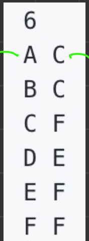
Employee and Managers

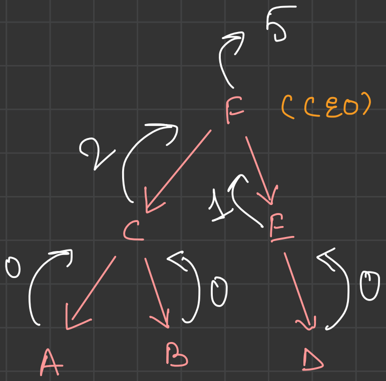
eg :



```
6
A C
B C
C F
D E
E F
F F
```

key (Emp)

value (Mgr)



o/p

A	→ 0
B	→ 0
C	→ 2
D	→ 0
E	→ 1
F	→ 5

faith : tell me number of emp under you?

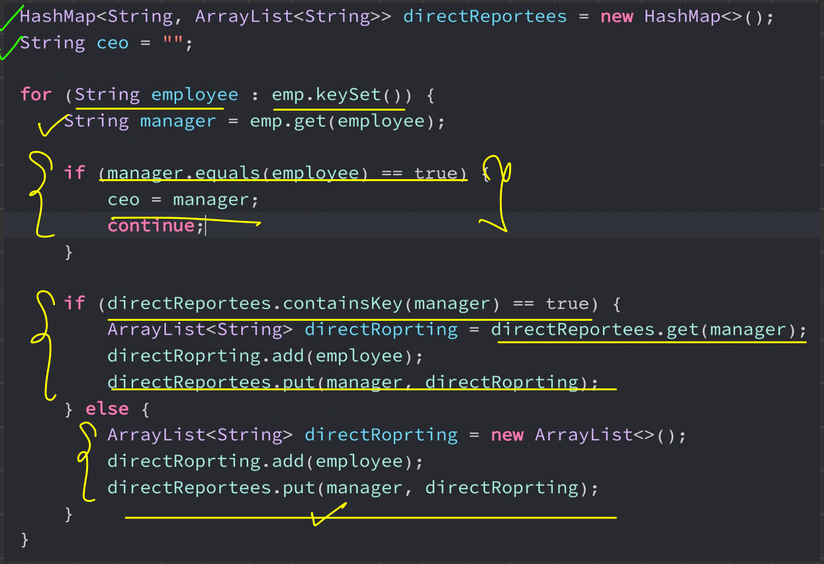
direct reportees (HashMap)

Key (String)

C	→ { A, B }
E	→ { D }
F	→ { C, D }

AL<String>

A	→ 0
B	→ 0
C	→ 2
D	→ 0
E	→ 1
F	→ 5

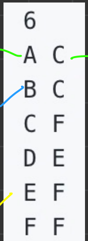


```
HashMap<String, ArrayList<String>> directReportees = new HashMap<>();
String ceo = "";

for (String employee : emp.keySet()) {
    String manager = emp.get(employee);

    if (manager.equals(employee) == true) {
        ceo = manager;
        continue;
    }

    if (directReportees.containsKey(manager) == true) {
        ArrayList<String> directRoprting = directReportees.get(manager);
        directRoprting.add(employee);
        directReportees.put(manager, directRoprting);
    } else {
        ArrayList<String> directRoprting = new ArrayList<>();
        directRoprting.add(employee);
        directReportees.put(manager, directRoprting);
    }
}
```



```
6
A C
B C
C F
D E
E F
F F
```

Key

Value

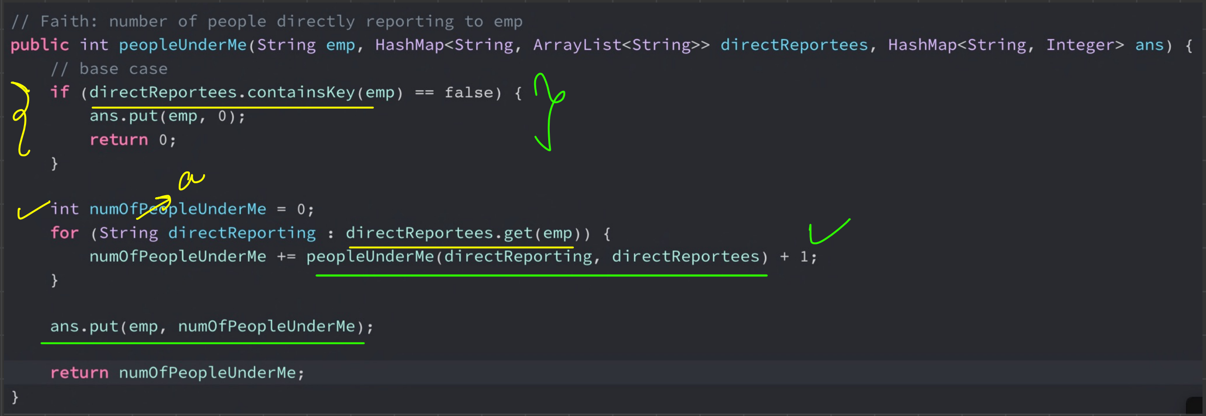
Emp

employee = F

manger = F

CEO = F

String	AL<String>
C	{A, B}
F	{C, E}
E	{D}



```java
// Faith: number of people directly reporting to emp
public int peopleUnderMe(String emp, HashMap<String, ArrayList<String>> directReportees, HashMap<String, Integer> ans) {
    // base case
    if (directReportees.containsKey(emp) == false) {
        ans.put(emp, 0);
        return 0;
    }

    int numOfPeopleUnderMe = 0;
    for (String directReporting : directReportees.get(emp)) {
        numOfPeopleUnderMe += peopleUnderMe(directReporting, directReportees) + 1;
    }

    ans.put(emp, numOfPeopleUnderMe);

    return numOfPeopleUnderMe;
}
```

C ⟶ {A, B}

E ⟶ {D}

F ⟶ {C, E}

A ⟶ 0

B ⟶ 0

C ⟶ 2

D ⟶ 0

E ⟶ 1

F ⟶ 5

Problem with given diff

arr[] = {5, 10, 3, 2, 50, 80}   B = 78

Brute force

```
for(int i → 0 → n)
    for(int j = i+1 → n)
        if (arr[i] - arr[j] == B or arr[j] - arr[i] == B)
            return true;

return false
```

TC: $O(N^2)$

SC: $O(1)$

$arr[] = \{5, 10, 3, 2, 50, 80\}$ $\quad B = 78$

$\cancel{x} \quad \cancel{x} \quad \cancel{x} \quad x$

$5, 10, 3, 2, 50, 80$ ✓

$\underline{\underline{(x, y)}}$

$\hookrightarrow x - y = B \longrightarrow y = x - B$

$\underline{\underline{\text{OR}}}$ $\quad y - x = B \longrightarrow y = x + B$

✓ $x = 2$, $y = 80$

$\underline{\underline{x = 5}}$

$y = 5 - 78 = -73$

$y = 5 + 78 = 83$

$\underline{\underline{x = 10}}$

$y = 10 - 78 = -68$

$y = 10 + 78 = 88$

$\underline{\underline{x = 3}}$

$y = 3 - 78 = -75$

$y = 3 + 78 = 81$

$\underline{\underline{x = 2}}$

$y = 2 - 78 = -76$

$\underline{\underline{y = 2 + 78 = 80}}$

Array Pair divisible by k

$$\text{arr[] - } \left\{ \begin{array}{cccccccccc} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ 1, & 2, & 3, & 4, & 5, & 6, & 7, & 8, & 9, & 10 \end{array} \right\} \qquad k = 5$$

$\downarrow$

Pairs = 5

(1,9), (2,8), (3,7), (4,6), (10,5)

T	T	T	T	T	T	T	T	T	T
0	1	2	3	4	5	6	7	8	9

arr[] - { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 } k = 5

(1, 4) (2, 3) (5, 10) (6, 9) (7, 8)

True

Brute force

{ TC: O(N^2) }
{ SC: O(N) } ✓

$$\begin{array}{lccccccccccl} & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & \\ arr[\,] = \{ & 1, & 2, & 3, & 4, & 5, & 6, & 7, & 8, & 9, & 10 & \} \end{array}$$

$k = 5$

$(x, y) \rightarrow$ pair

Two num, (x, y) can pair up together when sum of their sum is divisible by K.

✓ $(x + y)$ should be divisible by K

$$x = K \times q_1 + r_1$$

$$y = K \times q_2 + r_2$$

$$x + y = (Kq_1 + r_1) + (Kq_2 + r_2)$$

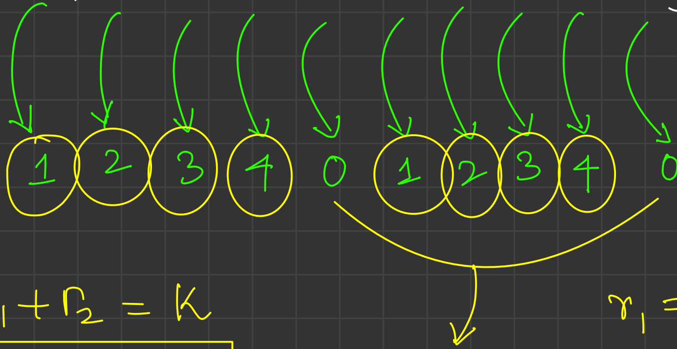
$$\boxed{x + y = K(q_1 + q_2) + (r_1 + r_2)}$$

$K(q_1 + q_2)$ ↳ divisible by K

$(r_1 + r_2)$ ✓ → should be divisible by K

0 1 2 3 4 5 6 7 8 9

arr[ ] - {1, 2, 3, 4, 5, 6, 7, 8, 9, 10} k = 5



Even quantity!

rem	freq
1	→ 2
2	→ 2
3	→ 2
4	→ 2
0	→ 2

$r_1 + r_2 = K$

$$\boxed{r_2 = K - r_1}$$

$r_1 = 1$
$r_2 = 4$

$r_1 = 2$
$r_2 = 3$

$r_1 = 3$
$r_2 = 2$

$r_1 = 4$
$r_2 = 1$

largest Subarray with zero Sum

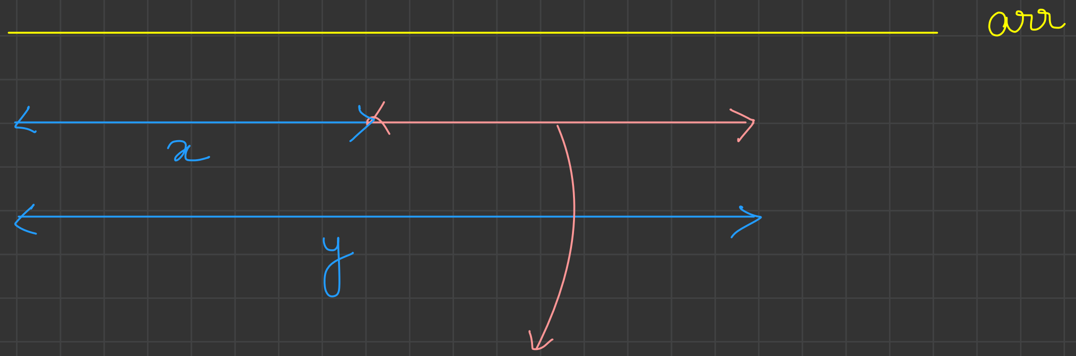
$$arr[\,] = \left\{ \begin{array}{cccccccc} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 15, & -2, & 2, & -8, & 1, & 7, & 10, & 23 \end{array} \right\}$$

Brute force.

Compute all subarray sum, and store max Len, when sum is zero.

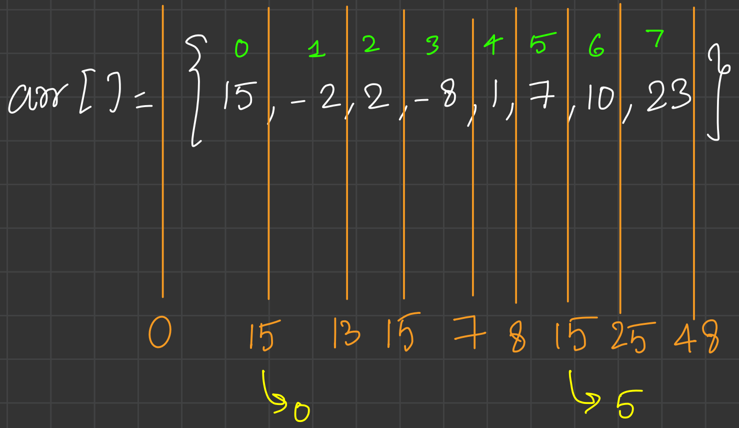
$$\left\{ \begin{array}{l} TC: O(N^2) \\ SC: O(1) \end{array} \right.$$

$$arr[\,] = \left\{ \begin{array}{cccccccc} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 15, & -2, & 2, & -8, & 1, & 7, & 10, & 23 \end{array} \right\}$$



{ if (x == y)

Subarray sum is zero



len = 2 − 0 = 2

len = 5 − 0 = 5

max len = ~~2~~ 5

$$\text{arr}[\,] = \left\{ \overset{0\quad\;\; 1\quad\;\; 2\quad\;\;\; 3\quad\;\;\;\; 4\quad\;\; 5\quad\;\;\; 6}{\boxed{2,\ 3,\ 10,\ -10,\ -3,\ 5,\ -7,}}\ \overset{7}{9} \right\}$$

maxlen = ~~2~~ ~~4~~ 7

Sum = ~~0~~ ~~2~~ ~~5~~ ~~15~~ ~~5~~ ~~2~~ ~~7~~ 0

len = 3 - 1 = 2

= 4 - 0 = 4

= 6 - (-1) = 7

Map

Sum	firstSeenIndex
2	0
5	1
15	2
7	5
0	-1

Subarray Sum Equal to K

int [] arr = { 10, 2, -2, -20, 10 }   K=10

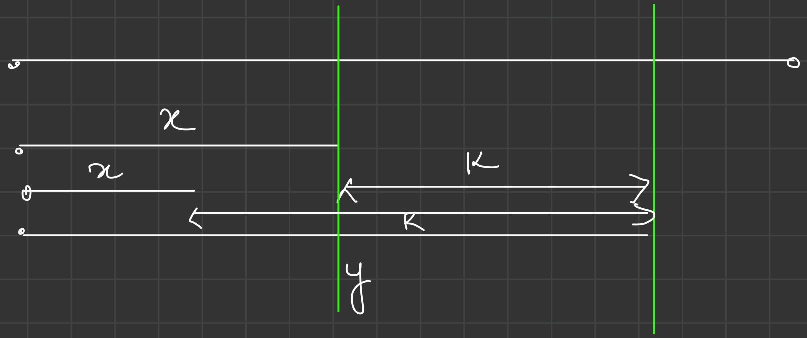
3- subarrays

brute force

↳ get all subarray sum, and check is equal to K inc. ans.

$$\begin{cases} TC: O(N^2) \\ SC: O(1) \end{cases}$$

int[] arr = {0, 0, 10, 2, -2, -20, 10}   K=10



$$\boxed{x = y - K}$$

<u>find</u>

int [] arr = { 0, 0, 10, 2, -2, -20, 10 }   K=10

0   0   0   10   12   10   -10   0

numOfSubarray = ~~3~~ ~~6~~ 7

Sum	freq
0	~~1~~ ~~2~~ 3
10	~~1~~ 2
12	1
-10	1

$x = y - K$

$0 - 10 = -10$

$10 - 10 = 0$

$12 - 10 = 2$

$10 - 10 = 0$

$-10 - 10 = -20$

$0 - 10 = -10$