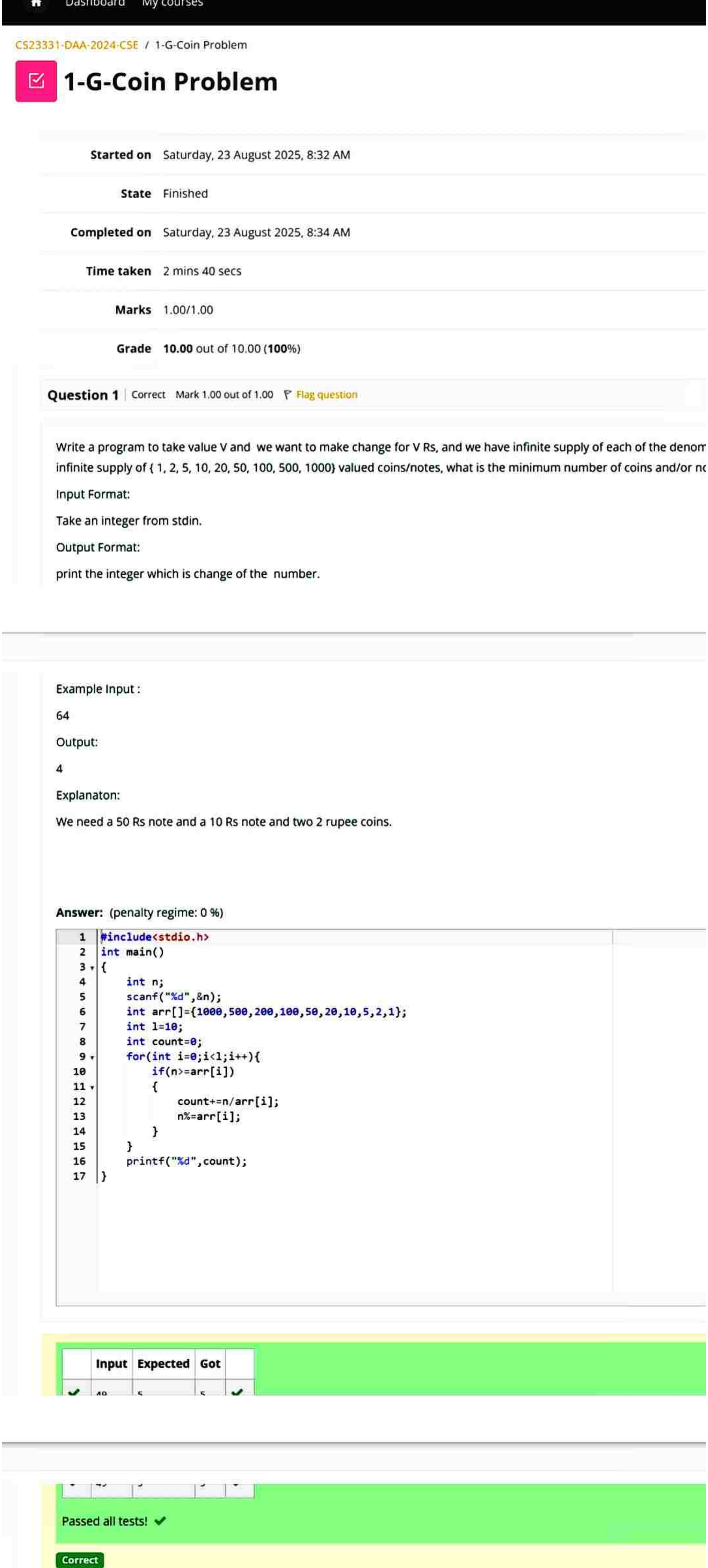


CS23331-DAA-2024-CSE / 1-G-Coin Problem

1-G-Coin Problem

Started on	Saturday, 23 August 2025, 8:32 AM
State	Finished
Completed on	Saturday, 23 August 2025, 8:34 AM
Time taken	2 mins 40 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

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 denom
infinite supply of { 1, 2, 5, 10, 20, 50, 100, 500, 1000} valued coins/notes, what is the minimum number of coins and/or nc

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.

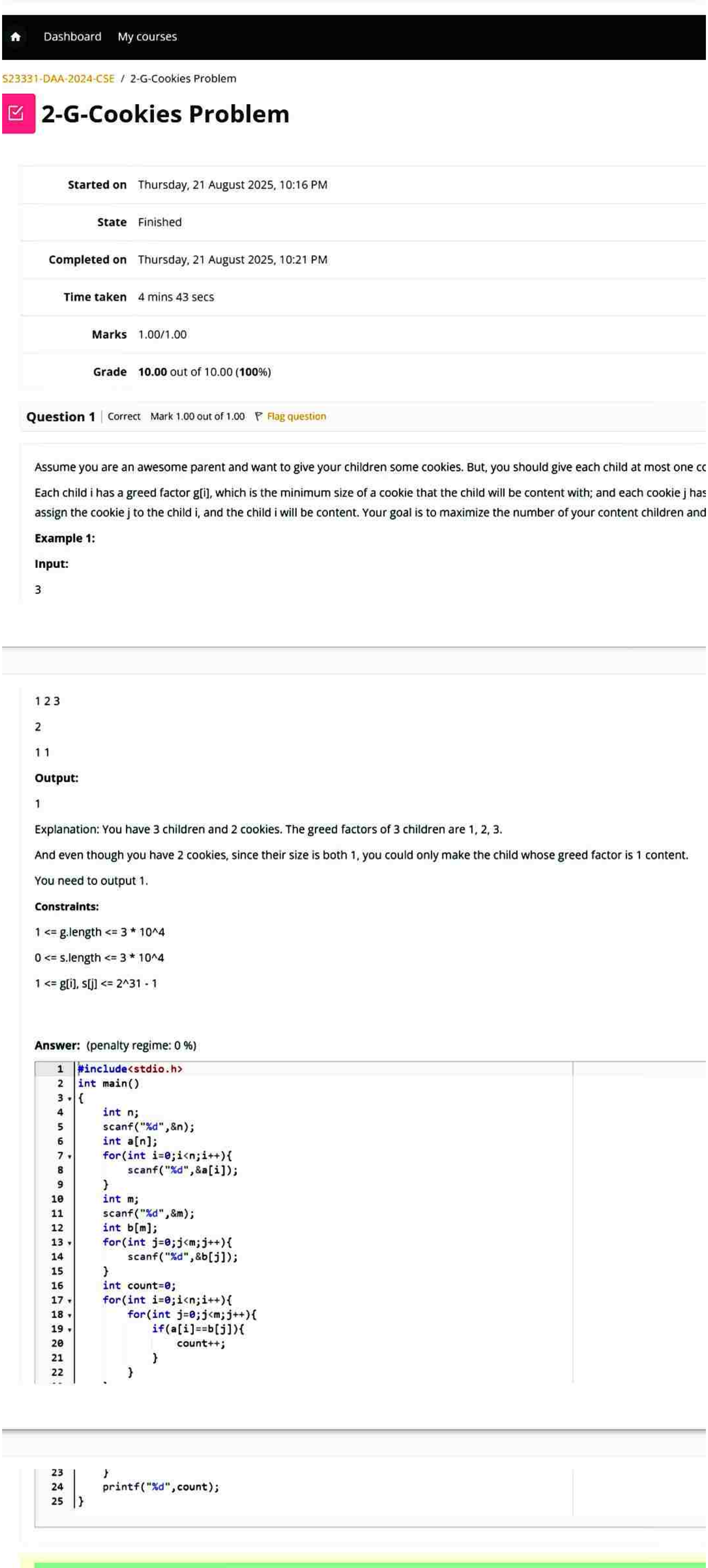
Answer: (penalty regime: 0 %)

```
#include<stdio.h>
int main()
{
    int n;
    scanf("%d",&n);
    int arr[]={1000,500,200,100,50,20,10,5,2,1};
    int l=10;
    int count=0;
    for(int i=0;i<l;i++){
        if(n>=arr[i])
        {
            count+=n/arr[i];
            n%=arr[i];
        }
    }
    printf("%d",count);
}
```

	Input	Expected	Got	
✓	49	5	5	✓

Passed all tests! ✓

Correct



S23331-DAA-2024-CSE / 2-G-Cookies Problem

2-G-Cookies Problem

Started on	Thursday, 21 August 2025, 10:16 PM
State	Finished
Completed on	Thursday, 21 August 2025, 10:21 PM
Time taken	4 mins 43 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 cc

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 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

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:

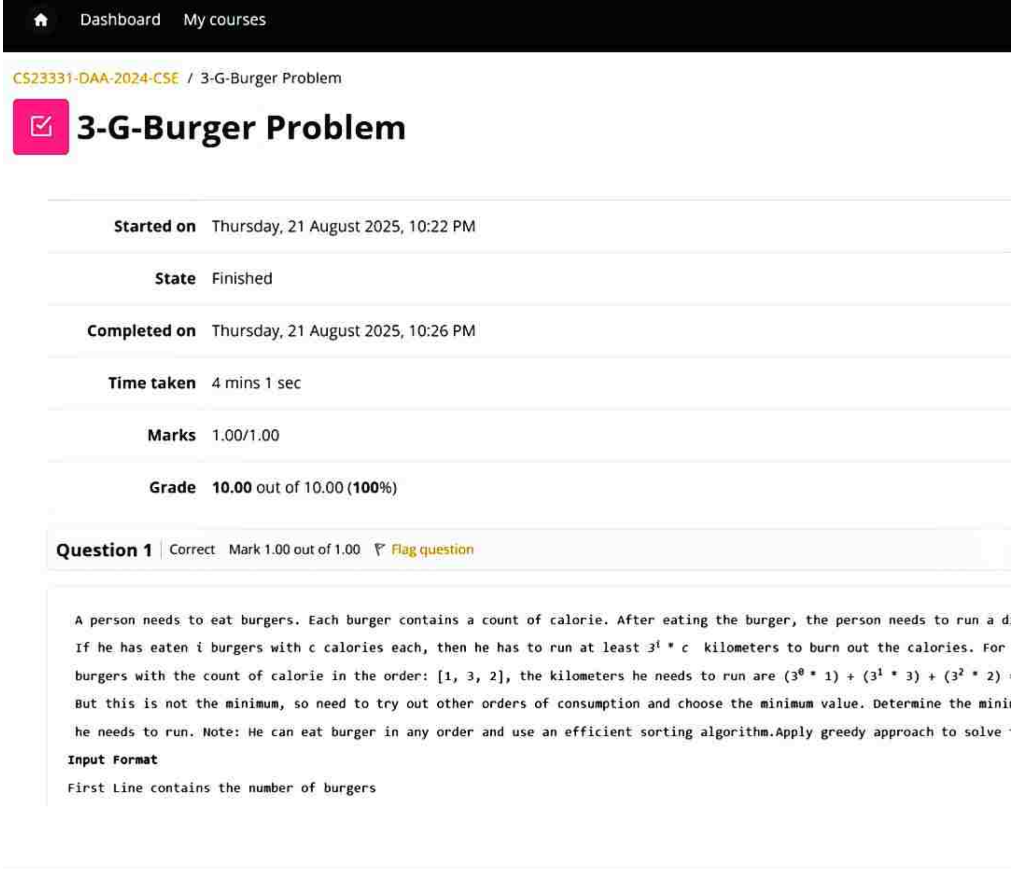
1 <= g.length <= 3 * 10^4

0 <= s.length <= 3 * 10^4

1 <= g[i], s[j] <= 2^31 - 1

Answer: (penalty regime: 0 %)

```
#include<stdio.h>
int main()
{
    int n;
    scanf("%d",&n);
    int a[n];
    for(int i=0;i<n;i++){
        scanf("%d",&a[i]);
    }
    int m;
    scanf("%d",&m);
    int b[m];
    for(int j=0;j<m;j++){
        scanf("%d",&b[j]);
    }
    int count=0;
    for(int i=0;i<n;i++){
        for(int j=0;j<m;j++){
            if(a[i]==b[j]){
                count++;
            }
        }
    }
    printf("%d",count);
}
```
)



CS23331-DAA-2024-CSE / 3-G-Burger Problem

3-G-Burger Problem

Started on	Thursday, 21 August 2025, 10:22 PM
State	Finished
Completed on	Thursday, 21 August 2025, 10:26 PM
Time taken	4 mins 1 sec
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

A person needs to eat burgers. Each burger contains a count of calorie. After eating the burger, the person needs to run a d

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

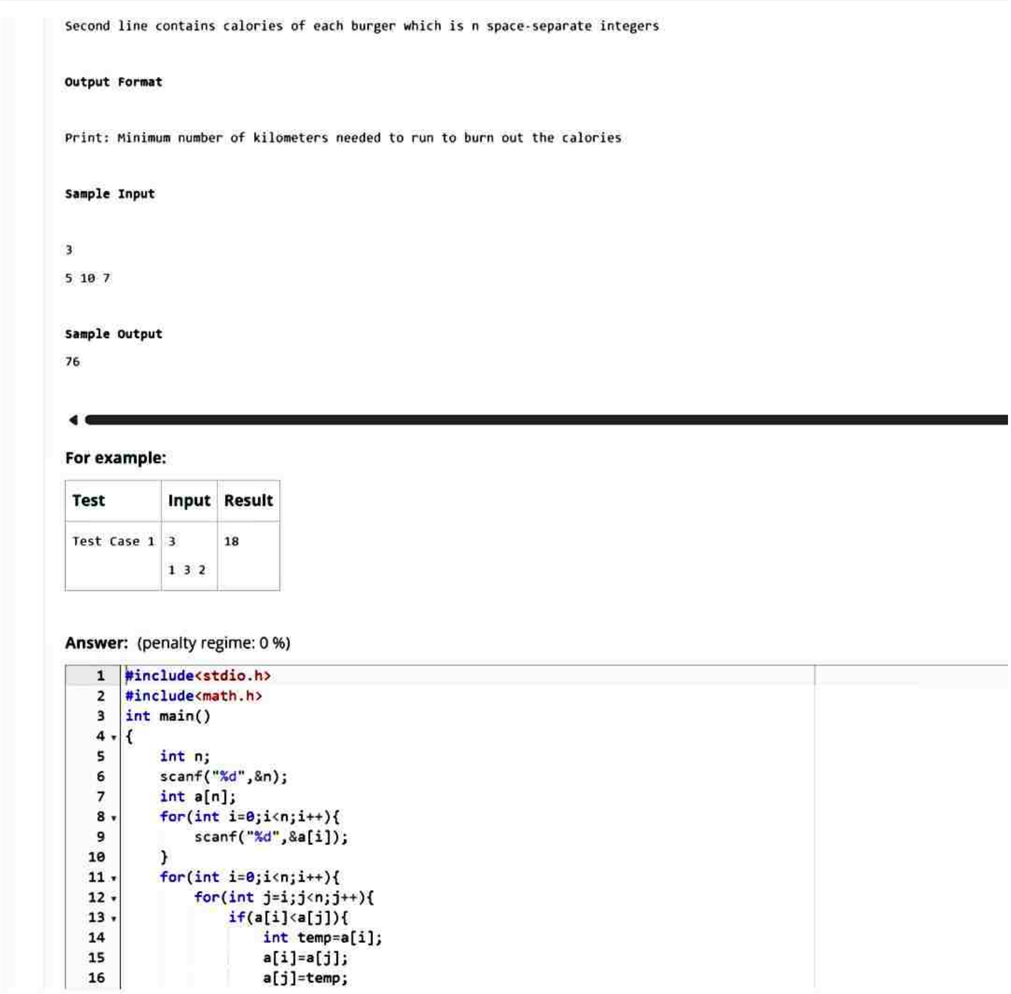
burgers with the count of calorie in the order: [1, 3, 2], the kilometers he needs to run are $(3^0 * 1) + (3^1 * 3) + (3^2 * 2)$

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

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

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

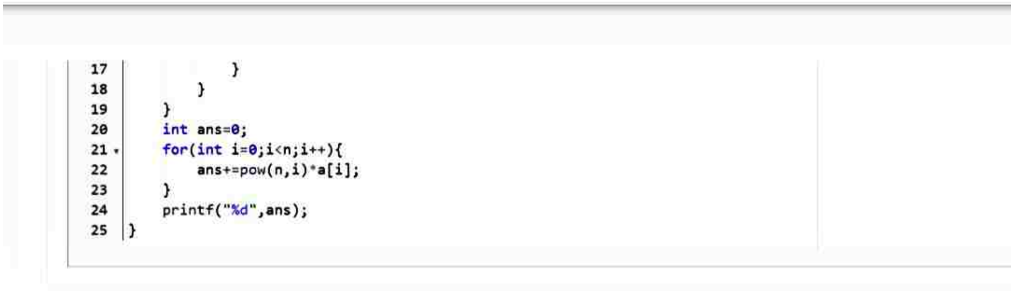
76

For example:

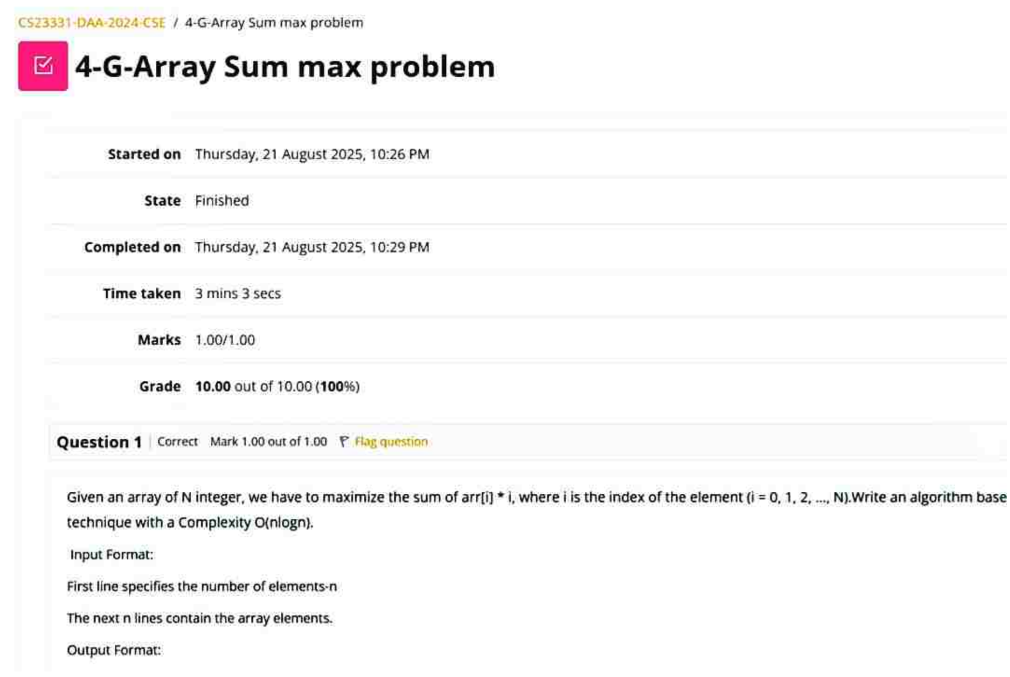
Test	Input	Result
Test Case 1	3 1 3 2	18

Answer: (penalty regime: 0 %)

```
#include<stdio.h>
#include<math.h>
int main()
{
    int n;
    scanf("%d",&n);
    int a[n];
    for(int i=0;i<n;i++){
        scanf("%d",&a[i]);
    }
    for(int i=0;i<n;i++){
        for(int j=i;j<n;j++){
            if(a[i]<a[j]){
                int temp=a[i];
                a[i]=a[j];
                a[j]=temp;
```



```
            }
        }
    }
    int ans=0;
    for(int i=0;i<n;i++){
        ans+=pow(n,i)*a[i];
    }
    printf("%d",ans);
}
```



CS23331-DAA-2024-CSE / 4-G-Array Sum max problem

4-G-Array Sum max problem

Started on	Thursday, 21 August 2025, 10:26 PM
State	Finished
Completed on	Thursday, 21 August 2025, 10:29 PM
Time taken	3 mins 3 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

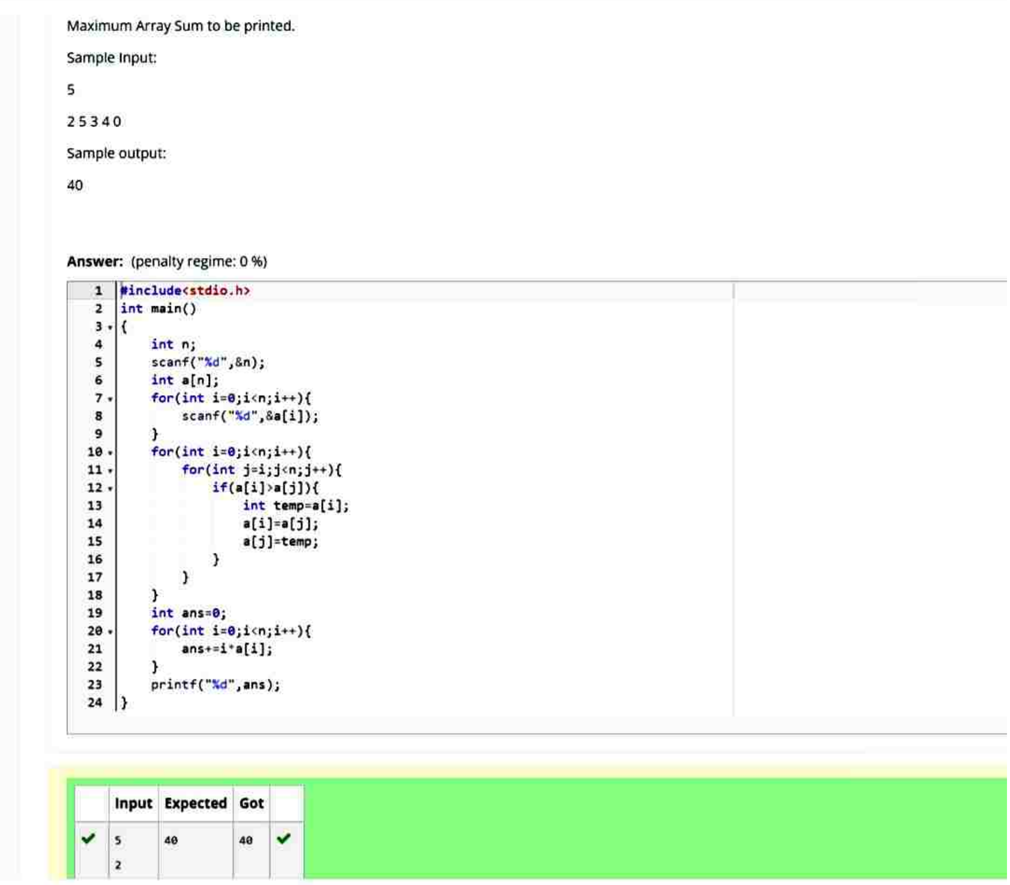
Given an array of N integer, we have to maximize the sum of arr[i] * i, where i is the index of the element (i = 0, 1, 2, ..., N).Write an algorithm base technique with a Complexity O(nlogn).

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

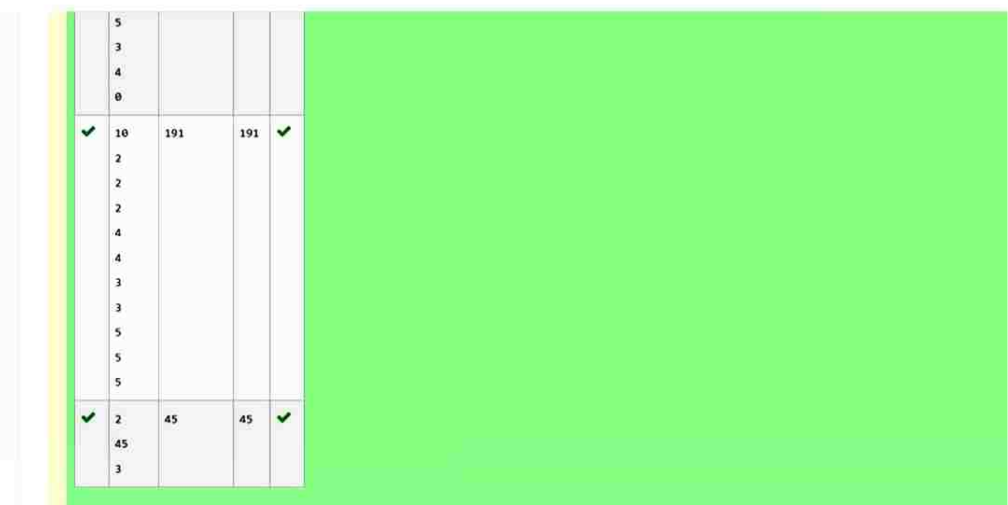
2 5 3 4 0

Sample output:

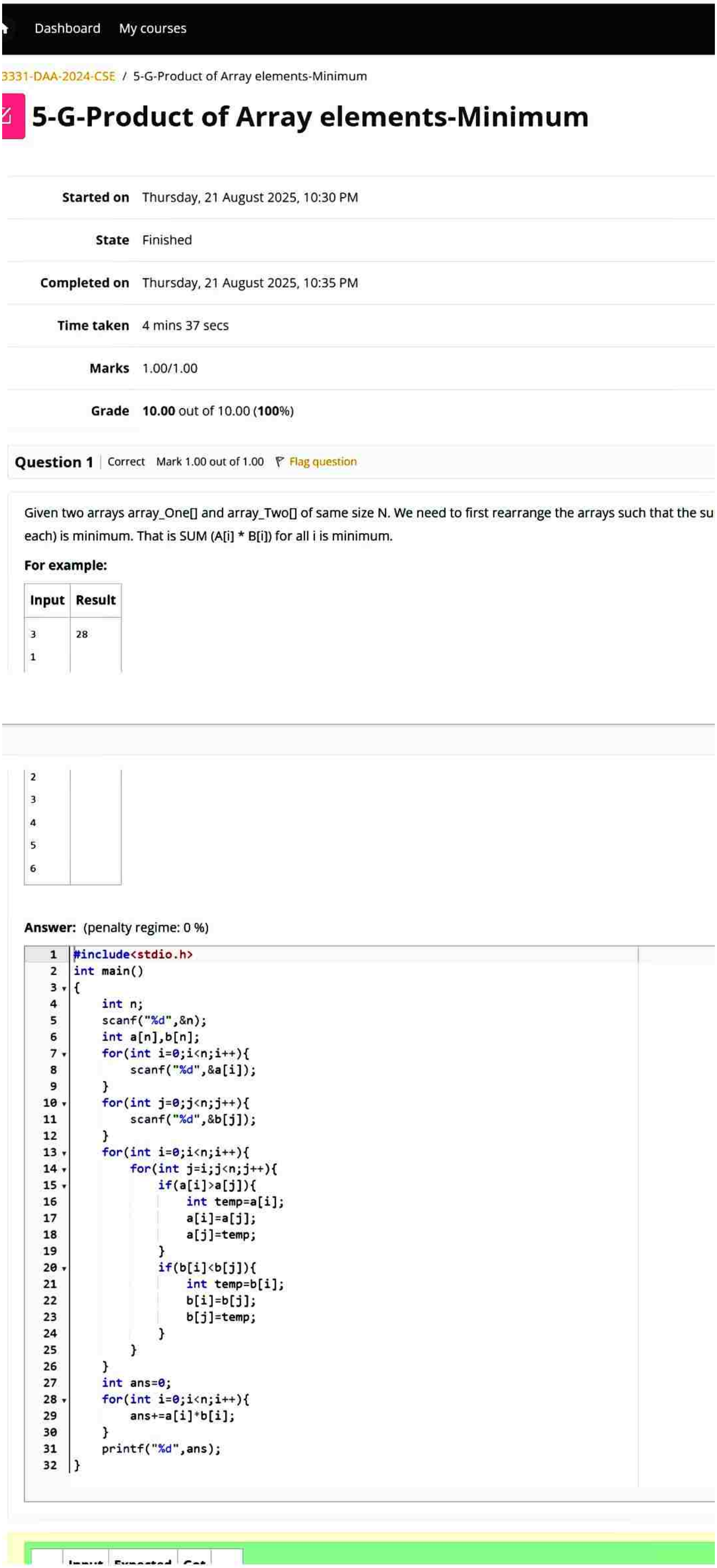
40

Answer: (penalty regime: 0 %)

```c
#include<stdio.h>
int main()
{
    int n;
    scanf("%d",&n);
    int a[n];
    for(int i=0;i<n;i++){
        scanf("%d",&a[i]);
    }
    for(int i=0;i<n;i++){
        for(int j=i;j<n;j++){
            if(a[i]>a[j]){
                int temp=a[i];
                a[i]=a[j];
                a[j]=temp;
            }
        }
    }
    int ans=0;
    for(int i=0;i<n;i++){
        ans+=i*a[i];
    }
    printf("%d",ans);
}
```



	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	✓



Dashboard My courses

3331-DAA-2024-CSE / 5-G-Product of Array elements-Minimum

5-G-Product of Array elements-Minimum

Started on	Thursday, 21 August 2025, 10:30 PM
State	Finished
Completed on	Thursday, 21 August 2025, 10:35 PM
Time taken	4 mins 37 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 su 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>
int main()
{
    int n;
    scanf("%d",&n);
    int a[n],b[n];
    for(int i=0;i<n;i++){
        scanf("%d",&a[i]);
    }
    for(int j=0;j<n;j++){
        scanf("%d",&b[j]);
    }
    for(int i=0;i<n;i++){
        for(int j=i;j<n;j++){
            if(a[i]>a[j]){
                int temp=a[i];
                a[i]=a[j];
                a[j]=temp;
            }
            if(b[i]<b[j]){
                int temp=b[i];
                b[i]=b[j];
                b[j]=temp;
            }
        }
    }
    int ans=0;
    for(int i=0;i<n;i++){
        ans+=a[i]*b[i];
    }
    printf("%d",ans);
}
```

	Input	Expected	Got	