Check Arithmetic Sequence

17 5 9 13 21 25

1st term = 5

2nd term = 9

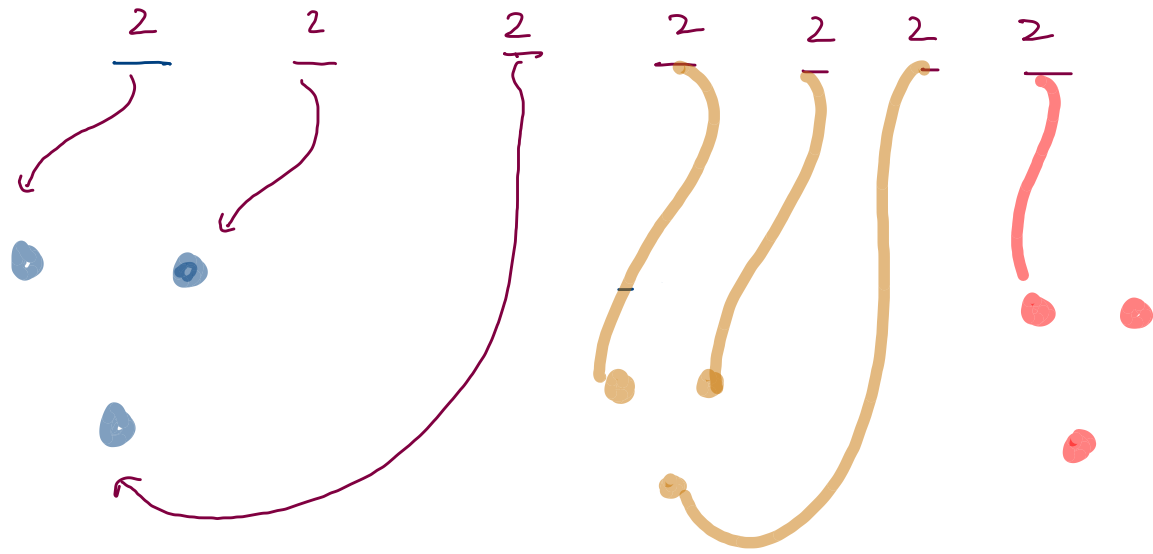
d = 4

5 5+4 9+4 13+4 17+4 21+4

781. Rabbits in Forest

There is a forest with an unknown number of rabbits. We asked n rabbits **"How many rabbits have the same color as you?"** and collected the answers in an integer array `answers` where `answers[i]` is the answer of the i^{th} rabbit.

Given the array `answers`, return *the minimum number of rabbits that could be in the forest*.



2_0 3_1 4_2 3_3 3_4 4_5 2_6 2_7 2_8 3_9

3_{10} 3_{11} 4_{12} 3_{13} 4_{14} 5_{15} 2_{16} 5_{17} 4_{18}

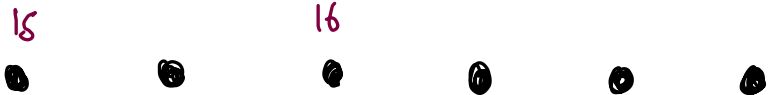
ans vs rabbit count

2 - 5

3 - 7

4 - 5

5 - 2



ans vs rabbit count

$4 * \left\lceil \frac{10}{4} \right\rceil = 12$ 3 - 10

$6 * \left\lceil \frac{9}{6} \right\rceil = 12$ 5 - 9

$5 * \left\lceil \frac{15}{5} \right\rceil = 15$ 4 - 15

a [1 2; 3 4] b [5 6; 7 8] c [9 10; 11 12]

d [13 14 15; 16 17 18] e [19 20 21 22; 23 24]

g [25 26 27; 28 29] h [30 31 32; 33 34] i [35 36; 37 38; 39]

```
int mr = 0;

for(int key : map.keySet()) {
    int val = map.get(key);
    int gc = key + 1;

    mr += gc * Math.ceil(val*1.0 / gc);
}

return mr;
```

ans = 39

100 / 16	10 / 25	2425 / 25	97 / 7	7 / 12
6.25	0.4	97	13.(857142)	0.58(3)

div will stop

[if rem = = 0]

division will never stop.

13.(857|428)

7) 97

91

60

56

40

35

50

49

10

7

30

28

20

14

60

56

4

4
5
1
3
2
6

0.583

12) 70

60

100

96

40

36

4

rem

10

4

0.58(3)

100	10	2425
16	25	25
6.25	0.4	97

```
int q = n / d ;
int rem = n % d ;
ans += q ;
String str ;
if (rem == 0) {

}
else {
    while (rem != 0) {
        n = rem * 10 ;
        q = n / d ;
        rem = n % d ;
        str += q ;
    }
}
```

```
while(rem != 0) {
    if(map.contains(rem) == true) {
        int pos = map.get(rem);
        p2 = p2.substring(0,pos) + "(" + p2.substring(pos) + ")";
        break;
    }

    map.put(rem,p2.length());

    num = rem*10;
    q = num / den;
    p2 += q;
    rem = num % den;
}
```

p2 = "" + (+ 857142 +) ;

7) 97 (

91

60

56

40

35

50

49

10

7

30

28

20

14

6

q = 2

r = 6

p1 = "13"

p2 = 857142

6 -> 0
4 -> 1
5 -> 2
1 -> 3
3 -> 4
2 -> 5

```
while(rem != 0) {
    if(map.contains(rem) == true) {
        int pos = map.get(rem);
        p2 = p2.substring(0,pos) + "(" + p2.substring(pos) + ")";
        break;
    }

    map.put(rem,p2.length());

    num = rem*10;
    q = num / den;
    p2 += q;
    rem = num % den;
}
```

$\frac{7}{12}$

12) 70

60

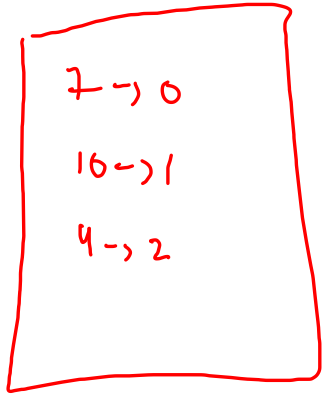
100

96

40

36

4



q = 3

r = 4

p1 = "0"

p2 = 583

p2 = "58" + (+3+)

```
long q = num / den;
long rem = num % den;


String p1 = q + "";
String p2 = "";
```

5 -> 0

2 -> 1

8 -> 2

$\frac{77}{12}$

n = 80

d = 12

```
if(rem == 0) {
    //this division is settled
    if(flag == true) {
        return p1;
    }
    else {
        return  "-" + p1;
    }
}
else {
    while(rem != 0) {
        if(map.containsKey(rem) == true) {
            int pos = map.get(rem);
            p2 = p2.substring(0,pos) + "(" + p2.substring(pos) + ")";
            break;
        }

        map.put(rem,p2.length());

        num = rem*10;
        q = num / den;
        rem = num % den;
        p2 += q;
    }
}
```

q = 6

r = 8

p1 = 6

p2 = 416

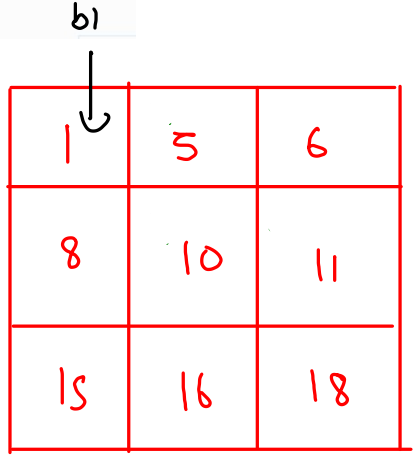
p2 = 41 + '(' + 6 + ')'

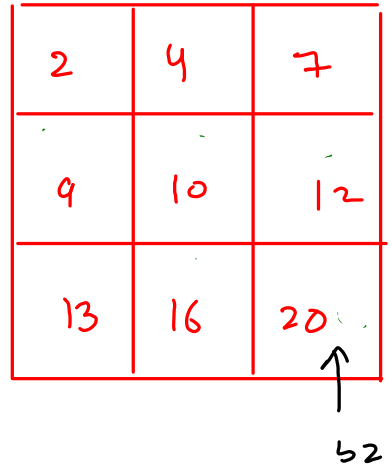
41 (6)

6. 41(6)

Pairs With Given Sum In Two Sorted Matrices

```
3
1 5 6
8 10 11
15 16 18
2 4 7
9 10 12
13 16 20
```





1 20

5 16

8 13

11 10

Distinct pairs

T : O(n2)

S : O(1)

non-distinct pairs

T : O(n2)

S : O(n²)