

Started on	Friday, 23 August 2024, 2:52 PM
State	Finished
Completed on	Tuesday, 27 August 2024, 5:08 PM
Time taken	4 days 2 hours
Marks	1.00/1.00
Grade	10.00 out of 10.00 (100%)

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  $O(n \log n)$ .

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 %)

```
2  #include<stdio.h>
3
4  void swap(int* a, int* b)
5  {
6      int temp = *a;
7      *a = *b;
8      *b = temp;
9  }
10
11 int partition(int arr[], int low, int high)
12 {
13     int pivot = arr[low];
14     int i = low;
15     int j = high;
16
17     while (i < j)
18     {
19         while (arr[i] <= pivot && i <= high - 1) i++;
20         while (arr[j] > pivot && j >= low + 1) j--;
21         if (i < j) swap(&arr[i], &arr[j]);
22     }
23     swap(&arr[low], &arr[j]);
24     return j;
25 }
26
27 void quickSort(int arr[], int low, int high)
28 {
29     if (low < high)
30     {
31         int partitionIndex = partition(arr, low, high);
32         quickSort(arr, low, partitionIndex - 1);
33         quickSort(arr, partitionIndex + 1, high);
34     }
35 }
36
37 int main()
38 {
39     int n;
40     scanf("%d", &n);
41     int arr[n];
42     for(int i=0; i<n; ++i)
43     {
44         scanf("%d", &arr[i]);
45     }
46     quickSort(arr, 0, n-1);
47     int sum=0;
48     for(int i=0; i<n; ++i)
49     {
50         sum+=i*arr[i];
51     }
52     printf("%d", sum);
53 }
```

	Input	Expected	Got	
✓	5	40	40	✓
	2			
	5			
	3			
	4			
	0			

	Input	Expected	Got	
✓	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.

[◀ 3-G-Burger Problem](#)

Jump to...

[5-G-Product of Array elements-Minimum ▶](#)