

[Dashboard](#) / [My courses](#) / [CS23331-DAA-2023-CSE](#) / [Greedy Algorithms](#) / [2-G-Cookies Problem](#)

Started on	Tuesday, 27 August 2024, 2:27 PM
State	Finished
Completed on	Tuesday, 3 September 2024, 1:38 PM
Time taken	6 days 23 hours
Marks	1.00/1.00
Grade	10.00 out of 10.00 (100%)

Question 1

Correct

Mark 1.00 out of 1.00

Assume you are an awesome parent and want to give your children some cookies. But, you should give each child at most one cookie.

Each child i has a greed factor $g[i]$, which is the minimum size of a cookie that the child will be content with; and each cookie j has a size $s[j]$. If $s[j] \geq g[i]$, we can assign the cookie j to the child i , and the child i will be content. Your goal is to maximize the number of your content children and output the maximum number.

Example 1:**Input:**

```
3
1 2 3
2
1 1
```

Output:

```
1
```

Explanation: You have 3 children and 2 cookies. The greed factors of 3 children are 1, 2, 3.

And even though you have 2 cookies, since their size is both 1, you could only make the child whose greed factor is 1 content.

You need to output 1.

Constraints:
 $1 \leq g.length \leq 3 \times 10^4$
 $0 \leq s.length \leq 3 \times 10^4$
 $1 \leq g[i], s[j] \leq 2^{31} - 1$
Answer: (penalty regime: 0 %)

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 int compare(const void *a, const void *b) {
4     return (*(int*)a - *(int*)b);
5 }
6 int main() {
7     int g, s;
8     scanf("%d", &g);
9     int a[g];
10    for (int i = 0; i < g; i++) scanf("%d", &a[i]);
11    scanf("%d", &s);
12    int b[s];
13    for (int i = 0; i < s; i++) scanf("%d", &b[i]);
14    qsort(a, g, sizeof(int), compare);
15    qsort(b, s, sizeof(int), compare);
16    int child = 0, cookie = 0;
17    while (child < g && cookie < s) {
18        if (b[cookie] >= a[child]) child++;
19        cookie++;
20    }
21    printf("%d\n", child);
22    return 0;
23 }
24
```

	Input	Expected	Got	
✓	2	2	2	✓
	1 2			
	3			
	1 2 3			

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

[◀ 1-G-Coin Problem](#)

Jump to...

[3-G-Burger Problem ▶](#)