Pair Of Songs With Total Durations Divisible By 60

time	30	20	150	100	40
	0	1	2	3	4

$$(x+y) \% 60 = 0$$

$$\Rightarrow (x \% 60 + y \% 60) \% 60 = 0$$

$$\Rightarrow (R_1 + R_2) \% 60 = 0$$

if $60 = z$

$$(x+y) \% z = 0$$

$$\Rightarrow (x \% z + y \% z) \% z = 0$$

$$\Rightarrow (R_1 + R_2) \% z = 0$$

$$0 \leq R_1 < Z$$

$$0 \leq R_2 < Z$$

$$\Rightarrow \boxed{0 \leq R_1 + R_2 < 2Z}$$

So, $0 \leq R_1 + R_2 < 120$

That means, if $(R_1 + R_2) \% Z = 0$

then $\boxed{\begin{array}{c} R_1 + R_2 = 0 \\ \text{or,} \\ R_1 + R_2 = Z \end{array}}$

remainder	30 % 60	20 % 60	150 % 60	100 % 60	40 % 60
	0	1	2	3	4

remainder	30	20	30	40	40
	0	1	2	3	4

Now, we just have to find all pairs whose sum is 60 i.e, (Z).

remainder	30	20	30	40	40
	0	1	2	3	4

hashMap

⋮
40, 2
20, 1
30, 2

30 — 2

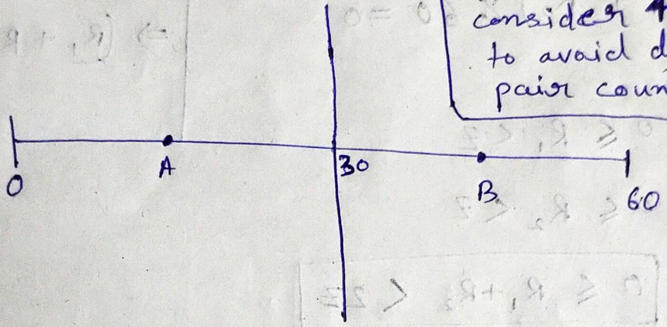
$\frac{2(2-1)}{2} = 1$

20 — 1

1 * 2 = 2

40 — 2

We won't consider 40 to avoid duplicate pair count



(A, B) is a pair

So, we just have to check from 1 to 29.

0 and 30 are special case

- For any remainder[i] in the range [1 to 29] :-

  No. of pairs = $f_1 * f_2$

  where f_1 = frequency of remainder[i]

  f_2 = frequency of 60 - remainder[i]

- For any remainder[i] == 0 or 30 :-

  No. of pairs = $\frac{n(n-1)}{2}$

  where n = frequency of remainder [i]

We can also solve it using DAT (Direct Access Table) :- (Instead of HashMap)

frequency

0	1	2	20	30 ...	40 ...	...56	57	58	59
0	0	0	1	2	2	0	0	0	0

This array is used to store frequency of remainders when each element of time[] array is divided by 60

```
int[] frequency = new int[60]
  for(int i=0; i<=n-1; i++){
       frequency[time[i]%60]++;
  }
```

```
int count=0;
 count += frequency[0] * (frequency[0]-1)/2;
 count += frequency[30] * (frequency[30]-1)/2;
 for(int i=1; i<=29; i++){
       count += frequency[i] * frequency[60-i];
 }
```

Now, count has total no. of pairs such that (time[i]+time[j]) % 60 = 0