1-G-Coin Problem

Done

Opened: Friday, 22 August 2025, 10:00 AM

Re-attempt quiz

Grading method: Highest grade

Summary of your previous attempts

Attempt	State	Marks / 1.00	Grade / 10.00	Review
1	Finished Submitted Tuesday, 26 August 2025, 7:53 PM	1.00	10.00	Review

Highest grade: 10.00 / 10.00.

Back to Course

2-G-Cookies Problem

Done

Opened: Friday, 22 August 2025, 10:00 AM

Re-attempt quiz

Attempts allowed: 3

Grading method: Highest grade

Summary of your previous attempts

Attempt	State	Marks / 1.00	Grade / 10.00	Review
1	Finished Submitted Tuesday, 26 August 2025, 8:37 PM	1.00	10.00	Review

Highest grade: 10.00 / 10.00.

Back to Course

3-G-Burger Problem

Done

Opened: Friday, 22 August 2025, 10:00 AM

Re-attempt quiz

Attempts allowed: 10

Grading method: Highest grade

Summary of your previous attempts

Attempt	State	Marks / 1.00	Grade / 10.00	Review
1	Finished Submitted Sunday, 7 September 2025, 11:25 AM	1.00	10.00	Review

Highest grade: 10.00 / 10.00.

Back to Course

4-G-Array Sum max problem

Started on	Sunday, 7 September 2025, 11:05 AM
State	Finished
Completed on	Sunday, 7 September 2025, 11:07 AM
Time taken	2 mins 28 secs
Marks	1.00/1.00
Grade	**10.00** out of 10.00 (**100**%)

Question 1 | Correct Mark 1.00 out of 1.00 Flag question

Given an array of N integer, we have to maximize the sum of arr[i] * i, where i is the index of the element (i = 0, 1, 2, ..., N).Write an algorithm based on Greedy technique with a Complexity O(nlogn).

Input Format:

First line specifies the number of elements-n

The next n lines contain the array elements.

Output Format:

Maximum Array Sum to be printed.

Sample Input:

5

2 5 3 4 0

Sample output:

40

Answer: (penalty regime: 0 %)

```c
#include <stdio.h>
#include <stdlib.h>

int compare(const void *a, const void *b) {
    return (*(int*)a - *(int*)b);
}

int main() {
    int n;
    scanf("%d", &n);

    int arr[n];

    for (int i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }

    qsort(arr, n, sizeof(int), compare);

    long long sum = 0;
    for (int i = 0; i < n; i++) {
        sum += (long long)arr[i] * i;
    }

    printf("%lld\n", sum);

    return 0;
}
```

	Input	Expected	Got	
✔	5 2 5 3 4 0	40	40	✔
✔	10 2 2 2 4 4 3 3 5 5 5	191	191	✔

✔	2 45 3	45	45	✔

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.

Finish review

Back to Course

5-G-Product of Array elements-Minimum

Started on	Sunday, 7 September 2025, 11:07 AM
State	Finished
Completed on	Sunday, 7 September 2025, 11:09 AM
Time taken	1 min 10 secs
Marks	1.00/1.00
Grade	**10.00** out of 10.00 (**100**%)

Question 1 | Correct Mark 1.00 out of 1.00 Flag question

Given two arrays array_One[] and array_Two[] of same size N. We need to first rearrange the arrays such that the sum of the product of pairs(1 element from each) is minimum. That is SUM (A[i] * B[i]) for all i is minimum.

For example:

Input	Result
3 1 2 3 4 5 6	28

Answer: (penalty regime: 0 %)

```c
#include <stdio.h>
#include <stdlib.h>

int compare_asc(const void *a, const void *b) {
    return (*(int*)a - *(int*)b);
}

int compare_desc(const void *a, const void *b) {
    return (*(int*)b - *(int*)a);
}

int main() {
    int n;
    scanf("%d", &n);

    int array_One[n];
    int array_Two[n];

    for (int i = 0; i < n; i++) {
        scanf("%d", &array_One[i]);
    }

    for (int i = 0; i < n; i++) {
        scanf("%d", &array_Two[i]);
    }

    qsort(array_One, n, sizeof(int), compare_asc);
    qsort(array_Two, n, sizeof(int), compare_desc);

    long long sum = 0;
    for (int i = 0; i < n; i++) {
        sum += (long long)array_One[i] * array_Two[i];
    }

    printf("%lld\n", sum);

    return 0;
}
```

	Input	Expected	Got	
✔	3 1 2 3 4 5 6	28	28	✔
✔	4 7 5 1	22	22	✔

	2 1 3 4 1			
✔	5 20 10 30 10 40 8 9 4 3 10	590	590	✔

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.

Finish review

Back to Course