

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 %)

```
1 #include<stdio.h>
2 int main(){
3     int arr[10]={1000,500,100,50,20,10,5,2,1};
4     int n,res;
5     int count=-1;
6     scanf("%d",&n);
7     for(int i=0;i<10;i++){
8         if(arr[i]<=n){
9             res=n-arr[i];
10            count++;
11            n=res;
12        }
13    }
14    printf("%d",count);
15 }
16
17 }
```

	Input	Expected	Got	
✓	49	5	5	✓

Passed all tests! ✓

**Correct**

Marks for this submission: 1.00/1.00.

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 %)

```
1 #include<stdio.h>
2 int main(){
3     int n,m;
4     scanf("%d%d",&n,&m);
5     int arr1[n];
6     int arr2[m];
7     for(int i=0;i<n;i++){
8         scanf("%d",&arr1[i]);
9     }
10    for(int j=0;j<m;j++){
11        scanf("%d",&arr2[j]);
12    }
13    int count=0;
14    for(int i=0;i<n;i++){
15        for(int j=0;j<m;j++){
16            if(arr2[i]<=arr1[j]){
17                count++;
18            }
19        }
20    }
21    printf("%d",count);
22 }
23 }
```

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

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

[Back to Course](#)

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 %)

```
1 #include<stdio.h>
2 #include<math.h>
3 int main(){
4     int n;
5     scanf("%d",&n);
6     int arr[n];
7     for(int i=0;i<n;i++){
8         scanf("%d",&arr[i]);
9     }
10    for(int i=0;i<n-1;i++){
11        for(int j=0;j<n-i-1;j++){
12            if(arr[j]>arr[j+1]){
13                int temp=arr[j];
14                arr[j]=arr[j+1];
15                arr[j+1]=temp;
16            }
17        }
18    }
19    int sum=0;
20    for(int i=0;i<n;i++){
21        sum+=pow(n,i)*arr[i];
22    }
23    printf("%d",sum);
24 }
```

	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](#)

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 %)

```
1 #include<stdio.h>
2 int main(){
3     int n;
4     scanf("%d",&n);
5     int arr[n];
6     for(int i=0;i<n;i++){
7         scanf("%d",&arr[i]);
8     }
9     int key,j;
10    for(int i=1;i<n;i++){
11        j=i-1;
12        key=arr[i];
13        while(j>=0 && key<=arr[j]){
14            arr[j+1]=arr[j];
15            j--;
16        }arr[j+1]=key;
17    }
18    // for(int i=0;i<n;i++){
19    //     printf("%d ",arr[i]);
20    // }
21    int sum=0;
22    for(int i=0;i<n;i++){
23        sum=sum+arr[i]*i;
24    }
25    printf("%d\n",sum);
26 }
27 }
```

	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.

[Back to Course](#)

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  $\text{SUM}(A[i] * B[i])$  for all `i` is minimum.

**For example:**

Input	Result
3	28
1	
2	
3	
4	
5	
6	

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

```

1  #include<stdio.h>
2  int main(){
3      int n;
4      scanf("%d",&n);
5      int arr1[n];
6      int arr2[n];
7      for(int i=0;i<n;i++){
8          scanf("%d",&arr1[i]);
9      }
10     for(int i=0;i<n;i++){
11         scanf("%d",&arr2[i]);
12     }
13     int key,j;
14     for(int i=1;i<n;i++){
15         j=i-1;
16         key=arr1[i];
17         while(j>=0 && key<arr1[j]){
18             arr1[j+1]=arr1[j];
19             j--;
20         }arr1[j+1]=key;
21     }
22     for(int i=1;i<n;i++){
23         j=i-1;
24         key=arr2[i];
25         while(j>=0 && key>arr2[j]){
26             arr2[j+1]=arr2[j];
27             j--;
28         }arr2[j+1]=key;
29     }
30     int sum=0;
31     for(int i=0;i<n;i++){
32         sum=sum+(arr1[i]*arr2[i]);
33     }
34     printf("%d",sum);
35
36 }
```



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

[Back to Course](#)