

You are given ' n ' roses and you are also given an array ' arr ' where ' $arr[i]$ ' denotes that the ' i^{th} ' rose will bloom on the ' $arr[i]$ 'th day.

You can only pick already bloomed roses that are adjacent to make a bouquet. You are also told that you require exactly ' k ' adjacent bloomed roses to make a single bouquet.

Find the minimum number of days required to make at least ' m ' bouquets each containing ' k ' roses. Return -1 if it is not possible.

Example :

Input: $n = 9, arr = [1, 2, 1, 2, 7, 2, 2, 3, 1], k = 3, m = 2$

Output: 3.

Explanation: This is because on the 3rd day: all the roses with ' $arr[i]$ ' less than equal to 3 have already bloomed, this means every rose except the 5th rose has bloomed. Now we can form the first bouquet from the first three roses and the second bouquet from the last three roses.

Detailed explanation (Input/output format, Notes, Images)

Sample Input 1 :

```
9
1 2 1 2 7 2 2 3 1
3 2
```

Sample Output 1 :

```
3
```

Explanation For Sample Input 1 :

We will return 3, because:
All the roses with ' $arr[i]$ ' less than equal to 3 have already bloomed after 3 days, this means every rose except the 5th rose has bloomed. Now we can form the first bouquet from the first three roses and the second bouquet from the last three roses.

Sample Input 2 :

```
4
1 1 1 1
1 1
```

Sample Output 2 :

```
1
```



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

$m = 2$
↓
least no of bouquets
 $k = 3,$
↑
adjacent
bloomed roses required.

Day 1 : Bloomed roses

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

Day 2 : Bloomed roses

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

1st bouquet

Day 3 : Bloomed roses

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

2nd bouquet

Day 4 : Bloomed roses

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

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

Ans must lie b/w min & max.

start end
 ↑ ↑
1 7
 mid

Day 4 :

1	2	1	2		2	2	3	1
---	---	---	---	--	---	---	---	---

1 2 = m.

ans = 4

end = mid - 1

start end
1 2 3
 ↑
 mid.

Day 2 :

1	2	1	2		2	2		1
---	---	---	---	--	---	---	--	---

only 1

start = mid + 1

start
↓
3 ← mid
 ↑
end

Day 3 :

1	2	1	2		2	2	3	1
---	---	---	---	--	---	---	---	---

1st 2nd

ans = 3

start = mid + 1

goes out of while loop

