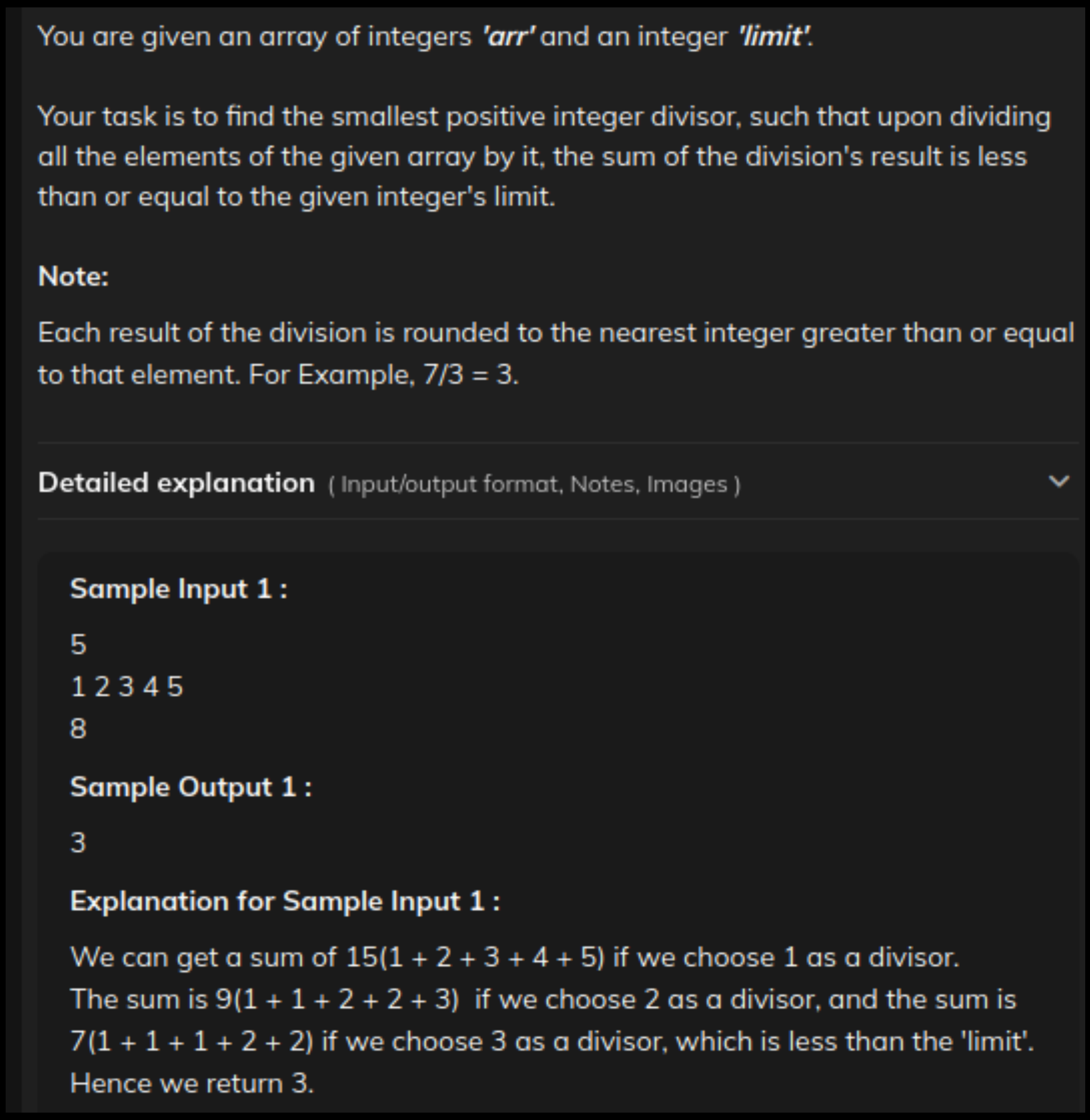
Smallest Divisor



You are given an array of integers *'arr'* and an integer *'limit'*.

Your task is to find the smallest positive integer divisor, such that upon dividing all the elements of the given array by it, the sum of the division's result is less than or equal to the given integer's limit.

Note:

Each result of the division is rounded to the nearest integer greater than or equal to that element. For Example, 7/3 = 3.

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

Sample Input 1 :

```
5
1 2 3 4 5
8
```

Sample Output 1 :

```
3
```

Explanation for Sample Input 1 :

We can get a sum of 15(1 + 2 + 3 + 4 + 5) if we choose 1 as a divisor.
The sum is 9(1 + 1 + 2 + 2 + 3) if we choose 2 as a divisor, and the sum is 7(1 + 1 + 1 + 2 + 2) if we choose 3 as a divisor, which is less than the 'limit'.
Hence we return 3.

→ Brute force :-

↳ Our answer will definetly lie b/w 1 & the max value of array.

↳ So simply run a loop from 1 → max(arr) and will return the value of the number whose division's result sum <= limit.

```
bool isPossible(vector<int> arr, int limit, int divisor) {
  int sum = 0;

  for (int i = 0; i < arr.size(); i++) {
    sum += ceil(arr[i] / (double)divisor);
  }
  return sum <= limit ? true : false;
}

int smallestDivisor(vector<int> &arr, int limit) {
  int maxi = 0, ans;
  for (int i = 0; i < arr.size(); i++) {
    maxi = maxi > arr[i] ? maxi : arr[i];
  }

  for (int i = 1; i <= maxi; i++) {
    if (isPossible(arr, limit, i)) {
      ans = i;
      break;
    } else
      continue;
  }

  return ans;
}
```

→ Optimal Approach :-

```
for (int i = 1; i <= maxi; i++) {
  if (isPossible(arr, limit, i)) {
    ans = i;
    break;
  } else
    continue;
}
```

→ We need to optimize this to logn using binary search.

```
bool isPossible(vector<int> arr, int limit, int divisor) {
  int sum = 0;

  for (int i = 0; i < arr.size(); i++) {
    sum += ceil(arr[i] / (double)divisor);
  }
  return sum <= limit ? true : false;
}

int smallestDivisor(vector<int>& arr, int limit)
{
        int start = 1, end = 1000000;
        int ans;
        while(start <= end) {
                int mid = (start + end) >> 1;
                if(isPossible(arr, limit, mid)) {
                        ans = mid;
                        end = mid - 1;
                } else start = mid + 1;
        }
        return ans;
}
```