



7

SIVA S 2024-CSE ▾

S2**Started on** Wednesday, 8 October 2025, 3:56 PM**State** Finished**Completed on** Wednesday, 8 October 2025, 3:56 PM**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:

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

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

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

Answer: (penalty regime: 0 %)

```
1 | #include <stdio.h>
2 | #include <stdlib.h>
3 |
4 | // Comparison function for qsort
5 | int compare(const void *a, const void *b) {
6 |     return (*(int *)a - *(int *)b);
7 | }
8 |
9 | int main() {
10 |     int gSize, sSize;
11 |     scanf("%d", &gSize);
12 |     int g[gSize];
13 |     for (int i = 0; i < gSize; i++) {
14 |         scanf("%d", &g[i]);
15 |     }
16 |
17 |     scanf("%d", &sSize);
18 |     int s[sSize];
19 |     for (int i = 0; i < sSize; i++) {
20 |         scanf("%d", &s[i]);
21 |     }
22 |
23 |     // Sort both arrays
24 |     qsort(g, gSize, sizeof(int), compare);
25 |     qsort(s, sSize, sizeof(int), compare);
26 |
27 |     int i = 0, j = 0, contentChildren = 0;
28 |
29 |     // Greedy assignment
30 |     while (i < gSize && j < sSize) {
31 |         if (s[j] >= g[i]) {
32 |             // Assign cookie j to child i
33 |             contentChildren++;
34 |             i++;
35 |             j++;
36 |         }
37 |     }
38 |
39 |     printf("%d", contentChildren);
40 |     return 0;
41 | }
```

	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](#)