num = 2 4 6

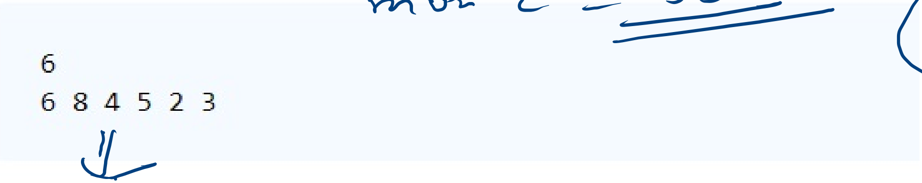
num 2 = 3 5 8

num R

num 358

0	1	2	3	4	5
2	3	4	5	6	8

2 4 6



```
6
6 8 4 5 2 3
```

2 3 4 5 6 8

num1 = 2 4 6

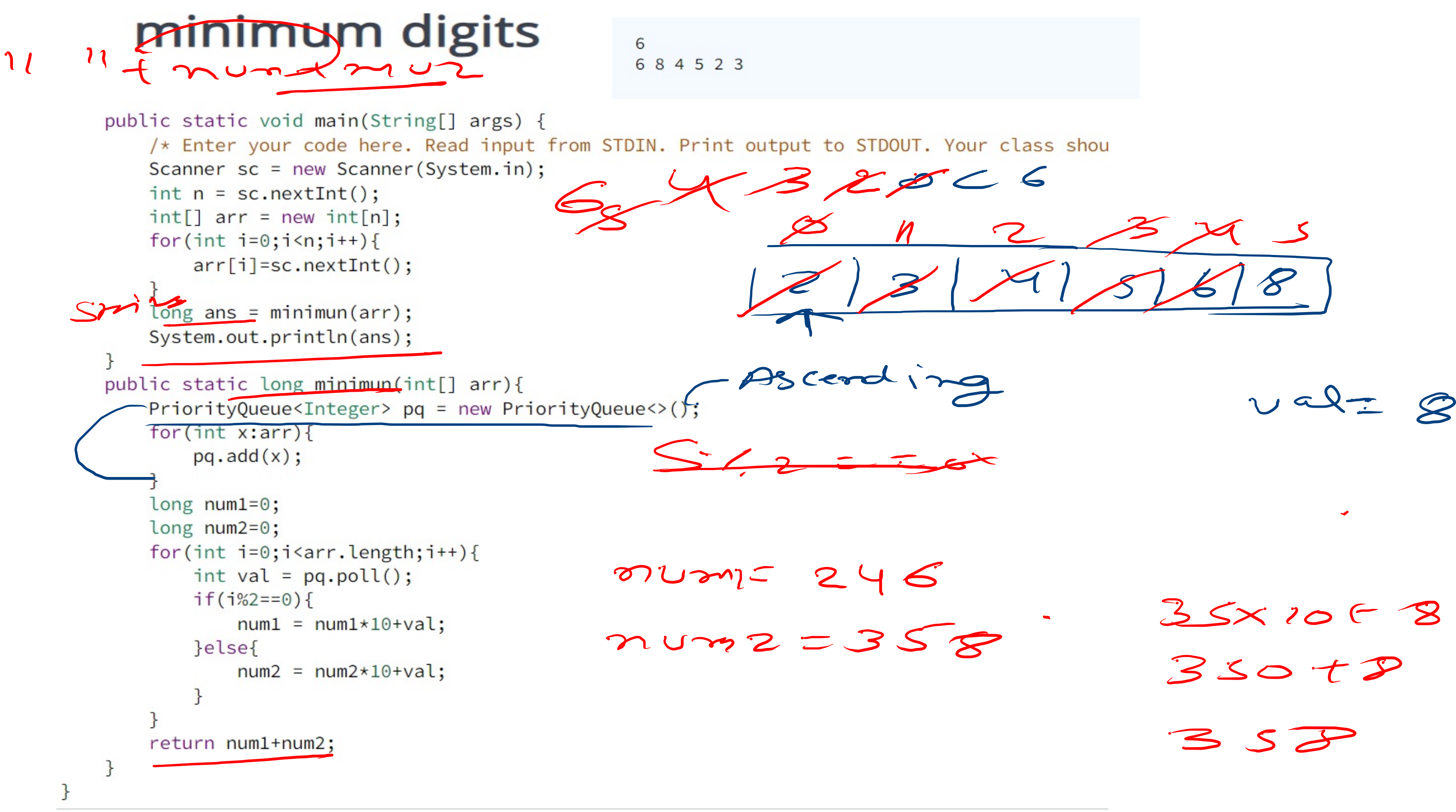
num2 = 3 5 8

$$\begin{array}{r} {}^{1}\,{}^{1} \\ 2\;4\;6 \\ 3\;5\;8 \\ \hline \boxed{6\;0\;4} \end{array}$$

0×10 + 2

2×10 + 4

24×10 + 6 = $\boxed{246}$



minimum digits

"" "" + num1 num2

```
6
6 8 4 5 2 3
```

```
    public static void main(String[] args) {
        /* Enter your code here. Read input from STDIN. Print output to STDOUT. Your class shou
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
        int[] arr = new int[n];
        for(int i=0;i<n;i++){
            arr[i]=sc.nextInt();
        }
        long ans = minimun(arr);
        System.out.println(ans);
    }
    public static long minimun(int[] arr){
        PriorityQueue<Integer> pq = new PriorityQueue<>();
        for(int x:arr){
            pq.add(x);
        }
        long num1=0;
        long num2=0;
        for(int i=0;i<arr.length;i++){
            int val = pq.poll();
            if(i%2==0){
                num1 = num1*10+val;
            }else{
                num2 = num2*10+val;
            }
        }
        return num1+num2;
    }
}
```

String

~~6~~ ~~8~~ ~~4~~ ~~3~~ ~~2~~ ~~0~~ < 6

~~8~~ n 2 ~~3~~ ~~4~~ 5

~~2~~	~~3~~	~~4~~	~~5~~	~~6~~	8

Ascending

~~Size = 6~~

val = 8

num1 = 246

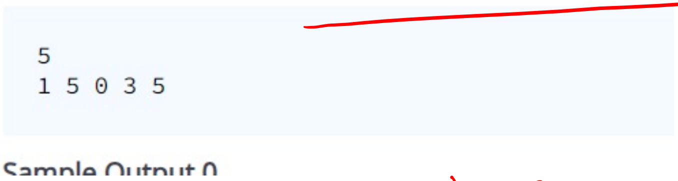
num2 = 358

35×10+8

350+8

358

Hasnset



```
5
1 5 0 3 5
```

Sample Output 0

1 5 0 3 5

⇓

[1, 5, 0, 3, 5]

⇓

[1 5, 3, 5]

3 [0 4 2 4]

⇓

[4, 2, 4]

2 [2 0 2]

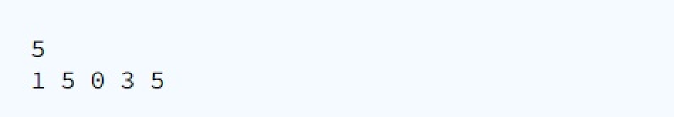
3 [2, 2] ✓

[0, 0]

[1, 5, 3] ✓ 3

↓

syso(hs.size());

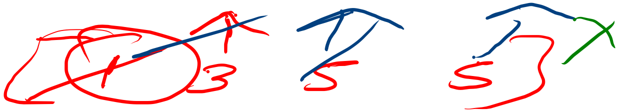


```
5
1 5 0 3 5
```

Hashset

↓

Ascending



int opercnt = 0 , int last = ~~0~~ ~~1~~ 3 5

while(size > 0) {

3 == ~~0~~

count = ~~1~~ 2

3

5 ~~== 5~~

return count

3

1

4

{1, 3, 5, 5, 5, 5}

0

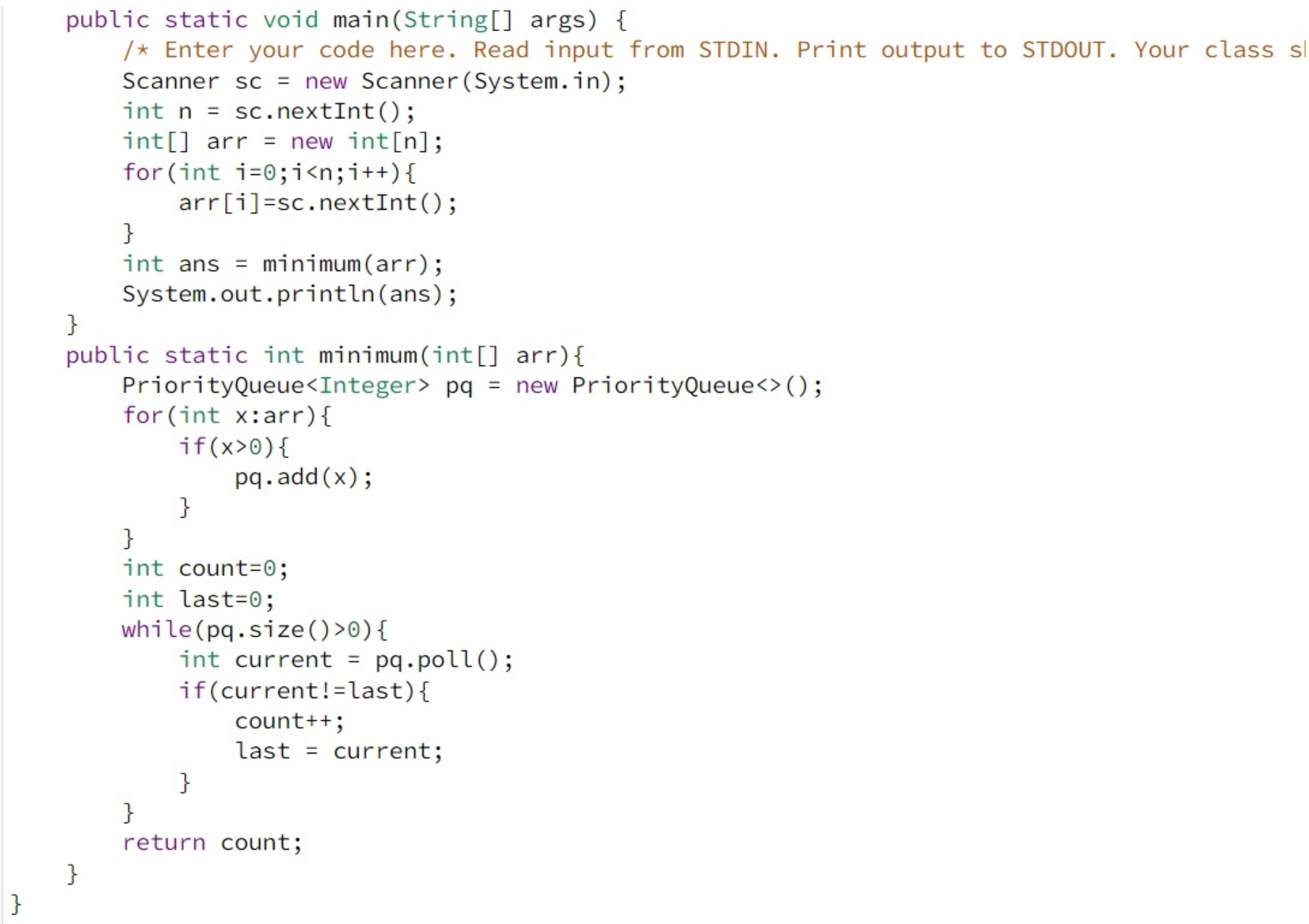
{0, 2, 4, 4, 4, 4}

{0, 2, 2, 2, 2, 2}

i < ( ) 0

{3}

subtract numbers 1

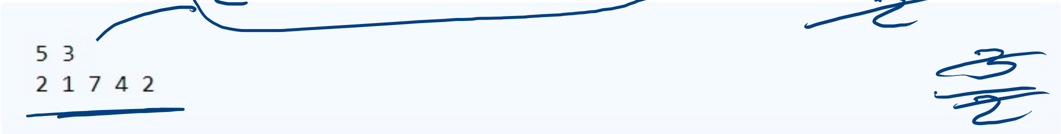


```
    public static void main(String[] args) {
        /* Enter your code here. Read input from STDIN. Print output to STDOUT. Your class s
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
        int[] arr = new int[n];
        for(int i=0;i<n;i++){
            arr[i]=sc.nextInt();
        }
        int ans = minimum(arr);
        System.out.println(ans);
    }
    public static int minimum(int[] arr){
        PriorityQueue<Integer> pq = new PriorityQueue<>();
        for(int x:arr){
            if(x>0){
                pq.add(x);
            }
        }
        int count=0;
        int last=0;
        while(pq.size()>0){
            int current = pq.poll();
            if(current!=last){
                count++;
                last = current;
            }
        }
        return count;
    }
}
```

k minmove

$\frac{4}{2}$ $\boxed{2}$

$\frac{3}{2}$ $\boxed{1.5} \Rightarrow \boxed{1}$



```
5 3
2 1 7 4 2
```

[2 1 7 4 2]

int ans = 0 + 7 = 7 + 4

= 11

11 + 3 = $\boxed{14}$

[2 1 3 4 2]

[2 1 3 2 2]

[2 ? 1 2 2]

maximum diamonds

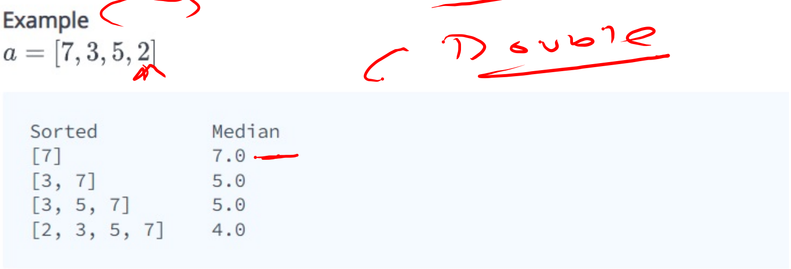
```
public class Solution {

    public static void main(String[] args) {
        /* Enter your code here. Read input from STDIN. Print output to STDOUT. Your class should be name
         Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
        int k = sc.nextInt();
        int[] arr = new int[n];
        for(int i=0;i<n;i++){
            arr[i]=sc.nextInt();
        }
        int ans  = maxDiamonds(arr,k);
        System.out.println(ans);
    }
    public static int maxDiamonds(int[] arr,int k){
        PriorityQueue<Integer> pq = new PriorityQueue<>(Collections.reverseOrder());
        for(int x:arr){
            pq.add(x);
        }
        int diamonds=0;
        for(int i=0;i<k;i++){
            int max = pq.poll();
            diamonds+=max;
            pq.add(max/2);
        }
        return diamonds;
    }
}
```

Example

$a = [7, 3, 5, 2]$

```
Sorted          Median
[7]             7.0
[3, 7]          5.0
[3, 5, 7]       5.0
[2, 3, 5, 7]    4.0
```



Example

$a = [7, 3, 5, 2]$

```
Sorted          Median
[7]             7.0
[3, 7]          5.0
[3, 5, 7]       5.0
[2, 3, 5, 7]    4.0
```

{3, 7}

(Double

2.0

4.0

[1, 2, 3, 5, 7]

{7, 3, 5, 2}

$[7] \rightarrow 7.0$

$\{3 \quad 7\} \rightarrow 5.0$

$[3, 5, 7] \rightarrow 5.0$

$\{2, 3, 5, 7\} \rightarrow 4.0$

$\cancel{\frac{5}{2}}$

$2.5 \quad \frac{3+7}{2} = \cancel{\frac{10}{2}}$

5.0

$\frac{3+5}{2}$

$\cancel{\frac{8}{2}}$ 4.0

[2, 3] [5, 7]

12.0
8.0
5.0

[3, 2] [5, 7]

min head

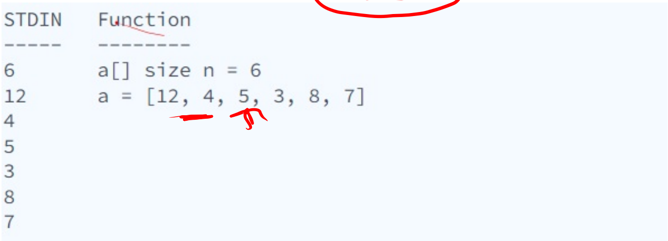
max head

$3 + 5 = \frac{8}{2} = 4.0$

13

PQ1

PQ2



```
STDIN   Function
-----   --------
6       a[] size n = 6
12      a = [12, 4, 5, 3, 8, 7]
4
5
3
8
7
```

min	max
~~2~~ 2	
4	12
5	

12+4

$\frac{16}{2} = 8.0$

8.0
~~5.0~~

$\boxed{4 + 5} \rightarrow 9$

$\boxed{5 + 4} \rightarrow 9$