



## 2-G-Cookies Problem

<b>Started on</b>	Tuesday, 30 September 2025, 11:59 AM
<b>State</b>	Finished
<b>Completed on</b>	Tuesday, 30 September 2025, 12:00 PM
<b>Time taken</b>	1 min 36 secs
<b>Marks</b>	1.00/1.00
<b>Grade</b>	<b>10.00</b> out of 10.00 ( <b>100%</b> )

**Question 1** | Correct   Mark 1.00 out of 1.00   [Flag question](#)

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:**

```
4
```

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:**

$1 \leq \text{g.length} \leq 3 \cdot 10^4$

$0 \leq \text{s.length} \leq 3 \cdot 10^4$

$1 \leq \text{g}[i], \text{s}[j] \leq 2^{31} - 1$

**Answer:** (penalty regime: 0 %)

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int cmp(const void *a, const void *b) {
5     int x = *(int*)a;
6     int y = *(int*)b;
7     return (x > y) - (x < y);
8 }
9
10 int main() {
11     int n, m;
12     scanf("%d", &n);
13     int g[n];
14     for (int i = 0; i < n; i++) scanf("%d", &g[i]);
15
16     scanf("%d", &m);
17     int s[m];
18     for (int i = 0; i < m; i++) scanf("%d", &s[i]);
19
20     qsort(g, n, sizeof(int), cmp);
21     qsort(s, m, sizeof(int), cmp);
22
23     int i = 0, j = 0, count = 0;
24
25     while (i < n && j < m) {
26         if (s[j] >= g[i]) {
27             count++;
28             i++;
29             j++;
30         } else {
31             j++;
32         }
33     }
34
35     printf("%d\n", count);
36     return 0;
37 }
38
```

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

Passed all tests! ✓

**Correct**

Marks for this submission: 1.00/1.00.

[Finish review](#)

[Back to Course](#)

[Data retention summary](#)