

SRIRAM S (12/08/2006) 2024-IT**S2****Started on** Sunday, 7 September 2025, 9:15 AM**State** Finished**Completed on** Sunday, 7 September 2025, 9:16 AM**Time taken** 34 secs**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:**

$1 \leq g.length \leq 3 * 10^4$

$0 \leq s.length \leq 3 * 10^4$

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

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

```

1 #include <stdio.h>
2 #include <stdlib.h>
3
4 // Comparison function for ascending sort
5 int compare(const void *a, const void *b) {
6     return (*(int *)a - *(int *)b);
7 }
8
9 int main() {
10     int n, m;
11     scanf("%d", &n); // Number of children
12
13     int g[n]; // Greed factors
14     for (int i = 0; i < n; i++) {
15         scanf("%d", &g[i]);
16     }
17
18     scanf("%d", &m); // Number of cookies
19
20     int s[m]; // Cookie sizes
21     for (int i = 0; i < m; i++) {
22         scanf("%d", &s[i]);
23     }
24
25     // Sort greed factors and cookie sizes
26     qsort(g, n, sizeof(int), compare);
27     qsort(s, m, sizeof(int), compare);
28
29     int i = 0, j = 0, content = 0;
30     ...
31 }
```

```
31 // Two-pointer greedy matching
32 while (i < n && j < m) {
33     if (s[j] >= g[i]) {
34         content++;
35         i++;
36         j++;
37     } else {
38         j++;
39     }
40 }
41 printf("%d\n", content); // Output the result
42
43 return 0;
44 }
45
46 }
```

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

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

[Back to Course](#)