Started on	Wednesday, 3 September 2025, 10:34 AM
State	Finished
Completed on	Wednesday, 3 September 2025, 10:35 AM
Time taken	32 secs
Marks	1.00/1.00
Grade	**10.00** out of 10.00 (**100**%)

Question 1 | Correct Mark 1.00 out of 1.00

A person needs to eat burgers. Each burger contains a count of calorie. After eating the burger, the person needs to run a distance to burn out his calories.

If he has eaten i burgers with c calories each, then he has to run at least $3^i * c$ kilometers to burn out the calories. For example, if he ate 3

burgers with the count of calorie in the order: [1, 3, 2], the kilometers he needs to run are $(3^0 * 1) + (3^1 * 3) + (3^2 * 2) = 1 + 9 + 18 = 28$.

But this is not the minimum, so need to try out other orders of consumption and choose the minimum value. Determine the minimum distance

he needs to run. Note: He can eat burger in any order and use an efficient sorting algorithm.Apply greedy approach to solve the problem.

Input Format

First Line contains the number of burgers

Second line contains calories of each burger which is n space-separate integers

Output Format

Print: Minimum number of kilometers needed to run to burn out the calories

Sample Input

```
3
5 10 7
```

Sample Output

```
76
```

For example:

Test	Input	Result
Test Case 1	3 1 3 2	18

Answer: (penalty regime: 0 %)

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

int compare_desc(const void *a, const void *b) {
    int val_a = *(const int *)a;
    int val_b = *(const int *)b;
    if (val_a < val_b) {
        return 1;
    }
    if (val_a > val_b) {
        return -1;
    }
    return 0;

}

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

    qsort(calories, n, sizeof(int), compare_desc);

    long long total_distance = 0;
    long long power_of_n = 1;

    for (int i = 0; i < n; i++) {
```

```
        total_distance += (long long)calories[i] * power_of_n;


        if (i < n - 1) {
            power_of_n *= n;
        }
    }

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

    return 0;
}
```

	Test	Input	Expected	Got	
✔	Test Case 1	3 1 3 2	18	18	✔
✔	Test Case 2	4 7 4 9 6	389	389	✔
✔	Test Case 3	3 5 10 7	76	76	✔

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.

Back to Course