

[Dashboard](#) / [My courses](#) / [CS23331-DAA-2023-AIDS](#) / [Greedy Algorithms](#) / [1-G-Coin Problem](#)

<b>Started on</b>	Thursday, 12 September 2024, 8:12 AM
<b>State</b>	Finished
<b>Completed on</b>	Thursday, 12 September 2024, 8:17 AM
<b>Time taken</b>	5 mins 20 secs
<b>Marks</b>	1.00/1.00
<b>Grade</b>	<b>10.00</b> out of 10.00 ( <b>100%</b> )

Question 1

Correct

Mark 1.00 out of 1.00

Write a program to take value V and we want to make change for V Rs, and we have infinite supply of each of the denominations in Indian currency, i.e., we have infinite supply of { 1, 2, 5, 10, 20, 50, 100, 500, 1000} valued coins/notes, what is the minimum number of coins and/or notes needed to make the change.

Input Format:

Take an integer from stdin.

Output Format:

print the integer which is change of the number.

Example Input :

64

Output:

4

Explanaton:

We need a 50 Rs note and a 10 Rs note and two 2 rupee coins.

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

```
#include <stdio.h>
int main() {
    int d[] = {1000, 500, 100, 50, 20, 10, 5, 2, 1};
    int s = sizeof(d) / sizeof(d[0]);
    int V;
    scanf("%d", &V);
    int c = 0;
    for (int i = 0; i < s; i++) {
        if (V >= d[i]) {
            c += V / d[i];
            V = V % d[i];
        }
    }
    printf("%d", c);
    return 0;
}
```

	Input	Expected	Got	
✓	49	5	5	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

◀ 6-Implementation of Quick Sort

Jump to...

2-G-Cookies Problem ▶

[Dashboard](#) / [My courses](#) / [CS23331-DAA-2023-AIDS](#) / [Greedy Algorithms](#) / [2-G-Cookies Problem](#)

<b>Started on</b>	Thursday, 19 September 2024, 8:10 AM
<b>State</b>	Finished
<b>Completed on</b>	Thursday, 19 September 2024, 8:20 AM
<b>Time taken</b>	10 mins 40 secs
<b>Marks</b>	1.00/1.00
<b>Grade</b>	<b>10.00</b> out of 10.00 ( <b>100%</b> )

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:

$1 \leq g.length \leq 3 \times 10^4$   
 $0 \leq s.length \leq 3 \times 10^4$   
 $1 \leq g[i], s[j] \leq 2^{31} - 1$

Answer: (penalty regime: 0 %)

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

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

int findContentChildren(int g[], int gSize, int s[], int sSize) {

    qsort(g, gSize, sizeof(int), compare);
    qsort(s, sSize, sizeof(int), compare);

    int childIndex = 0;
    int cookieIndex = 0;

    while (childIndex < gSize && cookieIndex < sSize) {
        if (s[cookieIndex] >= g[childIndex]) {
```

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

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

[◀ 1-G-Coin Problem](#)

Jump to...

[3-G-Burger Problem ▶](#)

# CS23331-Design and Analysis of Algorithms-2023 Batch-AIDS

[Dashboard](#) / [My courses](#) / [CS23331-DAA-2023-AIDS](#) / [Greedy Algorithms](#) / [3-G-Burger Problem](#)

## 3-G-Burger Problem

Mark as done

**Opened:** Monday, 9 September 2024, 11:40 AM  
**Closed:** Monday, 28 October 2024, 9:07 AM

Attempts allowed: 10  
Time limit: 1 hour  
Grading method: Highest grade

### Summary of your previous attempts

Attempt	State	Marks / 1.00	Grade / 10.00	Review
1	Finished Submitted Thursday, 19 September 2024, 8:40 AM	0.00	0.00	<a href="#">Review</a>
2	Never submitted	Not yet graded	Not yet graded	<a href="#">Review</a>

Your final grade for this quiz is 0.00/10.00.

Back to the course

[◀ 2-G-Cookies Problem](#)

Jump to...

[4-G-Array Sum max problem ▶](#)



[Dashboard](#) / [My courses](#) / [CS23331-DAA-2023-AIDS](#) / [Greedy Algorithms](#) / [4-G-Array Sum max problem](#)

Started on	Thursday, 19 September 2024, 8:18 AM
State	Finished
Completed on	Thursday, 19 September 2024, 8:51 AM
Time taken	33 mins 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

Given an array of N integer, we have to maximize the sum of  $\text{arr}[i] * i$ , where  $i$  is the index of the element ( $i = 0, 1, 2, \dots, N$ ). 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 %)

```
#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 = (int*)malloc(n * sizeof(int));
    for (int i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }
    qsort(arr, n, sizeof(int), compare);
    int max_sum = 0;
    for (int i = 0; i < n; i++) {
        max_sum += arr[i] * i;
    }
    printf("%d\n", max_sum);
}
```

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

	Input	Expected	Got	
✓	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 ▶

[Dashboard](#) / [My courses](#) / [CS23331-DAA-2023-AIDS](#) / [Greedy Algorithms](#) / [5-G-Product of Array elements-Minimum](#)

Started on	Thursday, 19 September 2024, 8:18 AM
State	Finished
Completed on	Thursday, 19 September 2024, 9:00 AM
Time taken	41 mins 48 secs
Marks	1.00/1.00
Grade	10.00 out of 10.00 (100%)

Question 1

Correct

Mark 1.00 out of 1.00

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 %)

```
#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 = malloc(n * sizeof(int));
    int *array_Two = malloc(n * sizeof(int));

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

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

	Input	Expected	Got	
✓	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.

◀ 4-G-Array Sum max problem

Jump to...

1-DP-Playing with Numbers ▶