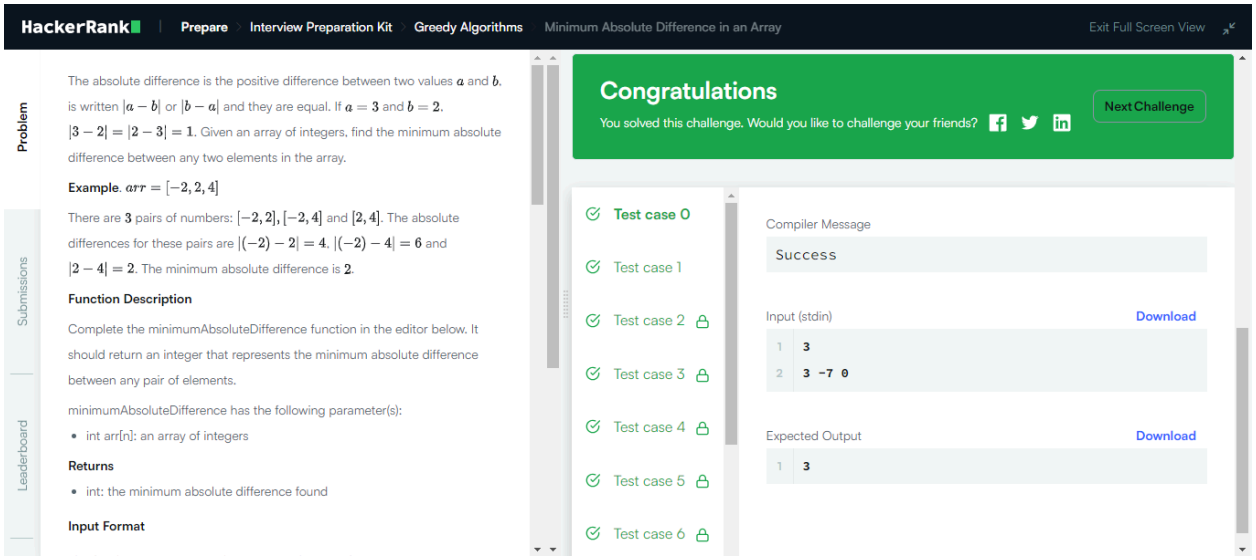
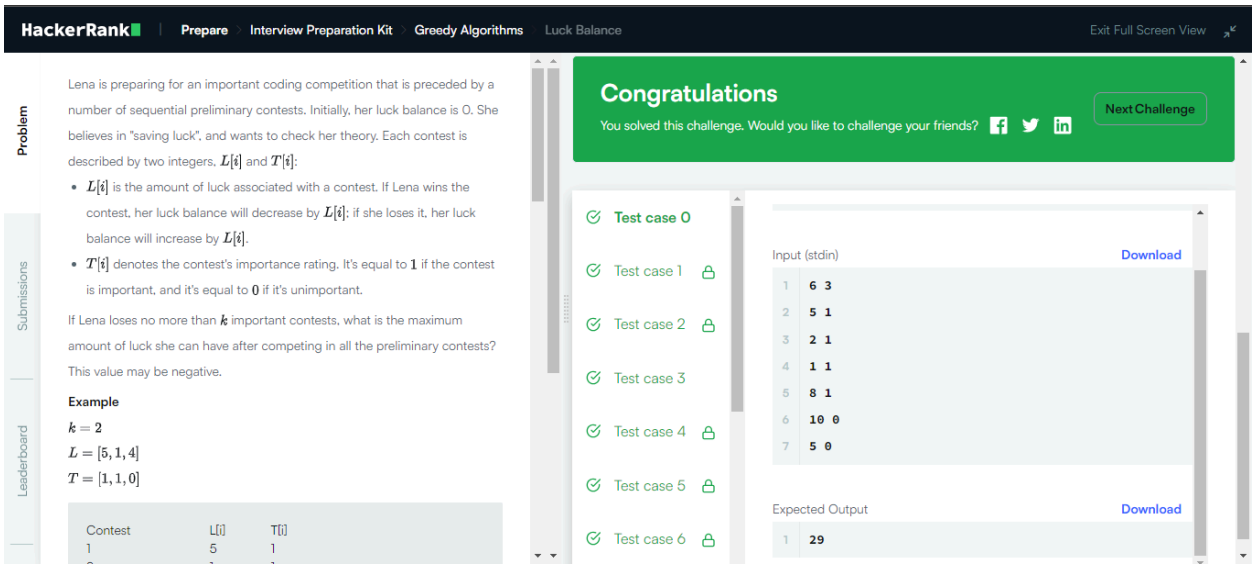



```
def minimumAbsoluteDifference(arr):
    # Write your code here
    arr.sort()
    return min(abs(arr[i]-arr[i+1]) for i in range(n-1))
```

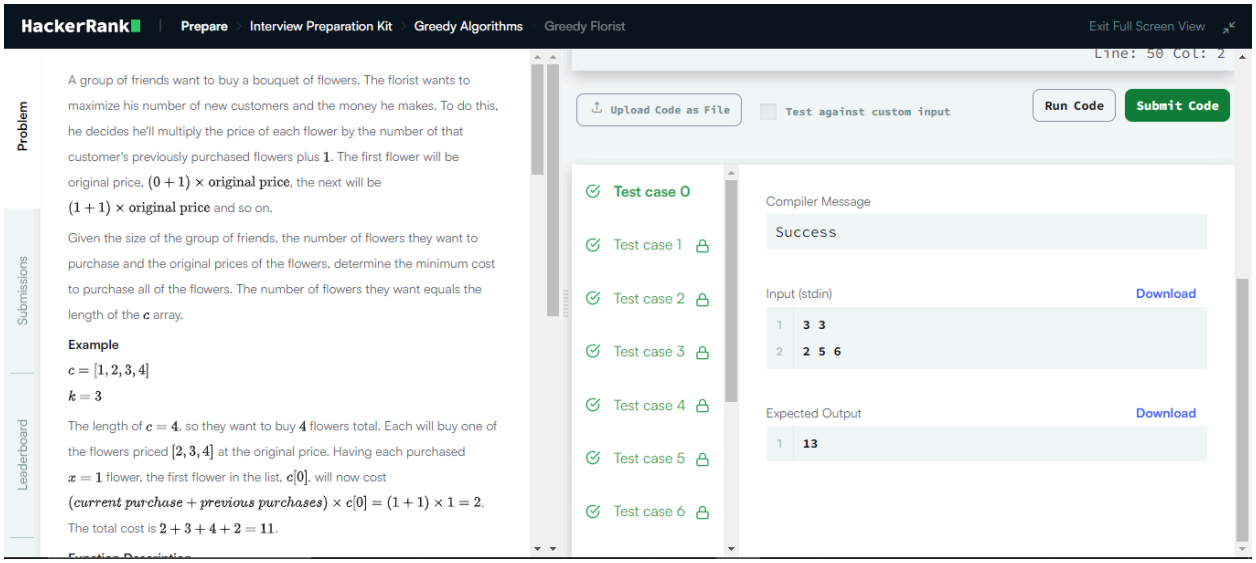
The code above first sorts the array and then returns the minimum absolute difference between two elements in the array. The code uses a greedy approach by assuming that sorting the array will lead to the smallest absolute difference residing between adjacent elements.




```
def luckBalance(k, contests):
    # Write your code here
```

```
    impcontests = sorted([x[0] for x in contests if x[1] == 1])
    unimpcontests = [x[0] for x in contests if x[1] == 0]
    l = max(0, len(impcontests) - k)
    return -sum(impcontests[:l]) + sum(impcontests[l:]) +
sum(unimpcontests)
```

The code above first sorts the array with what are considered the important contests and then calculates the losses and gains from each important contest and the losses and gains of each unimportant contest. Lastly, the algorithm calculates the final luck balance.




```
def getMinimumCost(k, c):
    return sum([(i // k + 1) * cost for i, cost in
enumerate(sorted(c, reverse=True))])
```

The code above first sorts the array, calculates each flower's cost, and then sums up the individual flower costs.

1. Which problems did you attempt to solve?
    - Minimum Absolute Difference in an Array
    - Luck Balance
    - Greedy Florist
2. Which was the easiest problem to solve?
    - The easiest problem to solve was the Minimum Absolute Difference in an Array problem.
3. Which was the most difficult problem to solve?

    - The most difficult problem to solve among the 3 I did was the Greedy Florist problem.
4. How did your understanding of the greedy paradigm evolve after solving this problem set? How "proficient" would you consider yourself in this paradigm? Did you gain any interesting insights about the paradigm that became apparent after this problem set?
    - After solving these problems, my understanding of the greedy paradigm has further increased. One prominent, interesting thing I noticed in my solution to the problems was that I sorted each array before finding the optimal solution to each problem. If I rate my proficiency in this paradigm on a scale from 1 to 10, it would be a 6.