[**GREEDY**](http://118.185.187.137/moodle/course/view.php?id=158#section-1) **ALGORITHMS**

**AIM:**

To implement a C program that calculates the minimum number of Indian currency coins and/or notes needed to make change for a given amount V using a greedy approach.

**PROGRAM:**

#include<stdio.h>

int main()

{

int n,q1,q2,q3,q4,q5,q6,q7,q8,c=0;

scanf("%d",&n);

while(n>0)

{

if(n>=1000)

{

//r1=n%1000;

q1=n/1000;

c=c+q1;

n=n-(1000\*q1);

}

else if(n>=500)

{

//r2=n%500;

q2=n/500;

c=c+q2;

n=n-(500\*q2);

}

else if(n>=100)

{

// r3=n%100;

q3=n/100;

c=c+q3;

n=n-(100\*q3);

}

else if(n>=50)

{

//r4=n%50;

q4=n/50;

c=c+q4;

n=n-(50\*q4);

}

else if(n>=20)

{

//r5=n%20;

q5=n/20;

c=c+q5;

n=n-(20\*q5);

}

else if(n>=10)

{

//r6=n%10;

q6=n/10;

c=c+q6;

n=n-(10\*q6);

}

else if(n>=5)

{

q8=n/5;

c=c+q8;

n=n-(5\*q8);

}

else if(n>=2)

{

//r7=n%2;

q7=n/2;

c=c+q7;

n=n-(2\*q7);

}

else if(n==1)

{

c=c+1;

n=n-1;

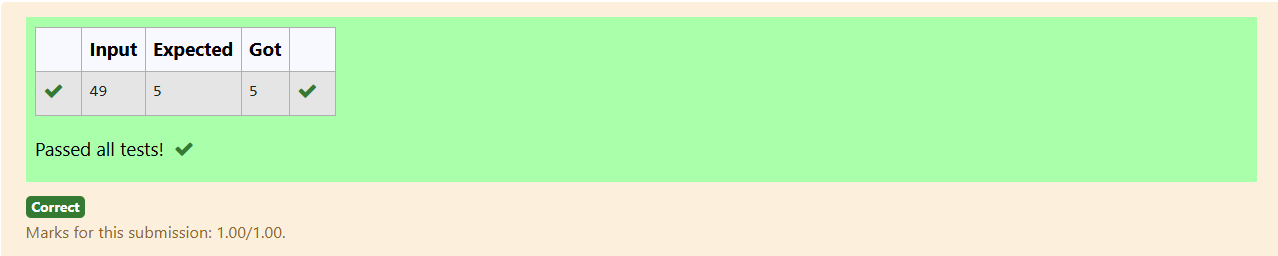
}

}

printf("%d",c);

}

**OUTPUT:**



**AIM:**

To implement a C program that maximizes the number of content children by distributing cookies, where each child receives at most one cookie that meets their minimum size requirement.

**PROGRAM:**

#include<stdio.h>

int main()

{

int n;

scanf("%d",&n);

int arr[n];

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

{

scanf("%d",&arr[i]);

}

int arr1[n];

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

{

int count=0;

for(int j=0;j<n;j++)

{

if(arr[i]==arr[j])

count++;

}

arr1[i]=count;

}

int max=arr1[0];

for(int i=1;i<n;i++)

{

if(arr1[i]>max)

max=arr1[i];

}

int indx=0;

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

{

if(arr1[i]==max)

{

indx=i;

break;

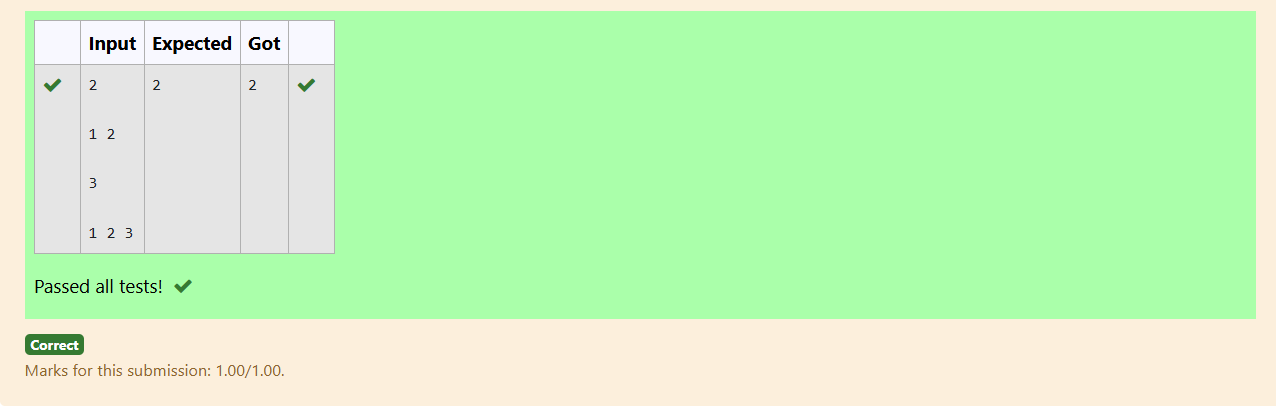
}

}

printf("%d",arr[indx]);

}

**OUTPUT:**



**AIM:**

To implement a C program that calculates the minimum distance a person needs to run to burn calories after eating burgers, using a greedy approach to determine the optimal order of consumption.

**PROGRAM:**

#include<stdio.h>

#include<math.h>

void bubbleSort(int arr[], int n) {

for (int i = 0; i < n - 1; i++) {

for (int j = 0; j < n - i - 1; j++) {

if (arr[j] > arr[j + 1]) {

// Swap arr[j] and arr[j + 1]

int temp = arr[j];

arr[j] = arr[j + 1];

arr[j + 1] = temp;

}

}

}

}

int main()

{

int n;

scanf("%d",&n);

int arr[n];

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

{

scanf("%d",&arr[i]);

}

bubbleSort(arr,n);

int i=0;

int res=0;

for(int j=n-1;j>=0;j--)

{

res=res+((int)pow(n,i)\*arr[j]);

i=i+1;

}

printf("%d ",res);

}

**OUTPUT:**



**AIM:**

To implement a C program that maximizes the sum of arr[i] \* i for a given array, where i is the index, by using a greedy approach with a time complexity of O(nlogn).

**PROGRAM:**

#include<stdio.h>

void bubbleSort(int arr[], int n) {

for (int i = 0; i < n - 1; i++) {

for (int j = 0; j < n - i - 1; j++) {

if (arr[j] > arr[j + 1]) {

int temp = arr[j];

arr[j] = arr[j + 1];

arr[j + 1] = temp;

}

}

}

}

int main()

{

int n;

scanf("%d",&n);

int arr[n];

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

{

scanf("%d",&arr[i]);

}

bubbleSort(arr,n);

int j=n-1;

int ms=0;

for(int i=n-1;i>=0;i--)

{

ms=ms+arr[i]\*j;

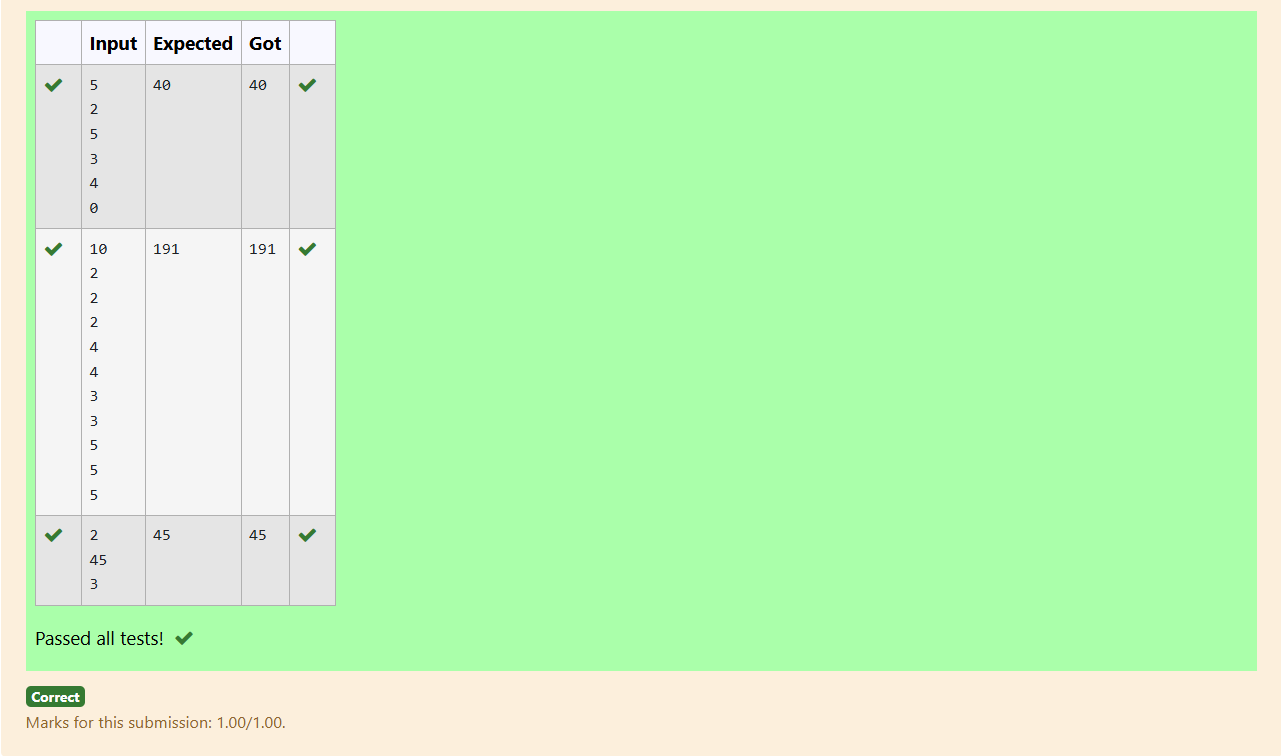
j=j-1;

}

printf("%d",ms);

}

**OUTPUT:**



**AIM:**

To implement a C program that rearranges two arrays of size N to minimize the sum of the products of paired elements, using a greedy approach.

**PROGRAM:**

#include<stdio.h>

void bubbleSort(int arr[], int n) {

for (int i = 0; i < n - 1; i++) {

for (int j = 0; j < n - i - 1; j++) {

if (arr[j] > arr[j + 1]) {

// Swap arr[j] and arr[j + 1]

int temp = arr[j];

arr[j] = arr[j + 1];

arr[j + 1] = temp;

}

}

}

}

int main()

{

int n;

scanf("%d",&n);

int arr1[n];

int arr2[n];

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

{

scanf("%d",&arr1[i]);

}

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

{

scanf("%d",&arr2[i]);

}

bubbleSort(arr1,n);

bubbleSort(arr2,n);

int res=0;

int k=0;

for(int i=n-1;i>=0;i--)

{

res=res+(arr1[i]\*arr2[k]);

k=k+1;

}

printf("%d",res);

}

**OUTPUT:**

