ROLL NO:230701036

NAME: ARUN MC

TOPIC: GREEDY ALGORITHM

1-G-Coin Problem

AIM:

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.

CODE:

```
#include<stdio.h>

#include<math.h>

int main()

{

  int a,b[]={1,2,5,10,20,50,100,500,1000},c=0;

  scanf("%d",&a);

  for(int i=8;a!=0;i--)

  {

    if (b[i]<a )

    {

      c++;

      a=a%b[i];

      i=8;

    }

  }
```

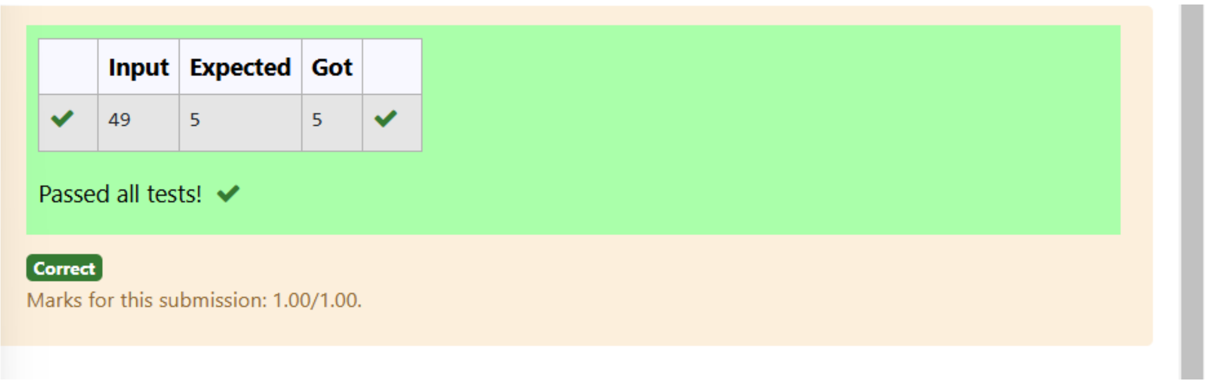
```
    printf("%d",c+2);
}
```

INPUT:

Take an integer from stdin.

64

OUTPUT:



	Input	Expected	Got	
✓	49	5	5	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

2-G-Cookies Problem

AIM:

To maximize the number of children who are content by assigning cookies, where each child gets at most one cookie, and the cookie size must meet or exceed the child's greed factor.

CODE:

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

INPUT:

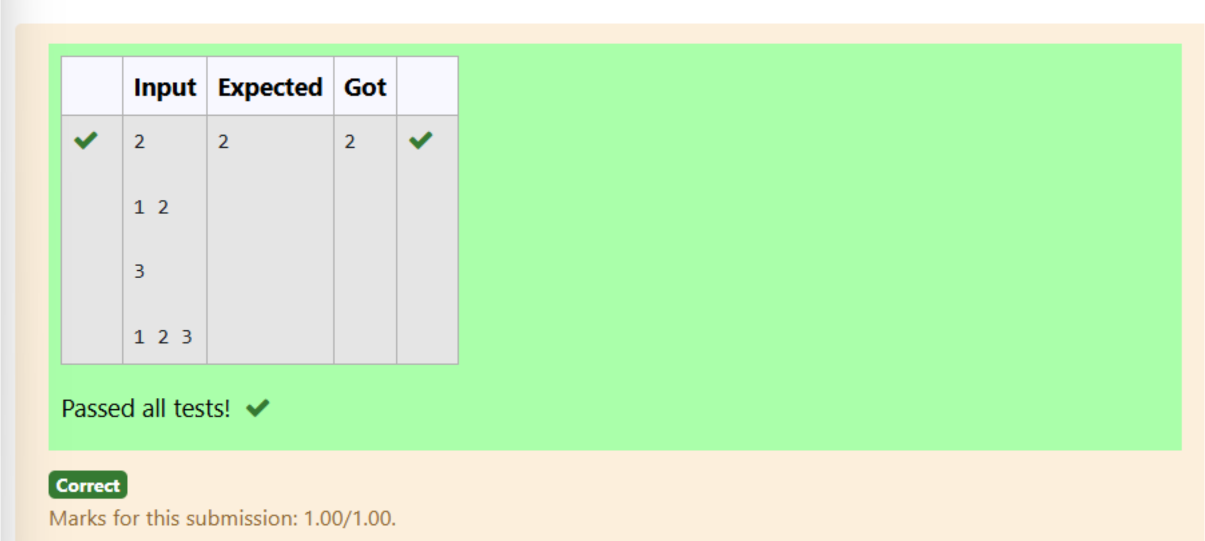
```
3
```

1 2 3

2

1 1

OUTPUT:



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

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

3-G-Burger Problem

AIM:

To determine the minimum distance a person needs to run to burn calories after eating burgers, using a greedy approach to find the optimal order of burger consumption that minimizes the total running distance.

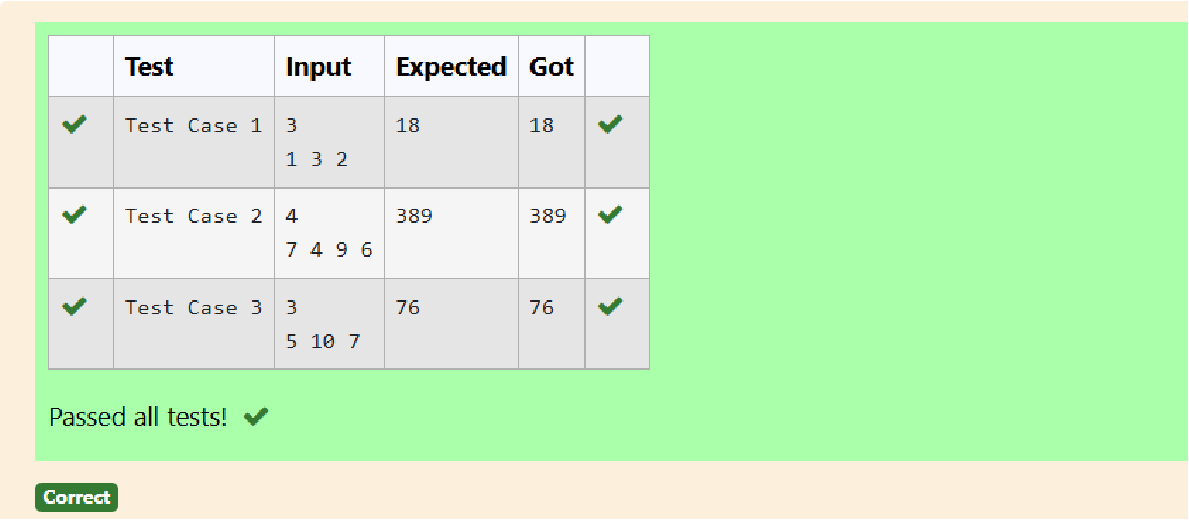
CODE:

```
#include<stdio.h>
#include<math.h>
int main(){
  int n;
  scanf("%d",&n);
  int a[n],cnt=0;
  for(int i=0;i<n;i++)
   scanf("%d",&a[i]);
  for(int i=0;i<n;i++)
   for(int j=0;j<n-1;j++)
    if(a[j]<a[j+1]){
      int t=a[j];
      a[j]=a[j+1];
      a[j+1]=t;
    }
  for(int i=0;i<n;i++)
    cnt+=pow(n,i)*a[i];
  printf("%d",cnt) ;
}
```

INPUT:

3
5 10 7

OUTPUT:



	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! ✓

4-G-Array Sum max problem

AIM:

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(nlogn).

INPUT:

First line specifies the number of elements-n

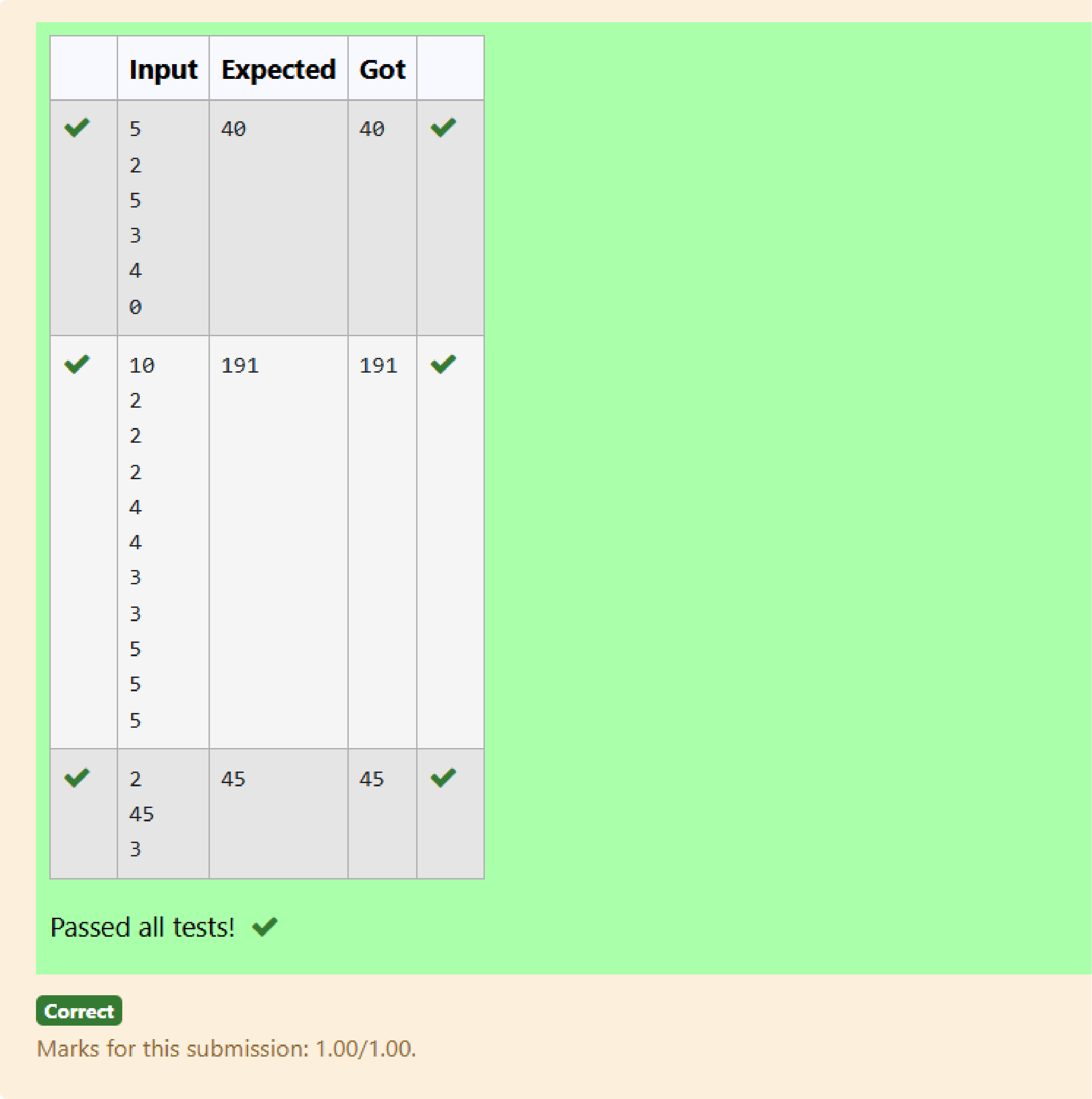
The next n lines contain the array elements.

CODE:

```
#include<stdio.h>
int main(){
   int n;
   scanf("%d",&n);
   int a[n],cnt=0;
   for(int i=0;i<n;i++)
     scanf("%d",&a[i]);
   for(int i=0;i<n;i++)
     for(int j=0;j<n-1;j++)
       if(a[j]>a[j+1])
       {
          int t=a[j];
          a[j]=a[j+1];
          a[j+1]=t;
       }
   for(int i=0;i<n;i++)
     cnt+=a[i]*i;
```

```
  printf("%d ",cnt)  ;
}
```

OUTPUT:



	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	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

5-G-Product of Array elements-Minimum

AIM:

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.

INPUT:

Input	Result
3 1 2 3 4 5 6	28

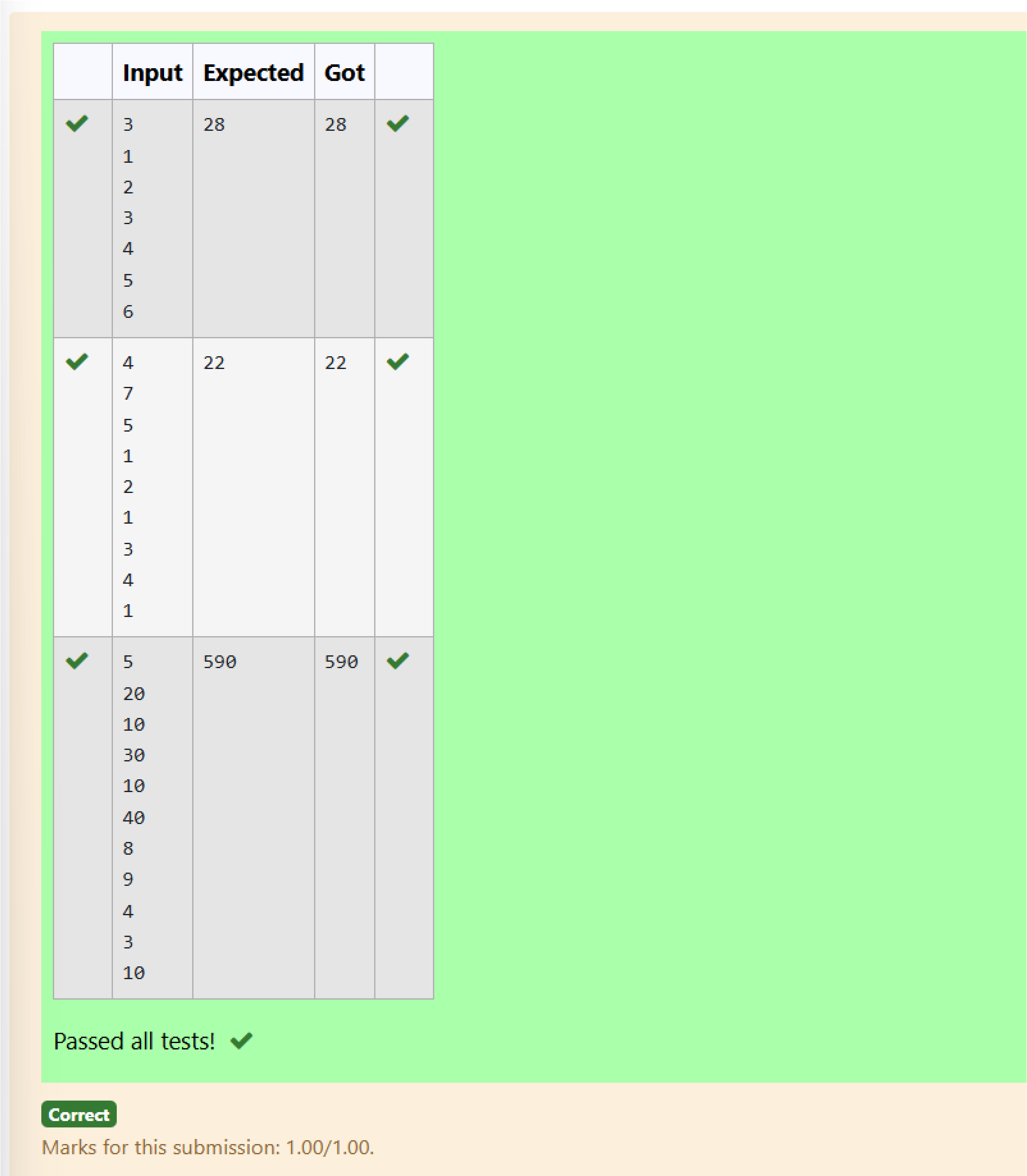
CODE:

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

```
   for(int i=0;i<n;i++)
     for(int j=0;j<n-1;j++){
        if (a[j]<a[j+1]){
           int t=a[j];
           a[j]=a[j+1];
           a[j+1]=t;
        }
        if(b[j]>b[j+1]){
           int t=b[j];
           b[j]=b[j+1];
           b[j+1]=t;
        }
     }
   for(int i=0;i<n;i++)
     cnt+=a[i]*b[i];
   printf("%d",cnt)  ;

}
```

OUTPUT:



	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.