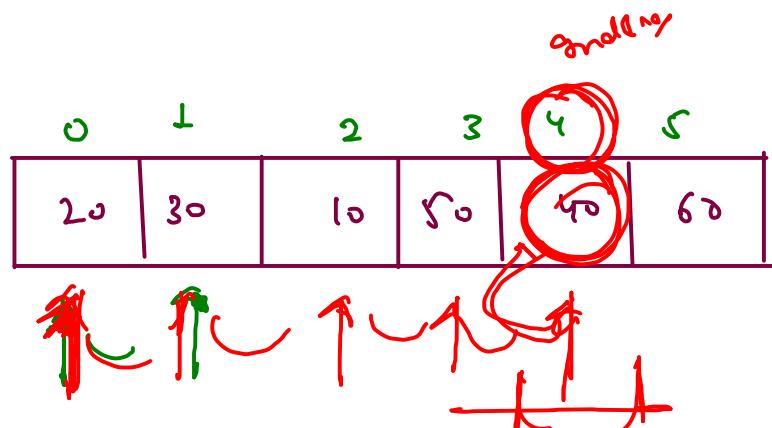
$$dtf = 40$$

o? Index of dtf -

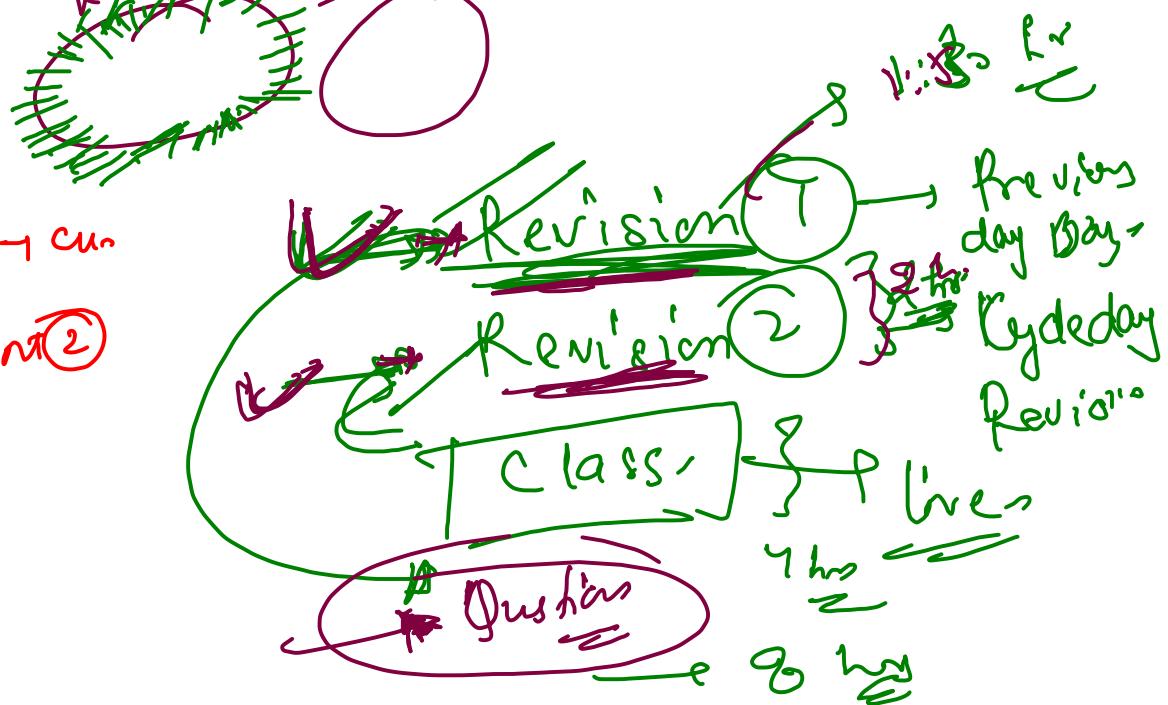
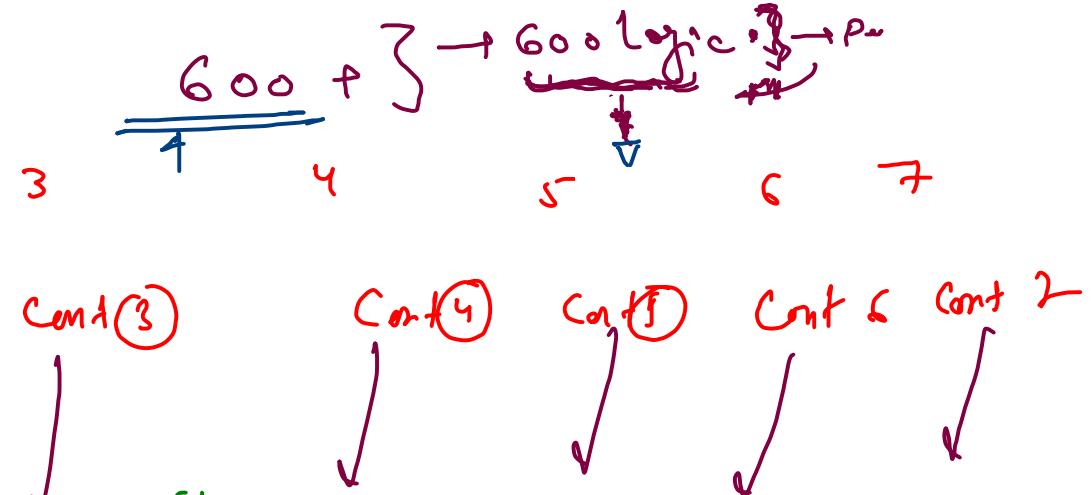
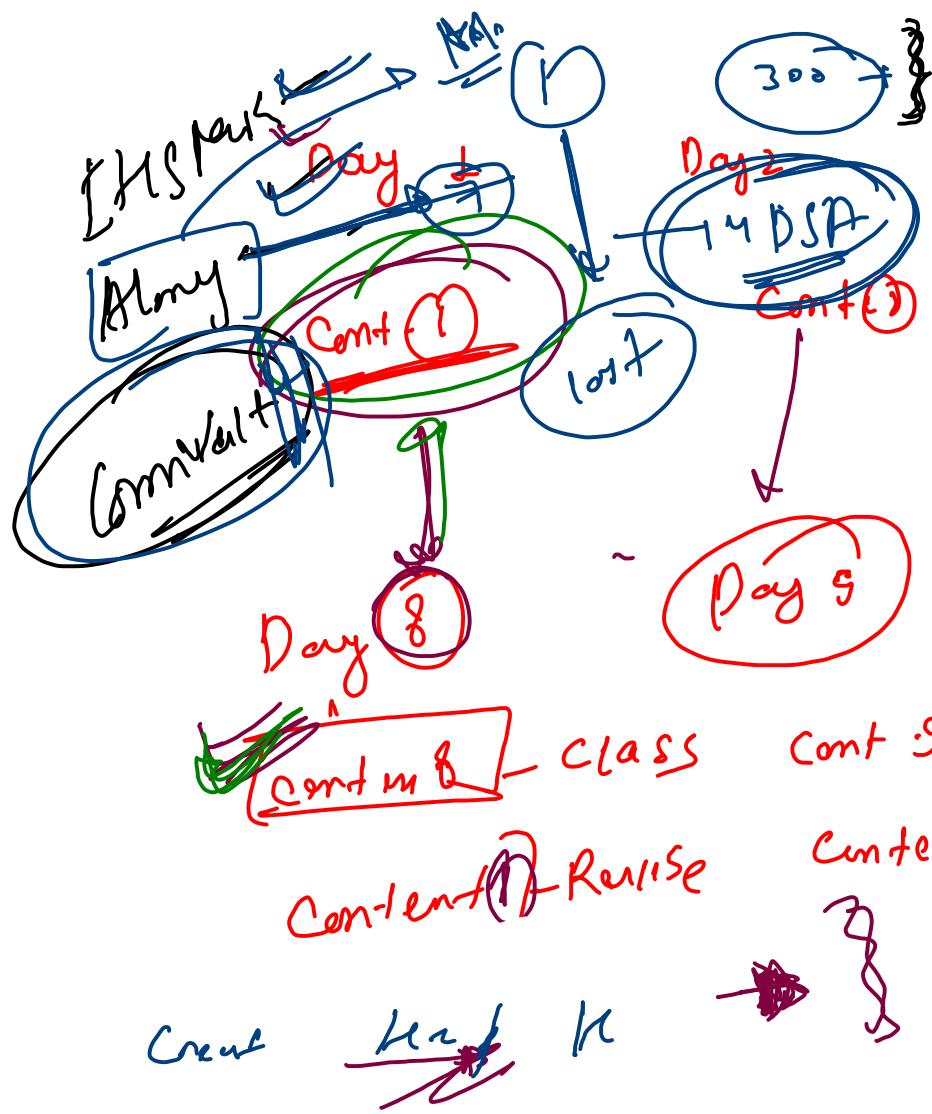
$$dtf = 40$$

isPresent = False;

present - To show
Absent - \rightarrow False



```
if(arr[i] == dtf) {
    isPresent = True;
    break;
}
```



MSIT

Amazon \rightarrow Test \rightarrow

