The painter

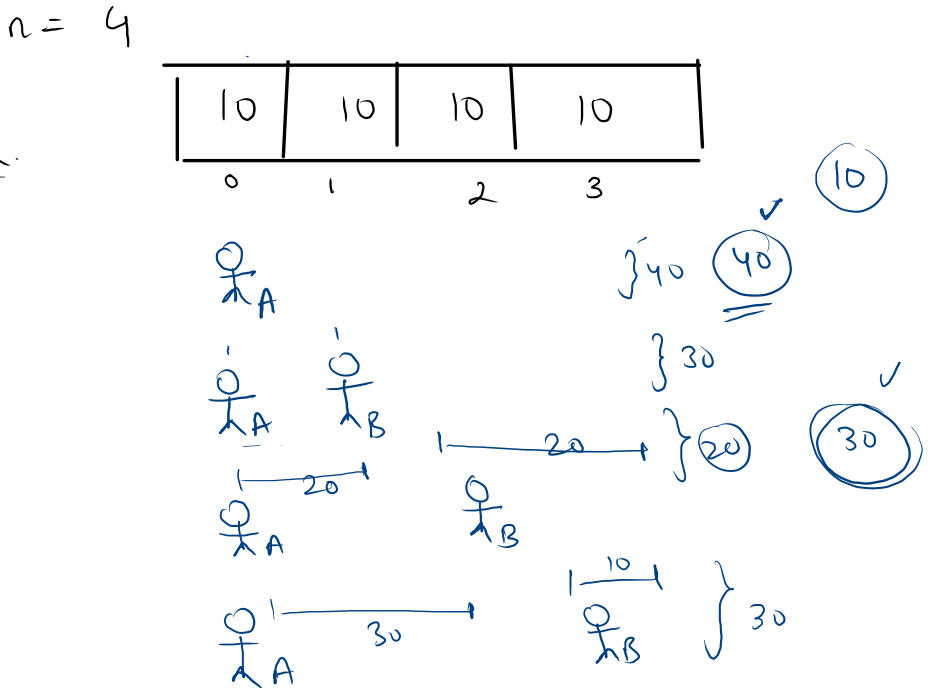
We have to paint n boards of length {A1, A2...An}. There are k painters available and each takes 1 unit of time to paint 1 unit of the board. The problem is to find the minimum time to get this job was done under the constraints that any painter will only paint continuous sections of boards, say board {2, 3, 4} or only board {1} or nothing but not board {2, 4, 5}.

Sample Input 0

```
4
10 10 10 10
2
```

Sample Output 0

```
20
```

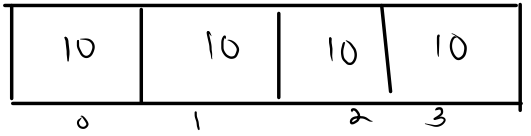


A B C D

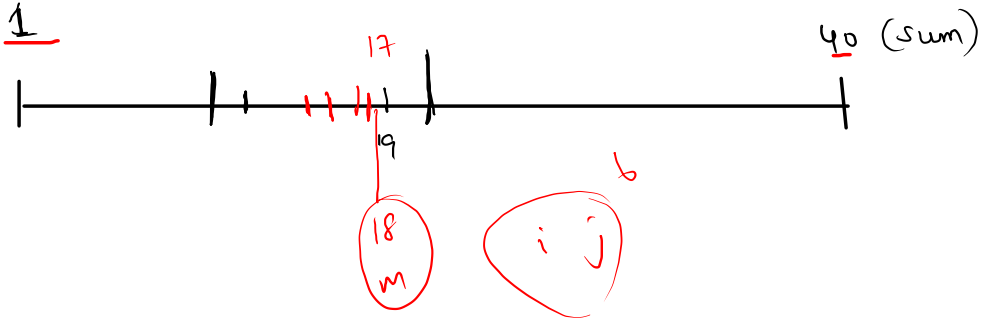
✓ ✓

$k = 2$

total painter available



minimum time

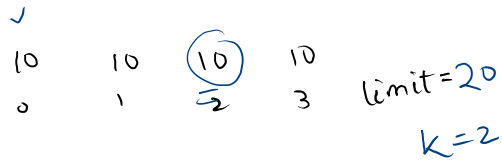


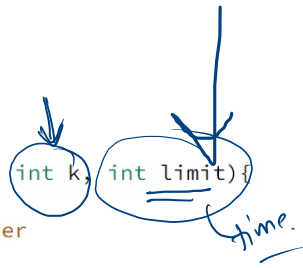
Avg = 20

```java
import java.io.*;
import java.util.*;

public class Solution {
    public static boolean isPossible(int [] A , int k, int limit){

    }
    public static void main(String[] args) {
        Scanner scn = new Scanner(System.in);
        int n = scn.nextInt();
        int [] A = new int[n];
        int sum = 0;

        for(int i = 0; i < n; i++){
            A[i] = scn.nextInt();
            sum += A[i];
        }
        int k = scn.nextInt();
        int i = 1;
        int j = sum;
        int ans = -1;

        while(i <= j){
            int m = (i + j)/2;
            if(isPossible(A, k, m)){
                ans = m;
                j = m - 1;
            }else{
                i = m + 1;
            }
        }
        System.out.println(ans);
    }
}
```

Press F11 to exit full screen

```java
public static boolean isPossible(int [] A , int k, int limit){
    int painterNeeded = 1;
    int work = 0;   //work of current painter
    for(int ele : A){
        if(work + ele <= limit){
            work += ele;
        }else{
            painterNeeded++;
            work = ele;
        }
    }
    return painterNeeded <= k;
}
```

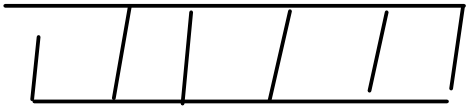




Arraylist ⟶ Data Structure

↳ Dynamic Array :→ size is not fixed.

```
int [ ] A = new int[5];
```



How arraylist is different from array?

Array List

initialize ✓

```
//init
ArrayList<Integer> arr = new ArrayList<>();
```

add

```
arr.add(20);
arr.add(30);
```

```
arr.add(0,40);
```

idx

remove

```
13          arr.remove(1);
14
```

get

```
System.out.println(arr.get(2));
```

traverse { print }

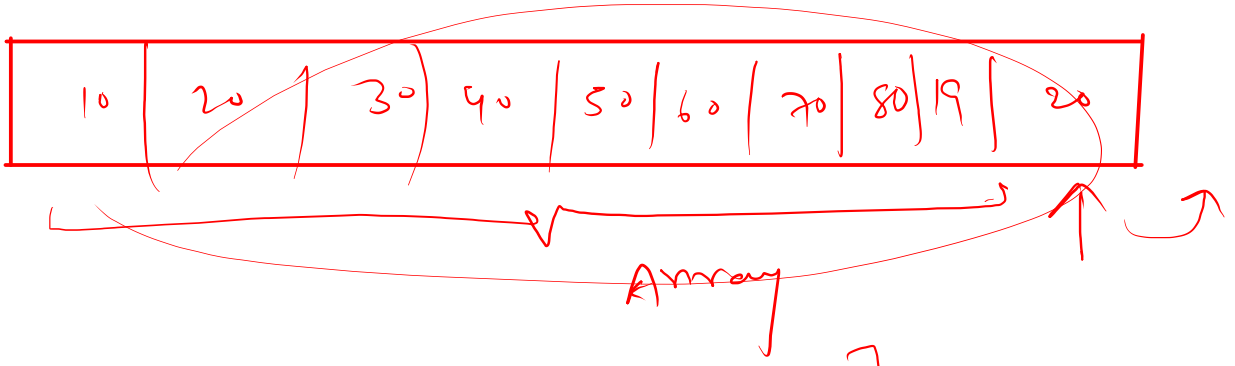
```
System.out.println(arr);
```

```
[10, 20, 30, 40]
```

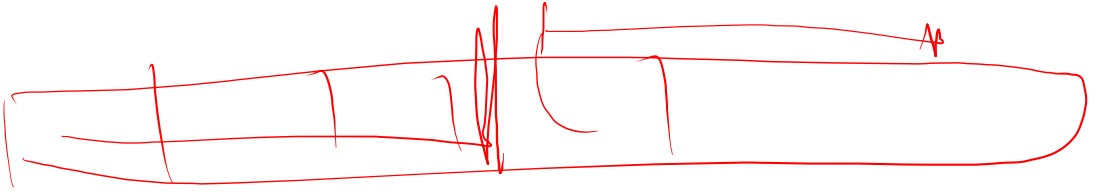
size ✓

```
System.out.println(arr.size());
```

static.



A = new int [A.length X 2]



Arraylist → if / else.

- **First Declare an ArrayList arr**.
- Then take T as an Integer input.

Format for next T Lines : $(case, x(optional))$

- **case 1**: Print the size of the ArrayList in a separate line.
- **case 2**: Print and Remove element from the last index of the ArrayList.
- **case 3**: Print x and Add x in last index of the ArrayList.
- **case 4**: Print and Remove an element from the starting (index = 0) of the ArraList.
- **case 5**: Print x and Add x at beginning (index = 0) of the ArrayList.
- **case 6**: Print all the elements from left to right that are there inside the ArrayList.

Note : In **case** $2, 4, 6$ when arr is **empty** the move is invalid, so print "**invalid-move** all lowercase".

Input Format

- Single Integer T.

Sample Input 0

```
8
2
6
3 2
5 1
6
1
3 3
2
```

Sample Output 0

```
invalid-move
invalid-move
2
1
1 2
2
3
3
```

```java
import java.io.*;
import java.util.*;
public class Solution {
    public static void main(String[] args) {
        Scanner scn = new Scanner(System.in);
        int T = scn.nextInt();
        ArrayList<Integer> arr = new ArrayList<>();
        while(T-- > 0){          // T times
            int caseNu = scn.nextInt();
            if(caseNu == 1){
                System.out.println(arr.size());
            }else if(caseNu == 2){
                if(arr.size() == 0){
                    System.out.println("invalid-move");
                }else{
                    System.out.println(arr.remove(arr.size()-1));
                }
            }else if(caseNu == 3){
                int x = scn.nextInt();
                System.out.println(x);
                arr.add(x);
            }else if(caseNu == 4){
                if(arr.size() == 0){
                    System.out.println("invalid-move");
                }else{
                    System.out.println(arr.remove(0));
                }
            }else if(caseNu == 5){
                int x = scn.nextInt();
                arr.add(0,x);
                System.out.println(x);
            }else{
                if(arr.size() == 0){
                    System.out.println("invalid-move");
                }else{
                    for(int e : arr){
                        System.out.print(e + " ");
                    }
                    System.out.println();
                }
            }
        }
    }
}
```

You are s

ArrayList Printing

for loop
for each loop.

Sample Input 0

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

Sample Output 0

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

```java
import java.io.*;
import java.util.*;

public class Solution {

    public static void main(String[] args) {
        Scanner scn = new Scanner(System.in);
        int n = scn.nextInt();
        ArrayList<Integer> arr = new ArrayList<>();
        while(n-- > 0){
            arr.add(scn.nextInt());
        }
        //for loop
        for(int i = 0; i < arr.size(); i++){
            System.out.print(arr.get(i) + " ");
        }
        System.out.println();
        //for each loop
        for(int e : arr){
            System.out.print(e + " ");
        }
    }
}
```