

Name : CH. SRIKANTH

ID NO : 2300032366

Section : 31

1. Permuting Two Arrays

Permuting Two Arrays

Problem	Submissions	Leaderboard	Discussions
---------	-------------	-------------	-------------

Submitted 3 minutes ago • Score: 10.00

Status: **Accepted**

✓	Test Case #0	✓	Test Case #1	✓	Test Case #2
✓	Test Case #3	✓	Test Case #4	✓	Test Case #5
✓	Test Case #6	✓	Test Case #7	✓	Test Case #8
✓	Test Case #9	✓	Test Case #10		

```
import math
```

```
import os
```

```
import random
```

```
import re
```

```
import sys
```

```
def twoArrays(k, A, B):
```

```
    A.sort()
```

```
    B.sort(reverse=True)
```

```
    if any(a + b < k for (a, b) in zip(A, B)):
```

```
        return "NO"
```

```
    return "YES"
```

```

if __name__ == '__main__':
    fptr = open(os.environ['OUTPUT_PATH'], 'w')
    q = int(input().strip())
    for q_itr in range(q):
        first_multiple_input = input().rstrip().split()
        n = int(first_multiple_input[0])
        k = int(first_multiple_input[1])
        A = list(map(int, input().rstrip().split()))
        B = list(map(int, input().rstrip().split()))
        result = twoArrays(k, A, B)
        fptr.write(result + '\n')
    fptr.close()

```

2. Jim and the Orders

Jim and the Orders

Problem	Submissions	Leaderboard	Discussions
Submitted 12 minutes ago • Score: 10.00			
Status: Accepted			
✓ Test Case #0	✓ Test Case #1	✓ Test Case #2	
✓ Test Case #3	✓ Test Case #4	✓ Test Case #5	
✓ Test Case #6	✓ Test Case #7	✓ Test Case #8	
✓ Test Case #9	✓ Test Case #10		

```
import math
```

```
import os
```

```
import random
```

```
import re
```

```
import sys
```

```
def jimOrders(orders):  
    serve_time = [sum(order) for order in orders]  
    indexed = sorted(enumerate(serve_time, 1), key=lambda x: x[1])  
    cust_order = [index[0] for index in indexed]  
  
    return cust_order  
  
if __name__ == '__main__':  
    fptr = open(os.environ['OUTPUT_PATH'], 'w')  
  
    n = int(input().strip())  
  
    orders = []  
  
    for _ in range(n):  
        orders.append(list(map(int, input().rstrip().split())))  
  
    result = jimOrders(orders)  
  
    fptr.write(' '.join(map(str, result)))  
    fptr.write('\n')  
  
    fptr.close()
```

3. Mark and Toys

Mark and Toys

Problem	Submissions	Leaderboard	Discussions
Submitted 16 minutes ago • Score: 10.00			
Status: Accepted			
✓	Test Case #0	✓	Test Case #1
✓	Test Case #3	✓	Test Case #4
✓	Test Case #6	✓	Test Case #7
✓	Test Case #9	✓	Test Case #10
✓	Test Case #12	✓	Test Case #13
✓	Test Case #15	✓	Test Case #16
✓		✓	Test Case #2
		✓	Test Case #5
		✓	Test Case #8
		✓	Test Case #11
		✓	Test Case #14
		✓	Test Case #17

```
import math
```

```
import os
```

```
import random
```

```
import re
```

```
import sys
```

```
def maximumToys(rates, x):
```

```
    rates.sort()
```

```
    for i in range(len(rates)):
```

```
        if x - rates[i] >= 0:
```

```
            x -= rates[i]
```

```
        else:
```

```
            return i
```

```
if __name__ == '__main__':
```

```
    fptr = open(os.environ['OUTPUT_PATH'], 'w')
```

```
    first_multiple_input = input().rstrip().split()
```

```
    n = int(first_multiple_input[0])
```

```
    x = int(first_multiple_input[1])
```

```

prices = list(map(int, input().rstrip().split()))
result = maximumToys(prices, x)
fptr.write(str(result) + '\n')
fptr.close()

```

4. Priyanka and Toys

Priyanka and Toys

Problem	Submissions	Leaderboard	Discussions
Submitted 19 minutes ago • Score: 10.00			
Status: Accepted			
✓ Test Case #0	✓ Test Case #1	✓ Test Case #2	
✓ Test Case #3	✓ Test Case #4	✓ Test Case #5	
✓ Test Case #6	✓ Test Case #7	✓ Test Case #8	
✓ Test Case #9	✓ Test Case #10	✓ Test Case #11	
✓ Test Case #12			

```
import math
```

```
import os
```

```
import random
```

```
import re
```

```
import sys
```

```
def toys(w):
```

```
    w.sort()
```

```
    no_of_con = 0
```

```
    i = 0
```

```
    while i < len(w):
```

```
        no_of_con += 1
```

```
        con_limit = w[i] + 4
```

```
        while i < len(w) and w[i] <= con_limit:
```

```

        i += 1

    return no_of_con

if __name__ == '__main__':
    fptr = open(os.environ['OUTPUT_PATH'], 'w')
    n = int(input().strip())
    w = list(map(int, input().rstrip().split()))
    result = toys(w)
    fptr.write(str(result) + '\n')
    fptr.close()

```

5. Greedy Florist

Greedy Florist

Problem	Submissions	Leaderboard	Discussions
Submitted 16 minutes ago • Score: 10.00			
Status: Accepted			
✓ Test Case #0	✓ Test Case #1	✓ Test Case #2	
✓ Test Case #3	✓ Test Case #4	✓ Test Case #5	
✓ Test Case #6	✓ Test Case #7	✓ Test Case #8	
✓ Test Case #9	✓ Test Case #10	✓ Test Case #11	

```

import math
import os
import random
import re
import sys

def getMinimumCost(k, c):
    n_flowers = len(c)
    c = sorted(c, reverse=True)
    price = 0

```

```
count = 0
for i in range(n_flowers):
    if i % k == 0 and i != 0 :
        count += 1
    price += c[i] * (count + 1)
return price
```

```
if __name__ == '__main__':
    fptr = open(os.environ['OUTPUT_PATH'], 'w')
```

```
nk = input().split()
```

```
n = int(nk[0])
```

```
k = int(nk[1])
```

```
c = list(map(int, input().rstrip().split()))
```

```
minimumCost = getMinimumCost(k, c)
```

```
fptr.write(str(minimumCost) + '\n')
```

```
fptr.close()
```