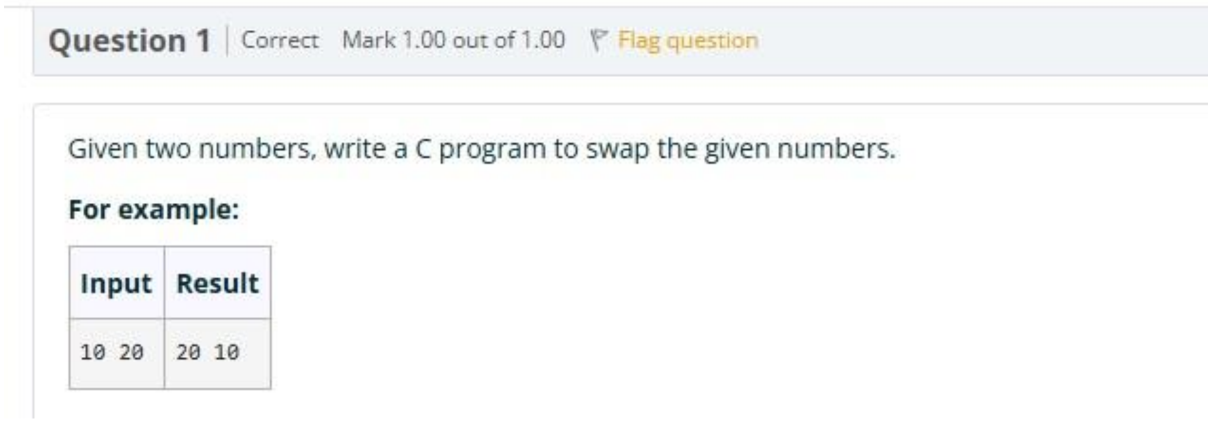
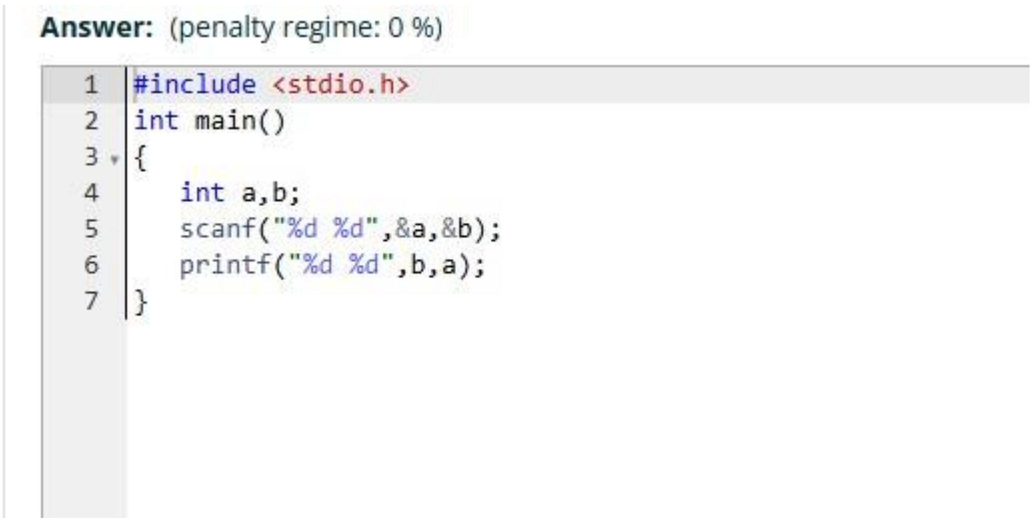
1.Basic C-Programming


Question 1 | Correct Mark 1.00 out of 1.00 Flag question

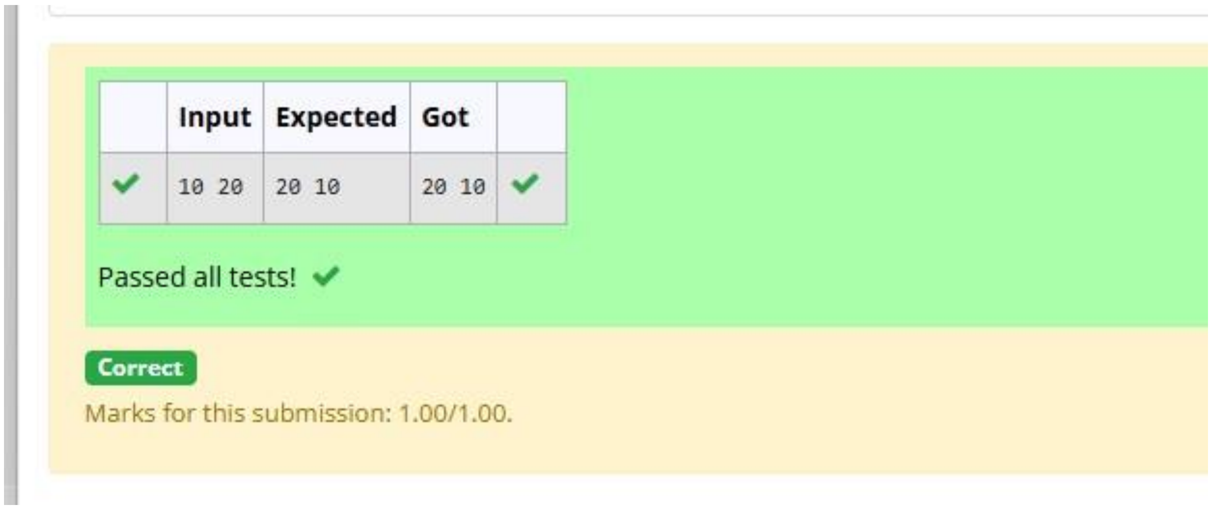
Given two numbers, write a C program to swap the given numbers.

For example:

Input	Result
10 20	20 10


Answer: (penalty regime: 0 %)

```c
#include <stdio.h>
int main()
{
    int a,b;
    scanf("%d %d",&a,&b);
    printf("%d %d",b,a);
}
```



	Input	Expected	Got	
✓	10 20	20 10	20 10	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 2 Correct Mark 1.00 out of 1.00 Flag question

Write a C program to find the eligibility of admission for a professional course based on the following criteria:

Marks in Maths >= 65

Marks in Physics >= 55

Marks in Chemistry >= 50

Or

Total in all three subjects >= 180

Answer: (penalty regime: 0 %)

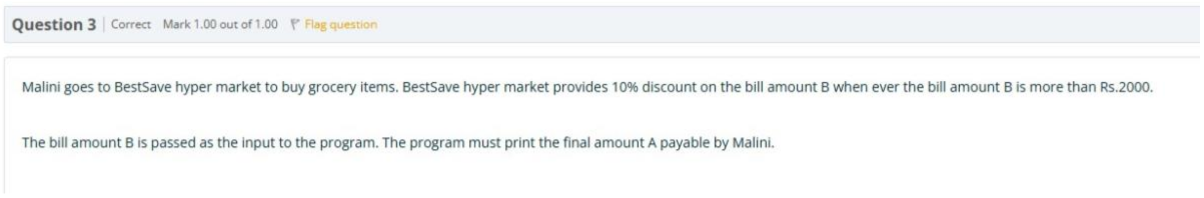
```c
#include <stdio.h>
int main()
{
    int a,b,c,d;
    scanf("%d %d %d",&a,&b,&c);
    d=a+b+c;
    if(d >= 180)
    {
        printf("The candidate is eligible");
    }
    else
    {
        printf("The candidate is not eligible");
    }
    return 0;
}

```

	Input	Expected	Got	
✓	70 60 80	The candidate is eligible	The candidate is eligible	✓
✓	50 80 80	The candidate is eligible	The candidate is eligible	✓

Passed all tests! ✓

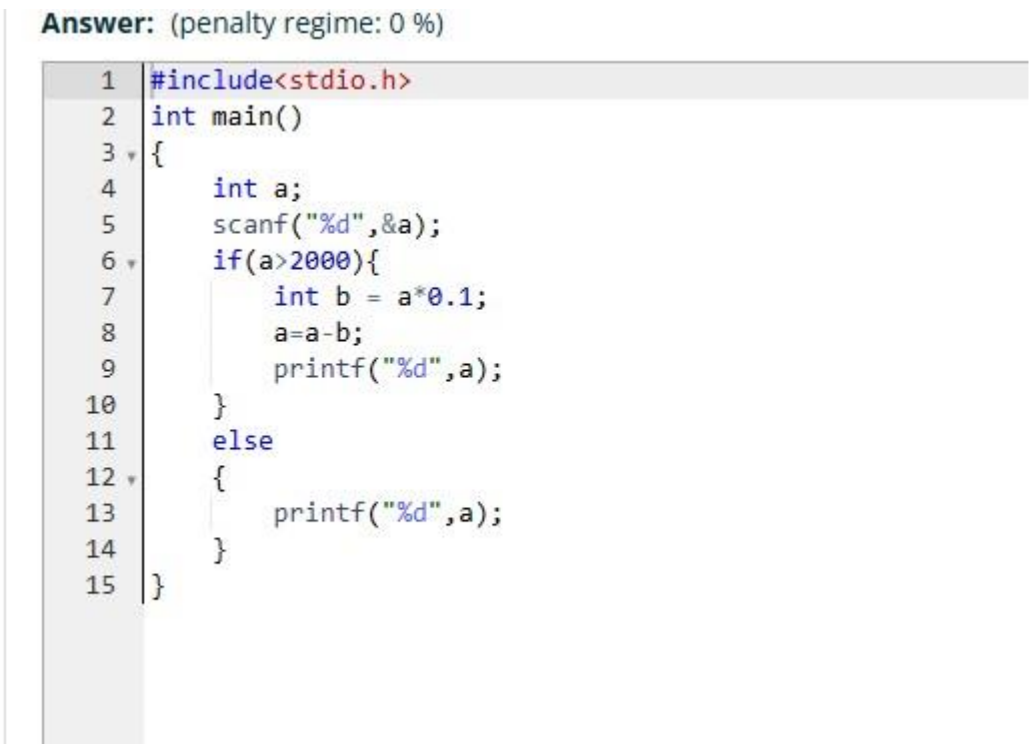
Correct

Marks for this submission: 1.00/1.00.

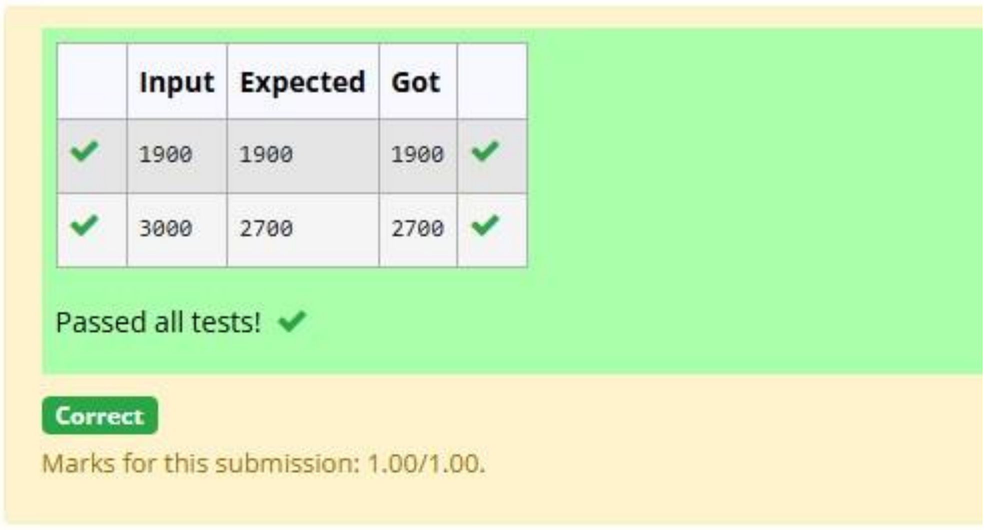

Question 3 | Correct Mark 1.00 out of 1.00 Flag question

Malini goes to BestSave hyper market to buy grocery items. BestSave hyper market provides 10% discount on the bill amount B when ever the bill amount B is more than Rs.2000.

The bill amount B is passed as the input to the program. The program must print the final amount A payable by Malini.


Answer: (penalty regime: 0 %)

```c
#include<stdio.h>
int main()
{
    int a;
    scanf("%d",&a);
    if(a>2000){
        int b = a*0.1;
        a=a-b;
        printf("%d",a);
    }
    else
    {
        printf("%d",a);
    }
}
```

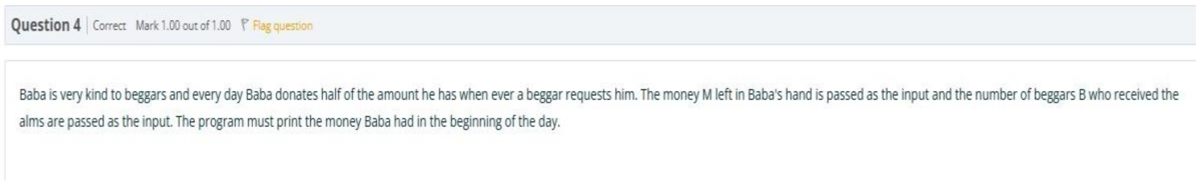


	Input	Expected	Got	
✓	1900	1900	1900	✓
✓	3000	2700	2700	✓

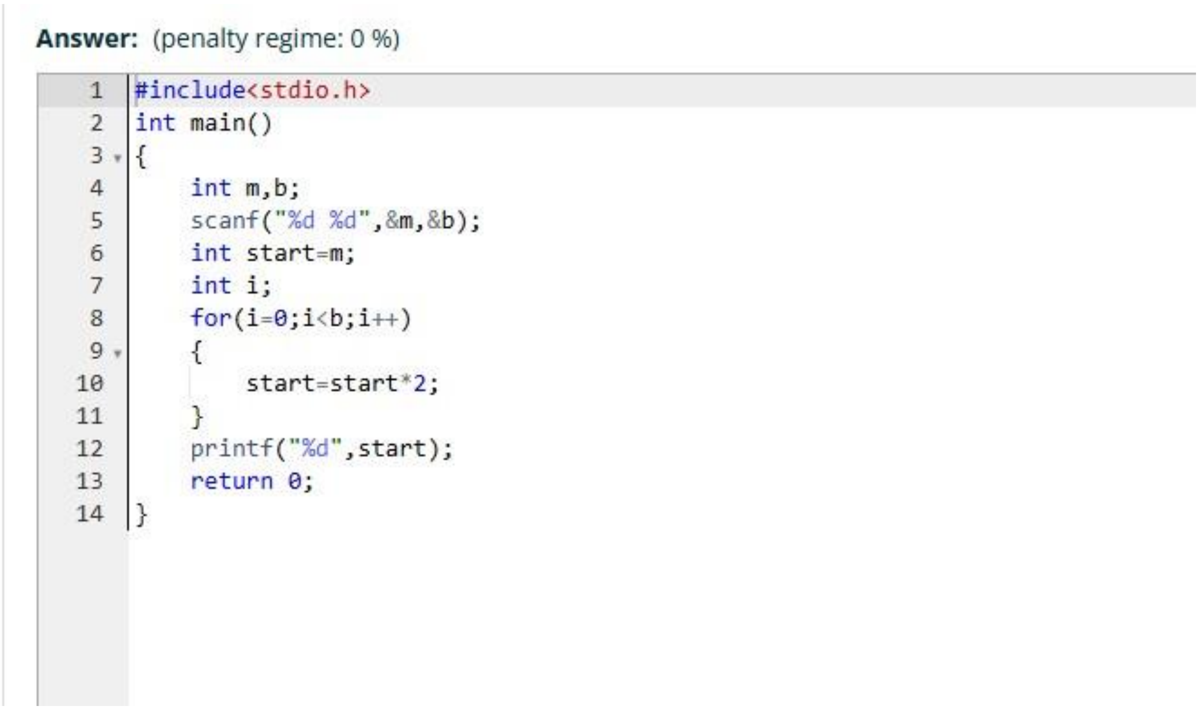
Passed all tests! ✓

Correct

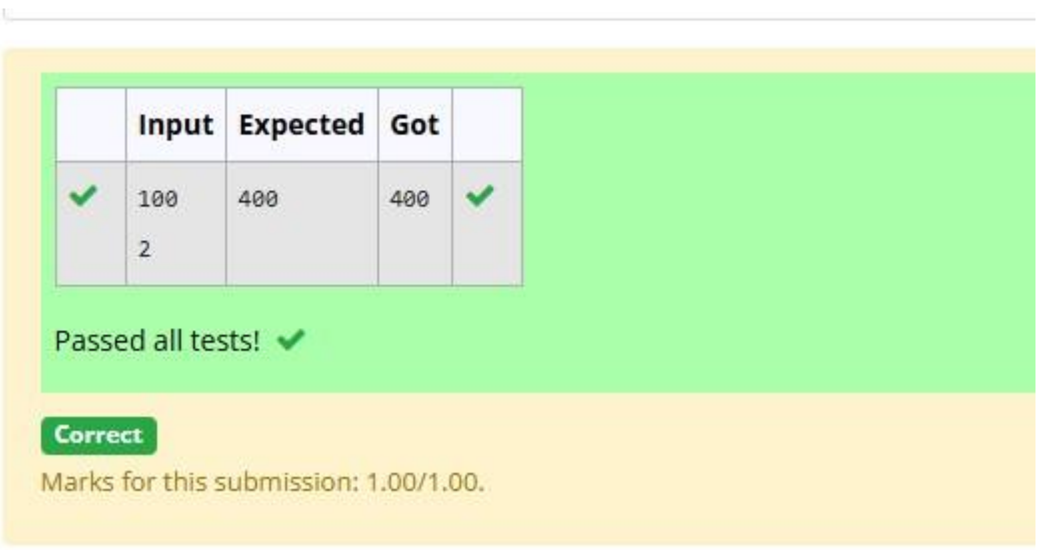
Marks for this submission: 1.00/1.00.


Question 4 | Correct Mark 1.00 out of 1.00 Flag question

Baba is very kind to beggars and every day Baba donates half of the amount he has when ever a beggar requests him. The money M left in Baba's hand is passed as the input and the number of beggars B who received the alms are passed as the input. The program must print the money Baba had in the beginning of the day.


Answer: (penalty regime: 0 %)

```
#include<stdio.h>
int main()
{
    int m,b;
    scanf("%d %d",&m,&b);
    int start=m;
    int i;
    for(i=0;i<b;i++)
    {
        start=start*2;
    }
    printf("%d",start);
    return 0;
}
```

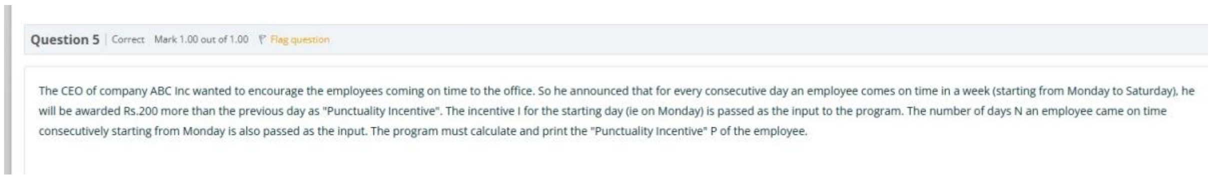


	Input	Expected	Got	
✓	100 2	400	400	✓

Passed all tests! ✓

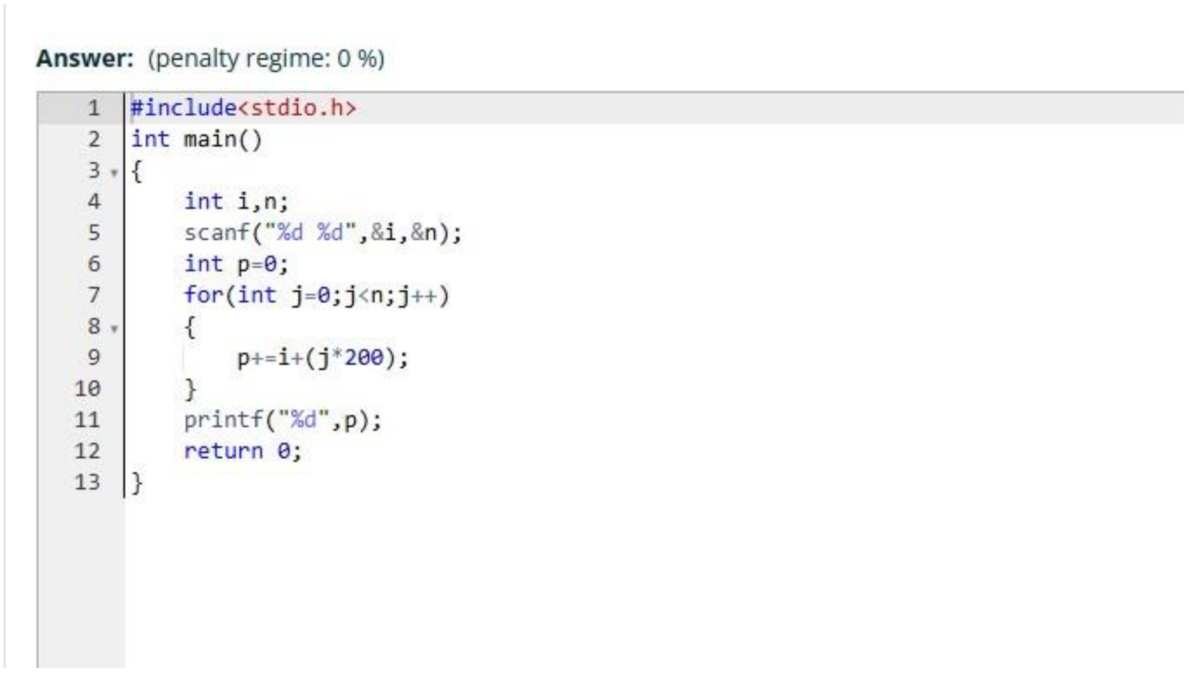
Correct

Marks for this submission: 1.00/1.00.



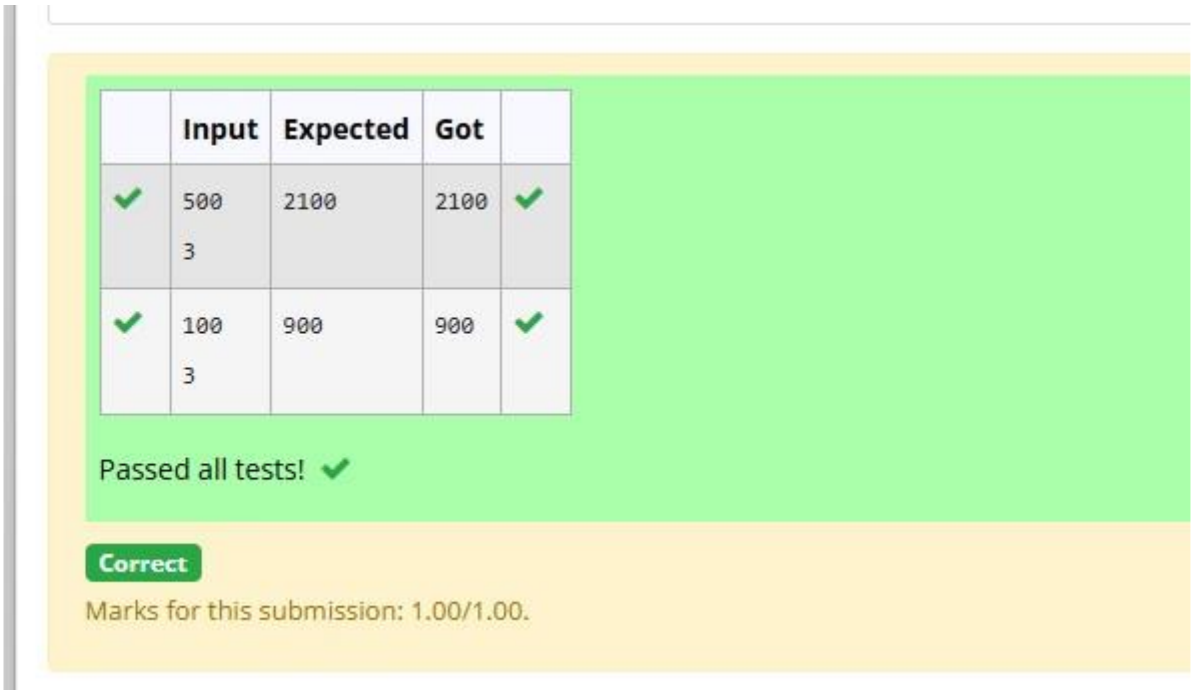
Question 5 | Correct Mark 1.00 out of 1.00 Flag question

The CEO of company ABC Inc wanted to encourage the employees coming on time to the office. So he announced that for every consecutive day an employee comes on time in a week (starting from Monday to Saturday), he will be awarded Rs.200 more than the previous day as "Punctuality Incentive". The incentive I for the starting day (ie on Monday) is passed as the input to the program. The number of days N an employee came on time consecutively starting from Monday is also passed as the input. The program must calculate and print the "Punctuality Incentive" P of the employee.



Answer: (penalty regime: 0 %)

```
#include<stdio.h>
int main()
{
    int i,n;
    scanf("%d %d",&i,&n);
    int p=0;
    for(int j=0;j<n;j++)
    {
        p+=i+(j*200);
    }
    printf("%d",p);
    return 0;
}
```



	Input	Expected	Got	
✓	500 3	2100	2100	✓
✓	100 3	900	900	✓

Passed all tests! ✓

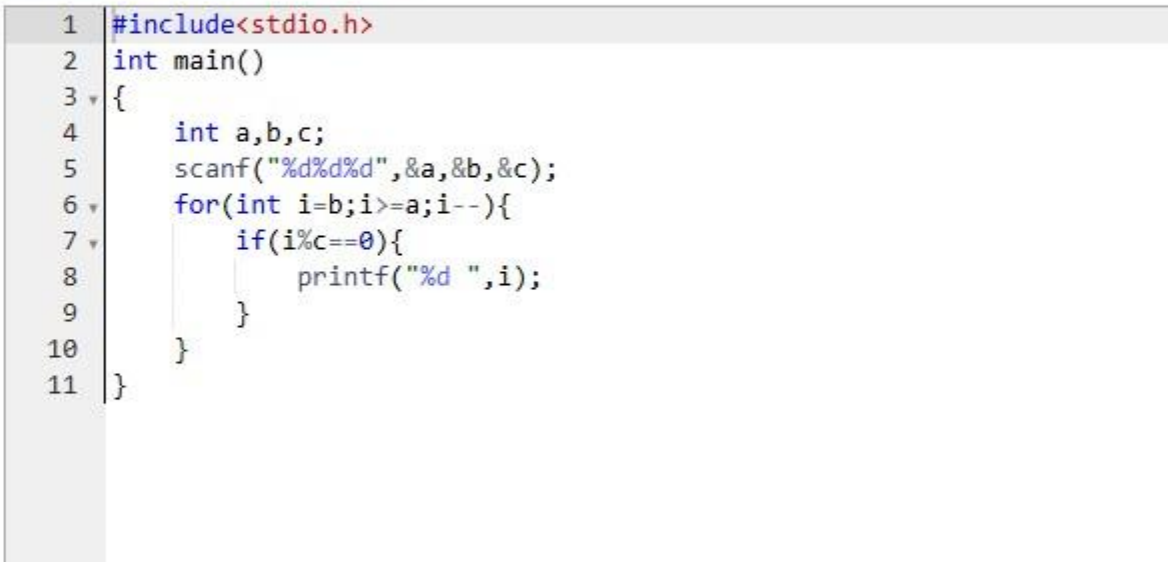
Correct

Marks for this submission: 1.00/1.00.

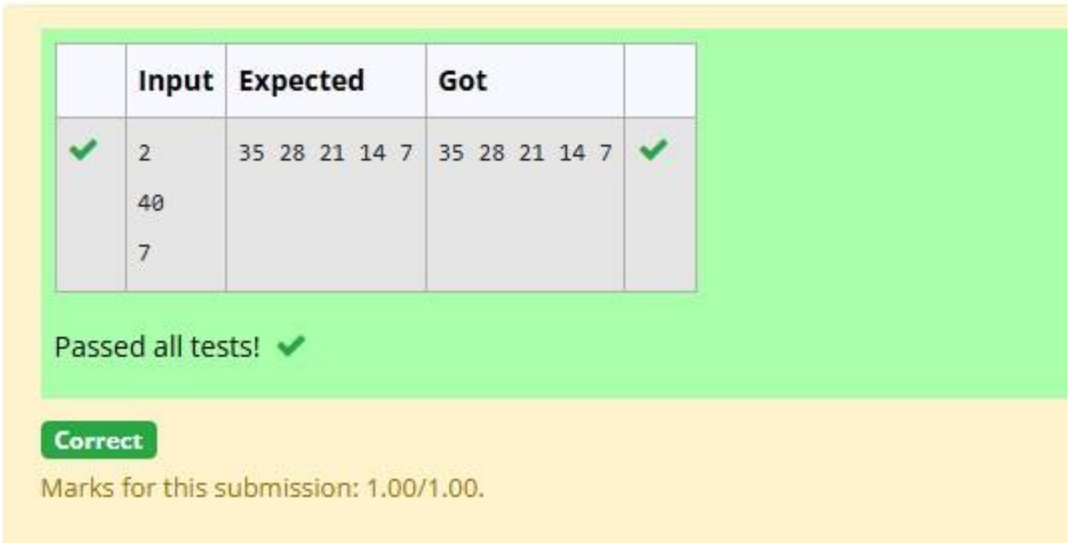
Question 6 | Correct Mark 1.00 out of 1.00 Flag question

Two numbers M and N are passed as the input. A number X is also passed as the input. The program must print the numbers divisible by X from N to M (inclusive of M and N).

Answer: (penalty regime: 0 %)



```c
#include<stdio.h>
int main()
{
    int a,b,c;
    scanf("%d%d%d",&a,&b,&c);
    for(int i=b;i>=a;i--){
        if(i%c==0){
            printf("%d ",i);
        }
    }
}
```

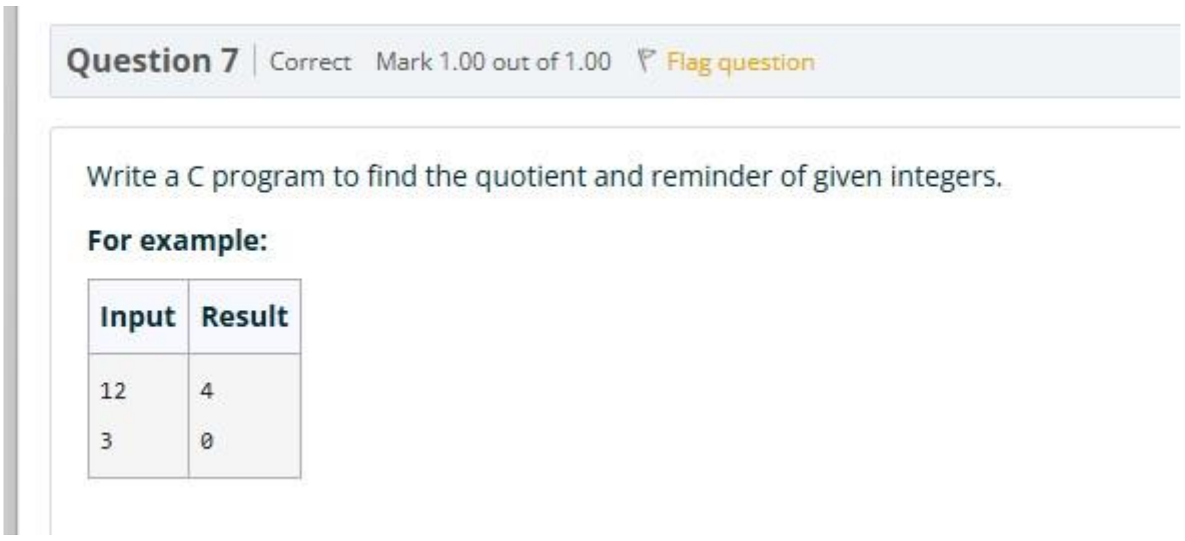


	Input	Expected	Got	
✓	2 40 7	35 28 21 14 7	35 28 21 14 7	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.



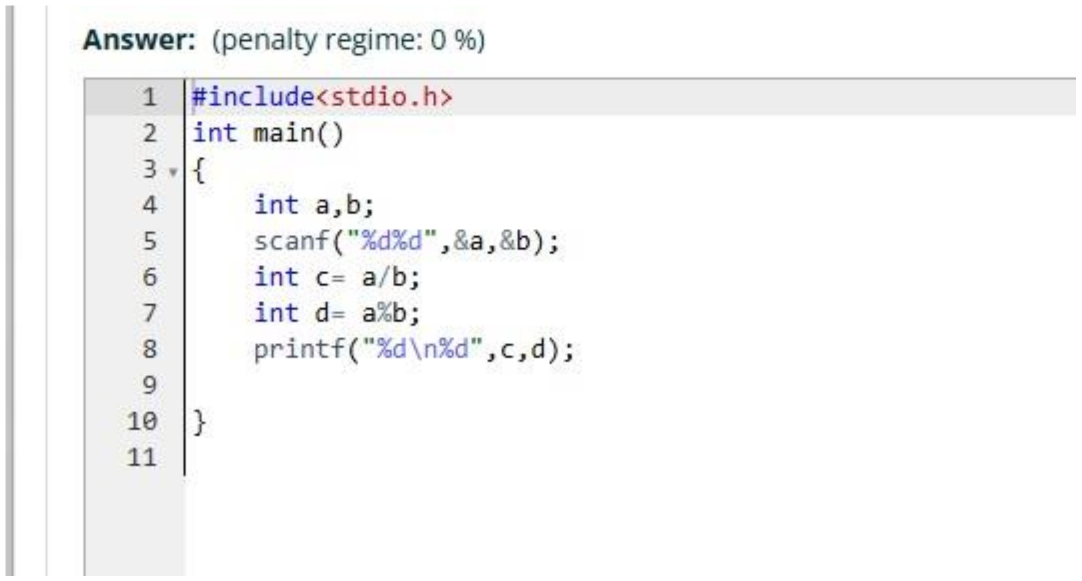
Question 7

Correct Mark 1.00 out of 1.00 Flag question

Write a C program to find the quotient and reminder of given integers.

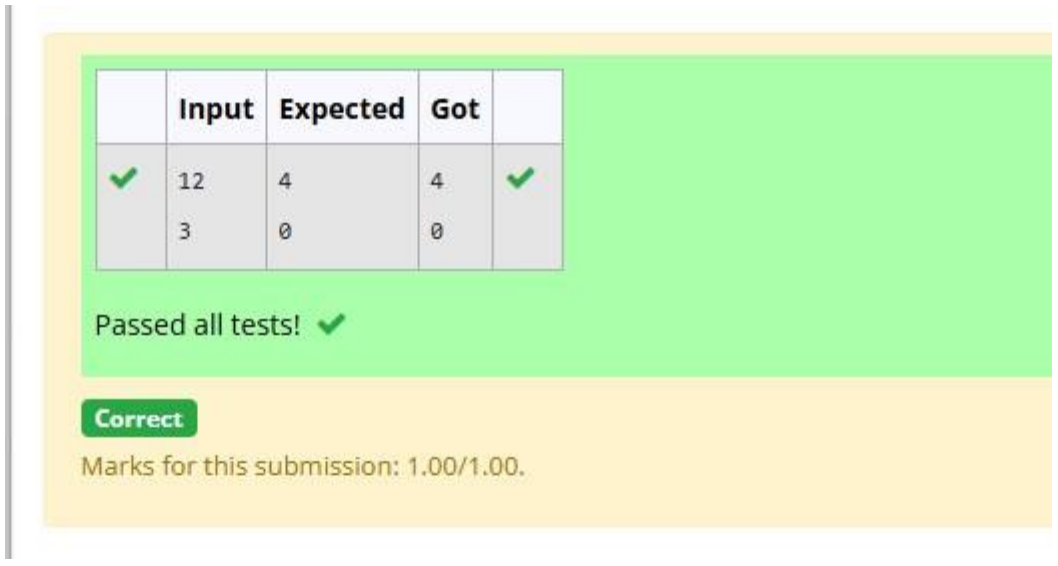
For example:

Input	Result
12 3	4 0



Answer: (penalty regime: 0 %)

```c
#include<stdio.h>
int main()
{
    int a,b;
    scanf("%d%d",&a,&b);
    int c= a/b;
    int d= a%b;
    printf("%d\n%d",c,d);

}
```

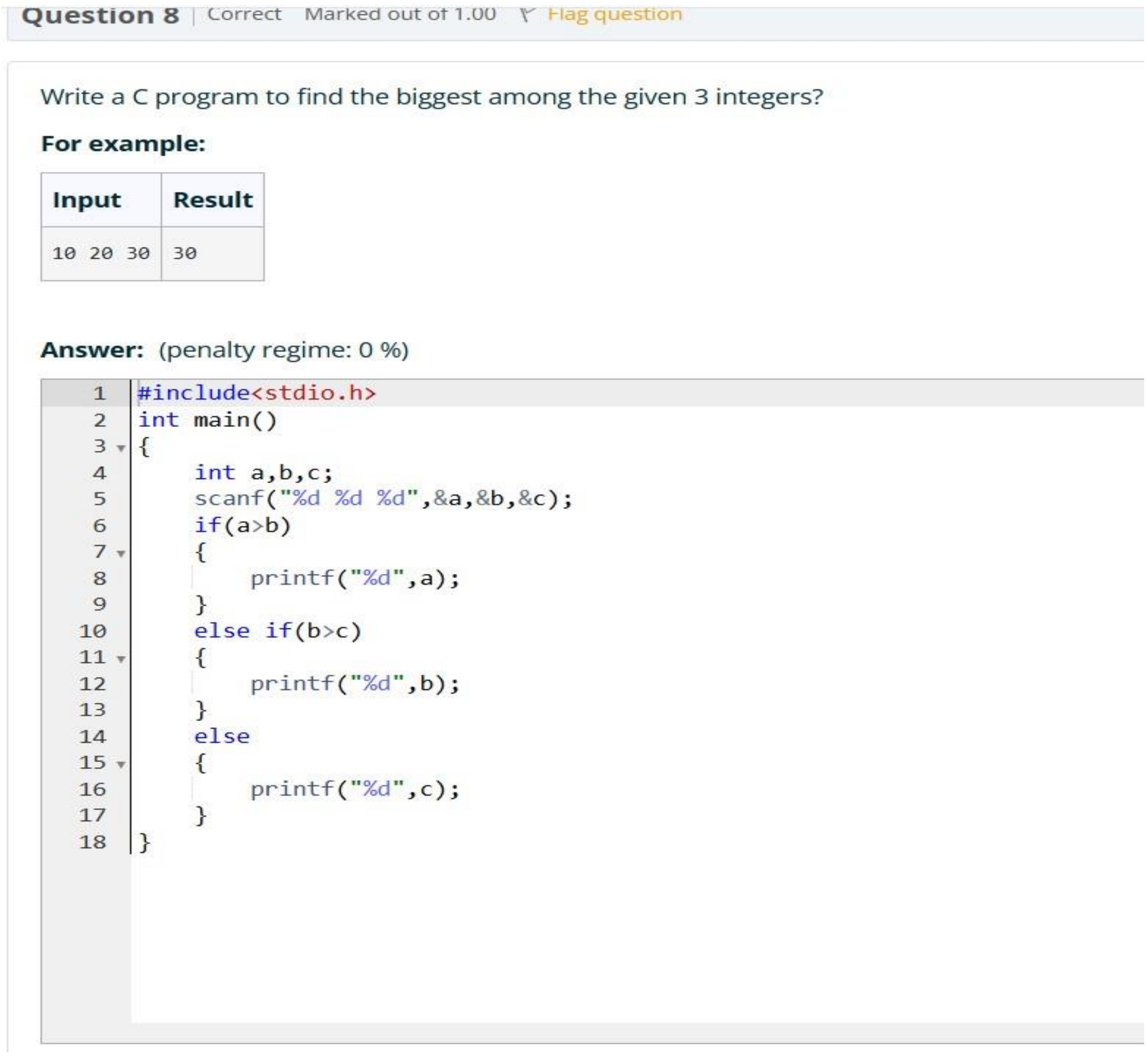


	Input	Expected	Got	
✓	12 3	4 0	4 0	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.


Question 8 | Correct Marked out of 1.00 Flag question

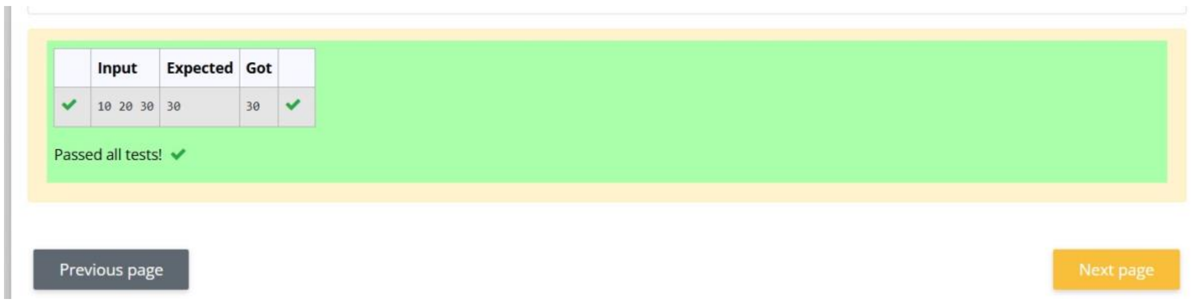
Write a C program to find the biggest among the given 3 integers?

For example:

Input	Result
10 20 30	30

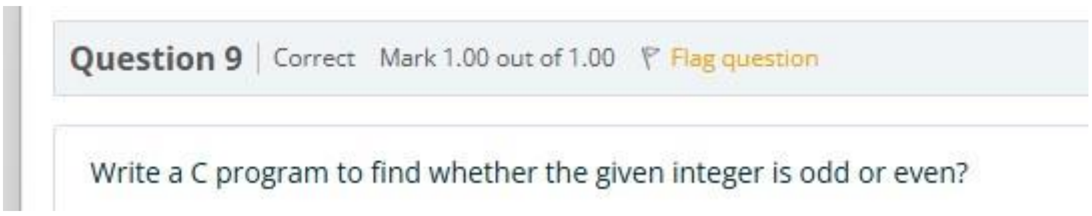
Answer: (penalty regime: 0 %)

```
#include<stdio.h>
int main()
{
    int a,b,c;
    scanf("%d %d %d",&a,&b,&c);
    if(a>b)
    {
        printf("%d",a);
    }
    else if(b>c)
    {
        printf("%d",b);
    }
    else
    {
        printf("%d",c);
    }
}
```



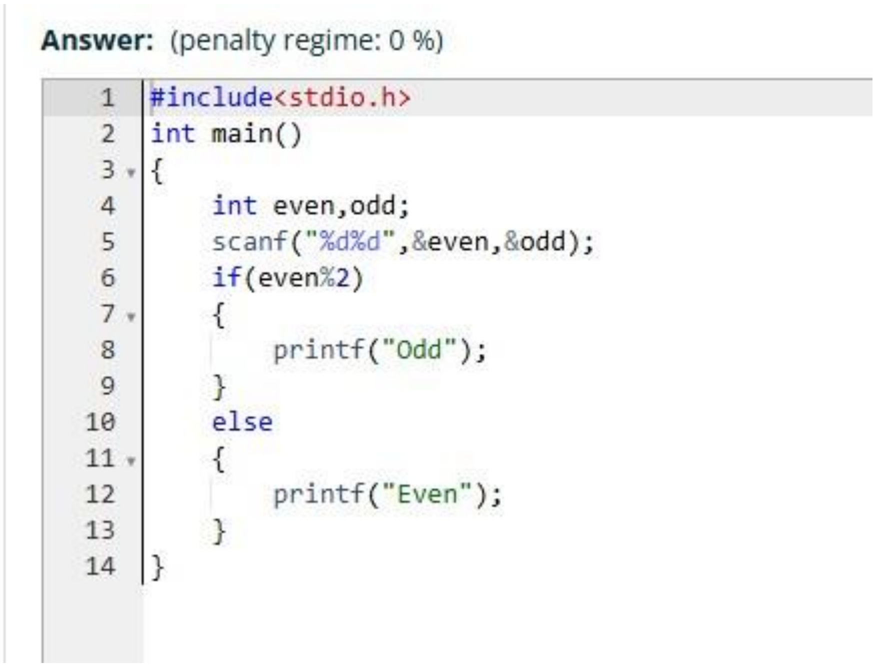
	Input	Expected	Got	
✓	10 20 30	30	30	✓

Passed all tests! ✓



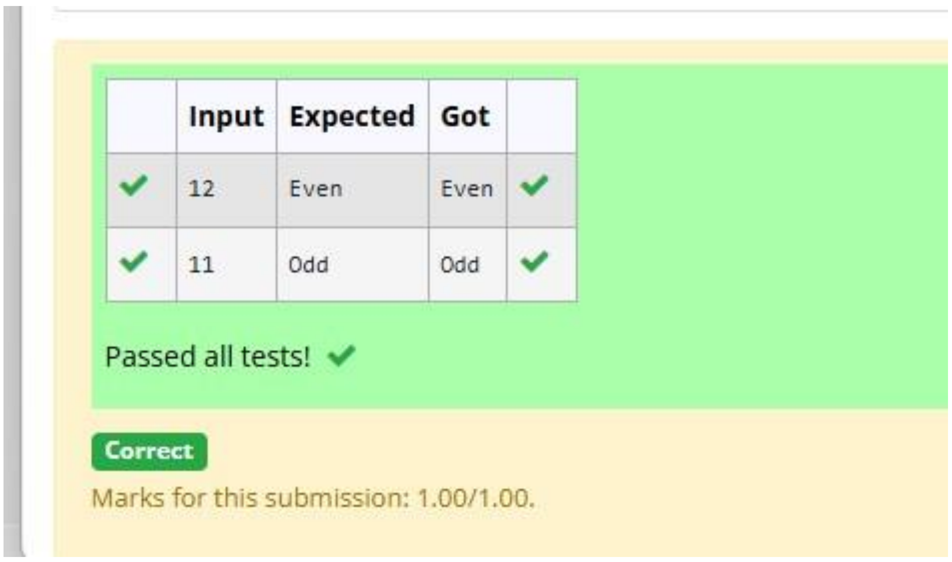
Question 9 | Correct Mark 1.00 out of 1.00 Flag question

Write a C program to find whether the given integer is odd or even?



Answer: (penalty regime: 0 %)

```c
#include<stdio.h>
int main()
{
    int even,odd;
    scanf("%d%d",&even,&odd);
    if(even%2)
    {
        printf("Odd");
    }
    else
    {
        printf("Even");
    }
}
```

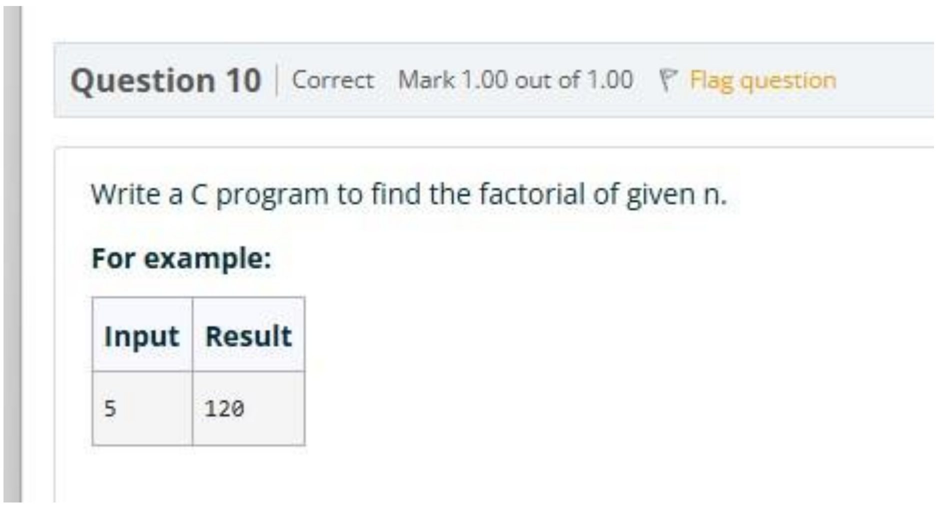


	Input	Expected	Got	
✓	12	Even	Even	✓
✓	11	Odd	Odd	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

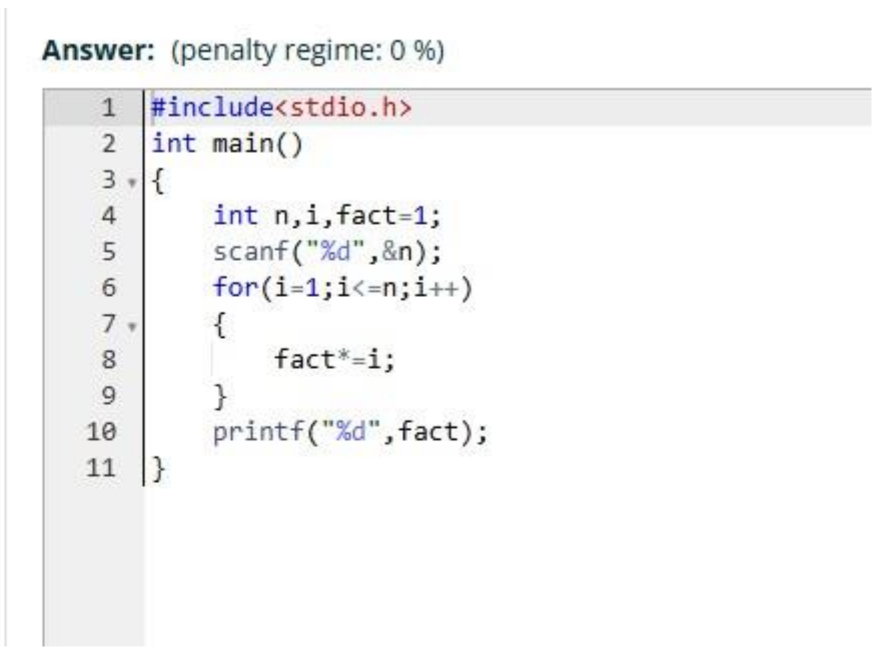


Question 10 | Correct Mark 1.00 out of 1.00 Flag question

Write a C program to find the factorial of given n.

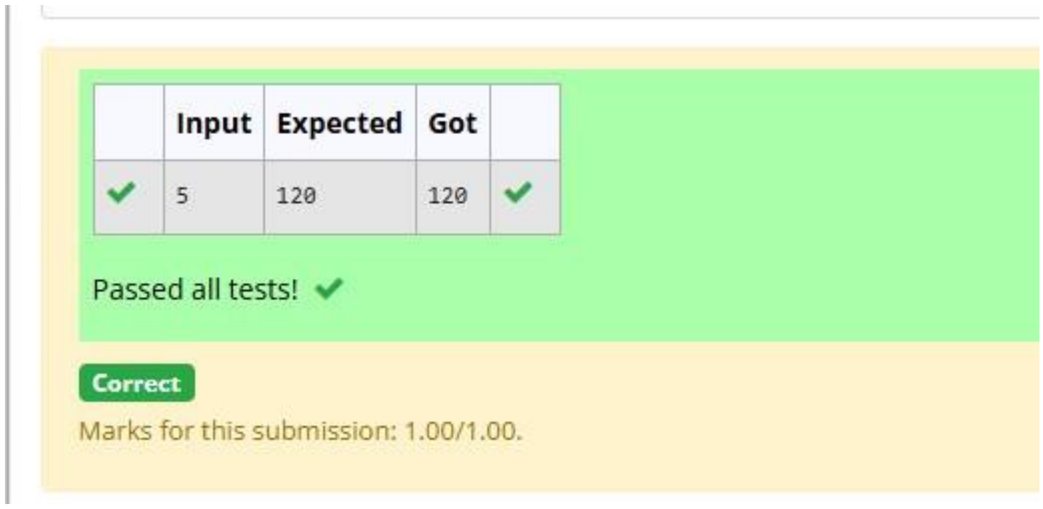
For example:

Input	Result
5	120



Answer: (penalty regime: 0 %)

```c
#include<stdio.h>
int main()
{
    int n,i,fact=1;
    scanf("%d",&n);
    for(i=1;i<=n;i++)
    {
        fact*=i;
    }
    printf("%d",fact);
}
```

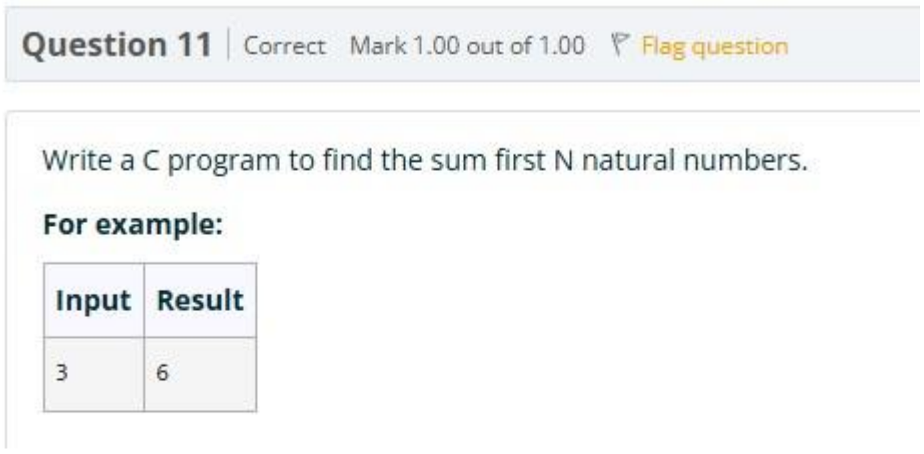


	Input	Expected	Got	
✓	5	120	120	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

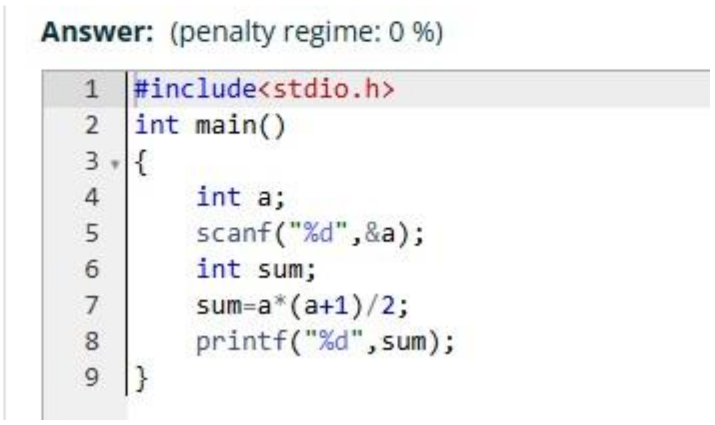


Question 11 | Correct Mark 1.00 out of 1.00 Flag question

Write a C program to find the sum first N natural numbers.

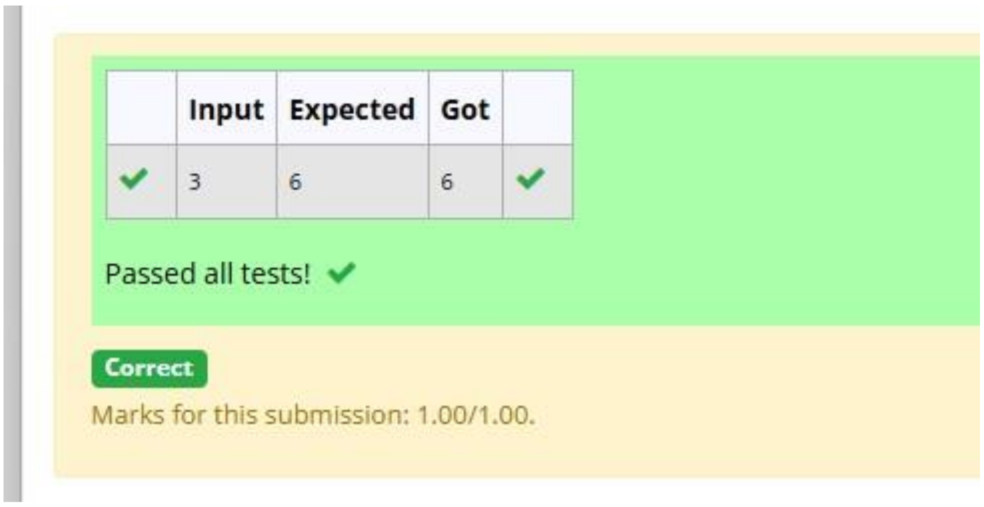
For example:

Input	Result
3	6



Answer: (penalty regime: 0 %)

```c
#include<stdio.h>
int main()
{
    int a;
    scanf("%d",&a);
    int sum;
    sum=a*(a+1)/2;
    printf("%d",sum);
}
```

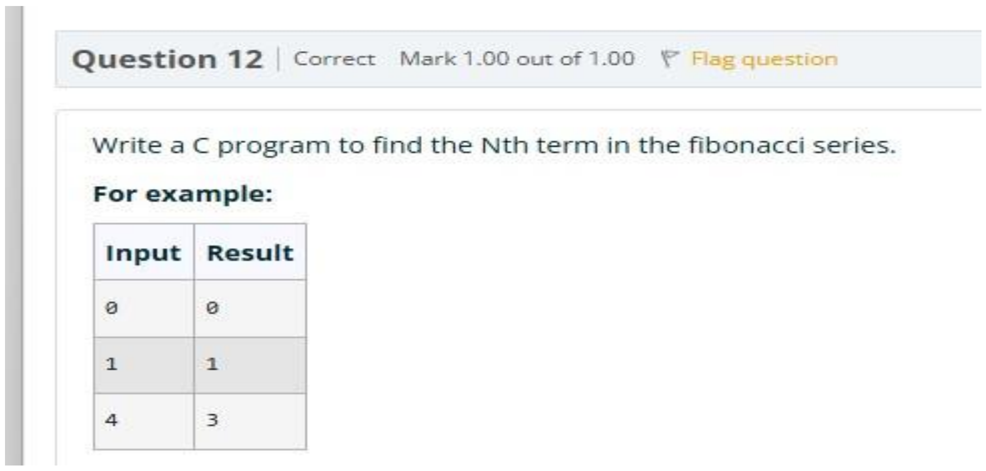


	Input	Expected	Got	
✓	3	6	6	✓

Passed all tests! ✓

Correct

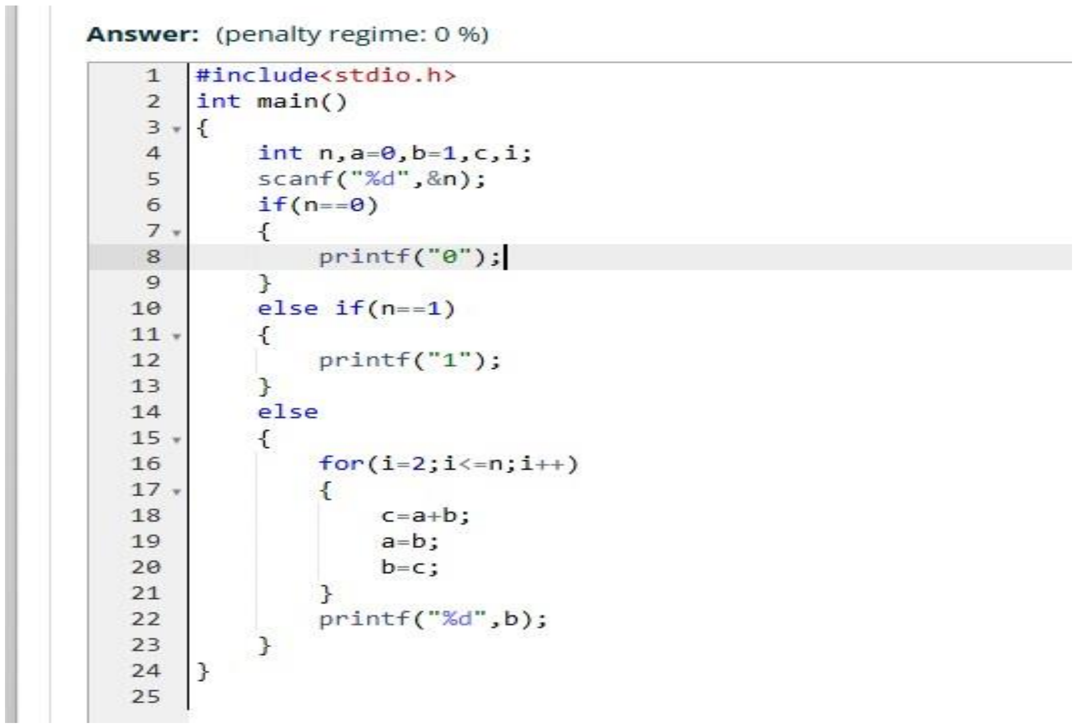
Marks for this submission: 1.00/1.00.


Question 12 | Correct Mark 1.00 out of 1.00 Flag question

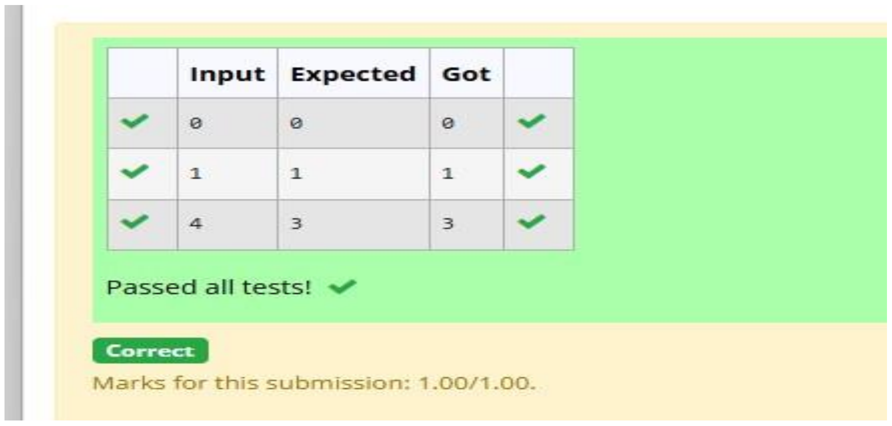
Write a C program to find the Nth term in the fibonacci series.

For example:

Input	Result
0	0
1	1
4	3


Answer: (penalty regime: 0 %)

```c
#include<stdio.h>
int main()
{
     int n,a=0,b=1,c,i;
     scanf("%d",&n);
     if(n==0)
     {
         printf("0");
     }
     else if(n==1)
     {
         printf("1");
     }
     else
     {
         for(i=2;i<=n;i++)
         {
             c=a+b;
             a=b;
             b=c;
         }
         printf("%d",b);
     }
}
```

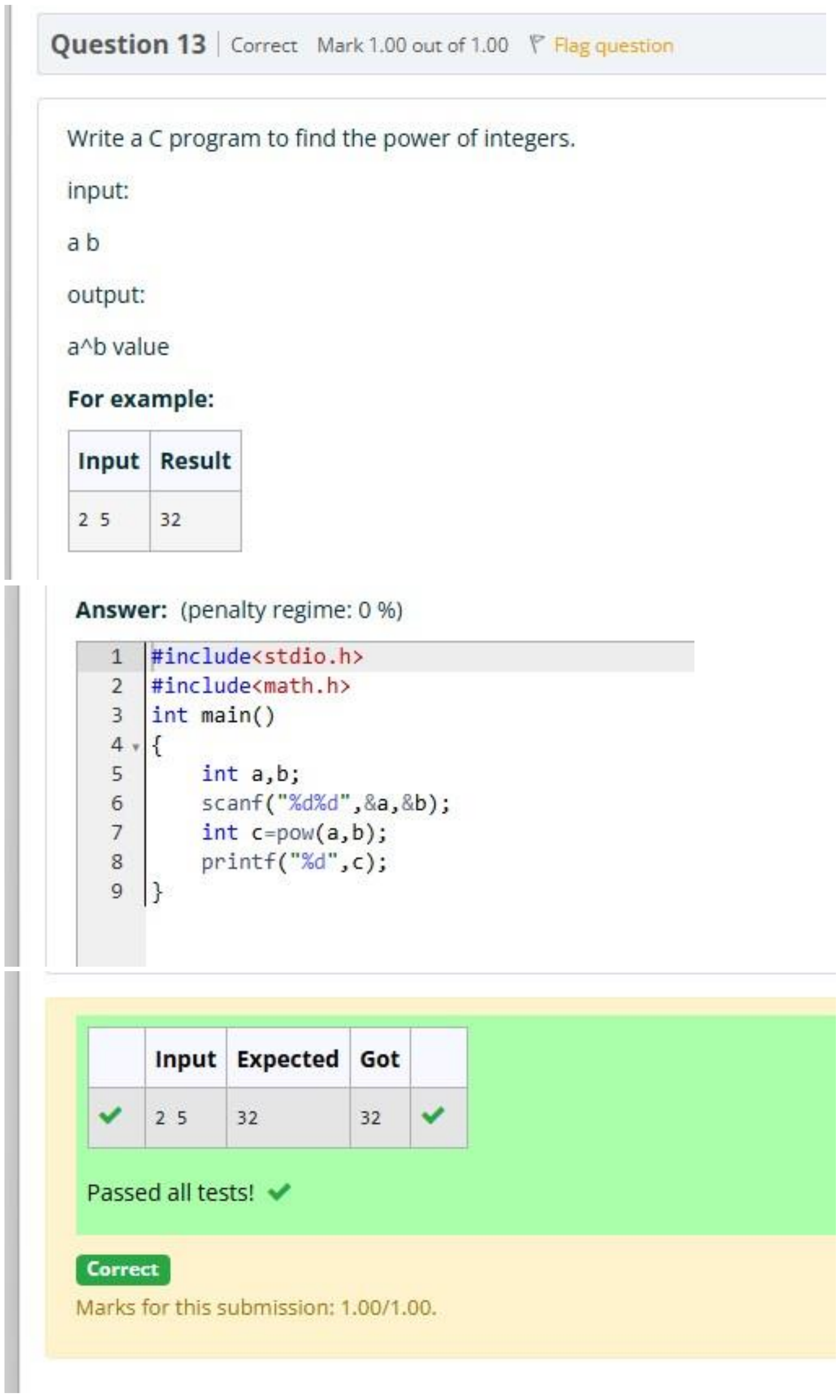


	Input	Expected	Got	
✓	0	0	0	✓
✓	1	1	1	✓
✓	4	3	3	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.



Question 13

Correct Mark 1.00 out of 1.00 Flag question

Write a C program to find the power of integers.

input:

a b

output:

a^b value

For example:

Input	Result
2 5	32

Answer: (penalty regime: 0 %)

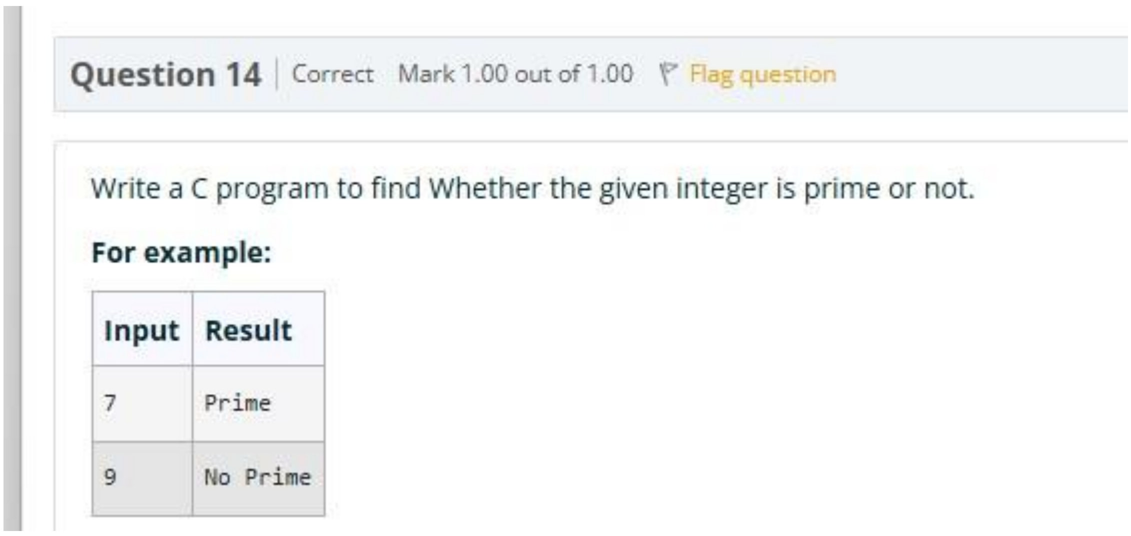
```c
#include<stdio.h>
#include<math.h>
int main()
{
    int a,b;
    scanf("%d%d",&a,&b);
    int c=pow(a,b);
    printf("%d",c);
}
```

	Input	Expected	Got	
✓	2 5	32	32	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

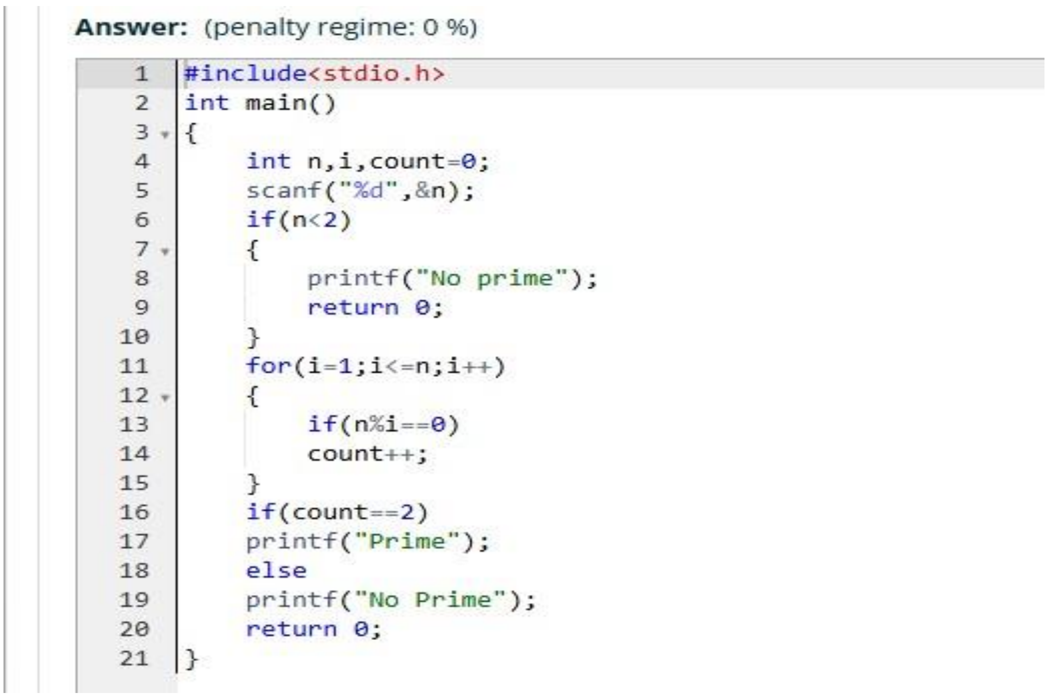


Question 14 | Correct Mark 1.00 out of 1.00 Flag question

Write a C program to find Whether the given integer is prime or not.

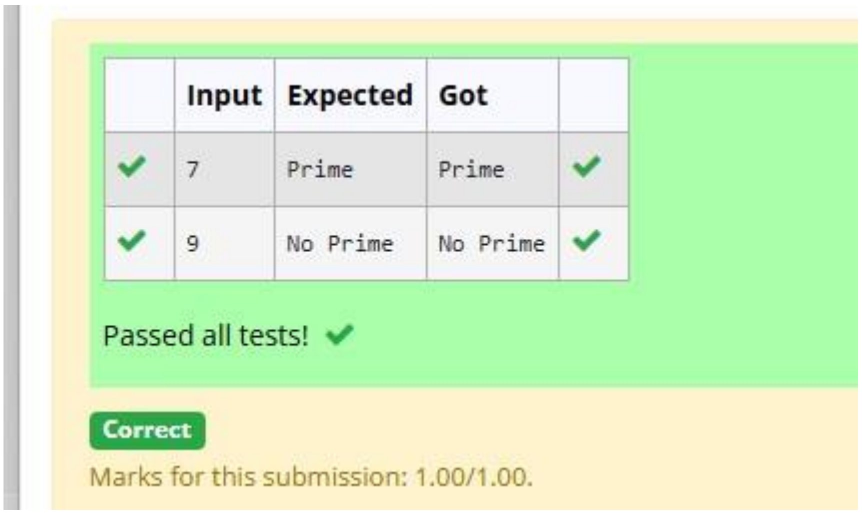
For example:

Input	Result
7	Prime
9	No Prime



Answer: (penalty regime: 0 %)

```c
#include<stdio.h>
int main()
{
    int n,i,count=0;
    scanf("%d",&n);
    if(n<2)
    {
        printf("No prime");
        return 0;
    }
    for(i=1;i<=n;i++)
    {
        if(n%i==0)
        count++;
    }
    if(count==2)
    printf("Prime");
    else
    printf("No Prime");
    return 0;
}
```

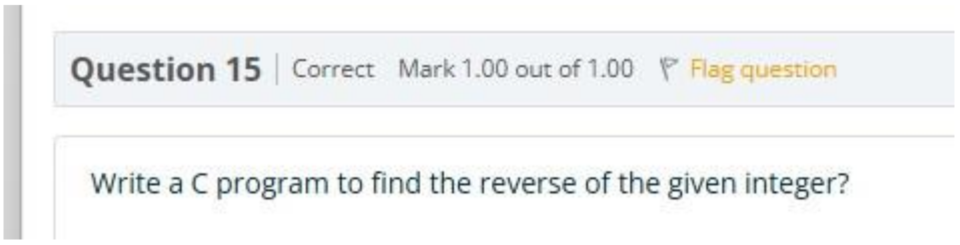


	Input	Expected	Got	
✓	7	Prime	Prime	✓
✓	9	No Prime	No Prime	✓

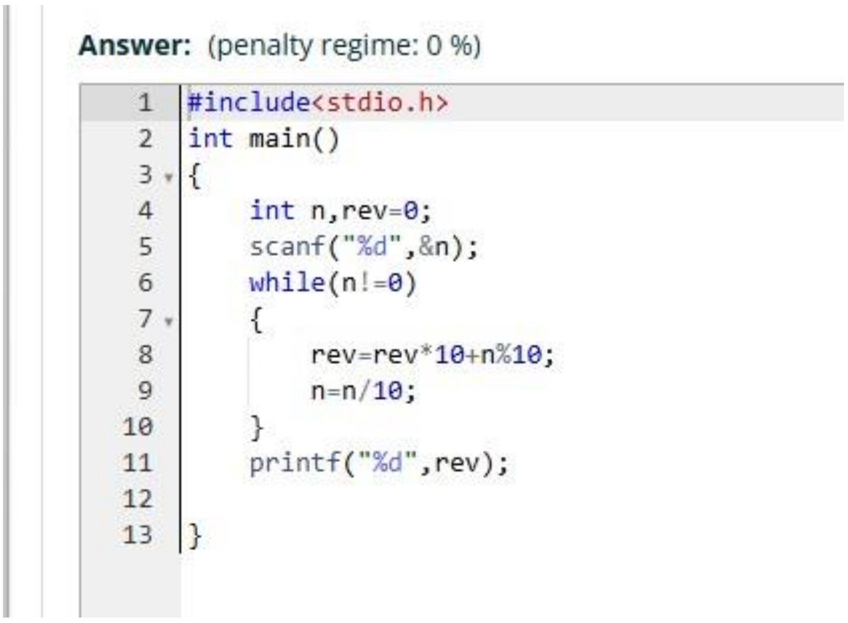
Passed all tests! ✓

Correct

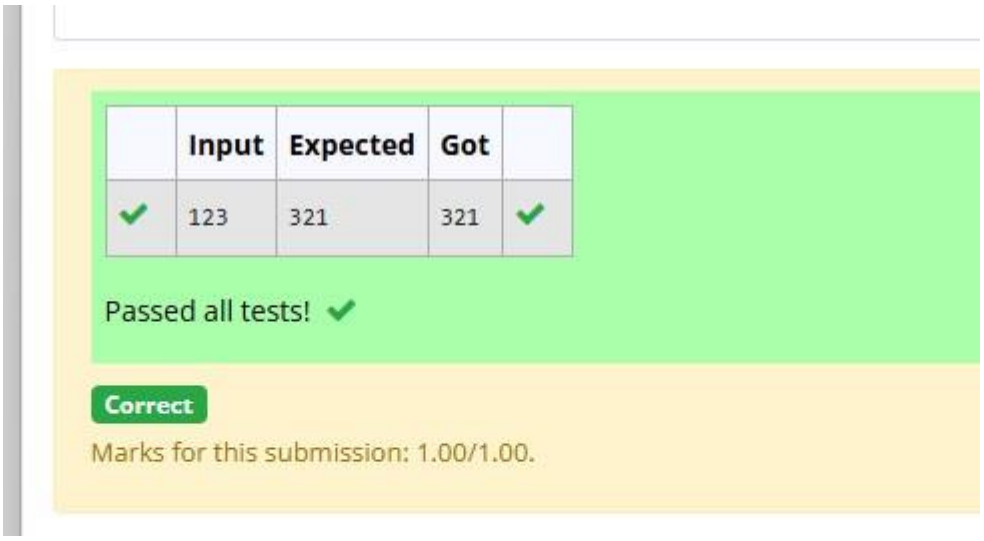
Marks for this submission: 1.00/1.00.


Question 15 | Correct Mark 1.00 out of 1.00 Flag question

Write a C program to find the reverse of the given integer?


Answer: (penalty regime: 0 %)

```c
#include<stdio.h>
int main()
{
    int n,rev=0;
    scanf("%d",&n);
    while(n!=0)
    {
        rev=rev*10+n%10;
        n=n/10;
    }
    printf("%d",rev);

}
```



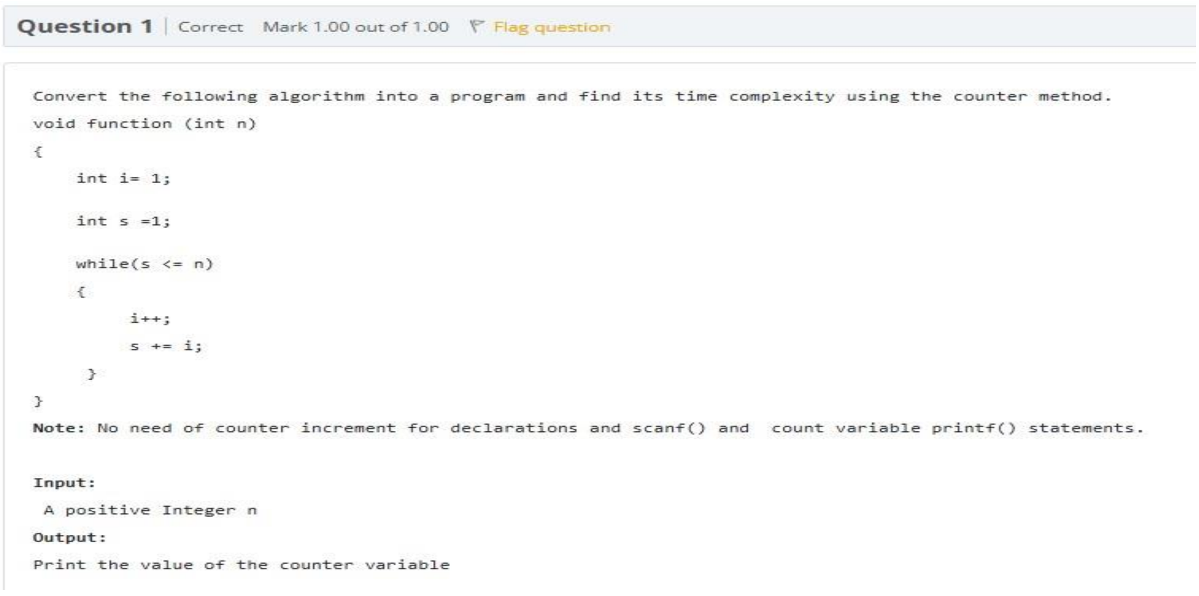
	Input	Expected	Got	
✓	123	321	321	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

2.Finding Time Complexity Q1:



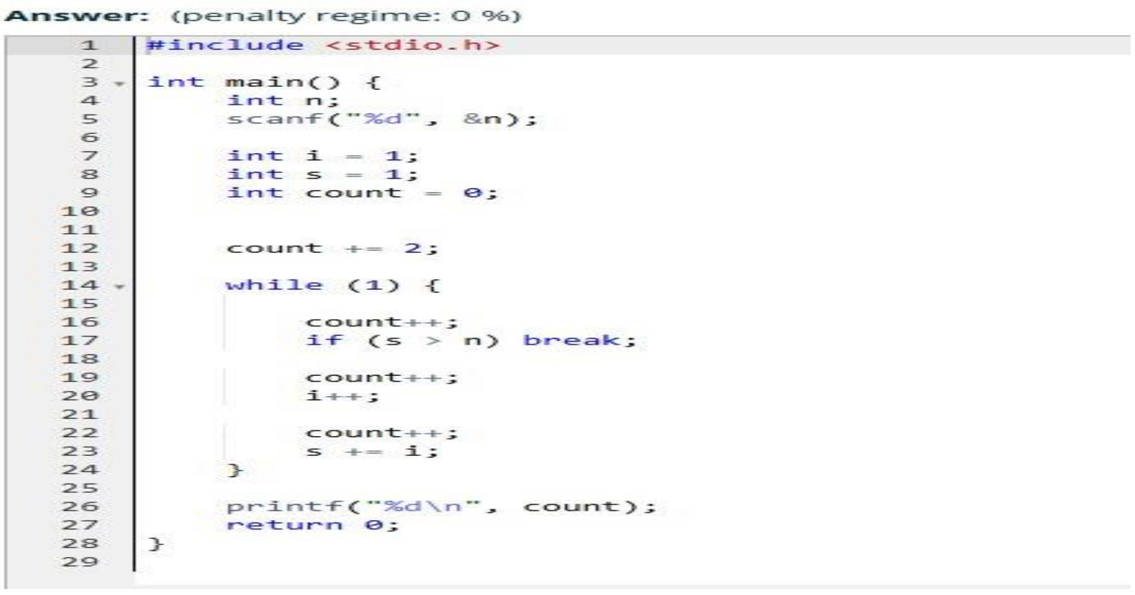
Question 1 | Correct Mark 1.00 out of 1.00 Flag question

```
Convert the following algorithm into a program and find its time complexity using the counter method.
void function (int n)
{
    int i= 1;

    int s =1;

    while(s <= n)
    {
         i++;
         s += i;
     }
}
```

Note: No need of counter increment for declarations and scanf() and count variable printf() statements.

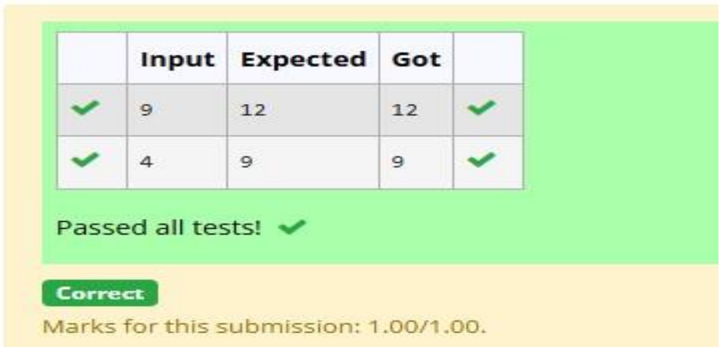
Input:
 A positive Integer n

Output:
Print the value of the counter variable



Answer: (penalty regime: 0 %)

```
#include <stdio.h>

int main() {
    int n;
    scanf("%d", &n);

    int i = 1;
    int s = 1;
    int count = 0;


    count += 2;

    while (1) {

        count++;
        if (s > n) break;

        count++;
        i++;

        count++;
        s += i;
    }

    printf("%d\n", count);
    return 0;
}
```



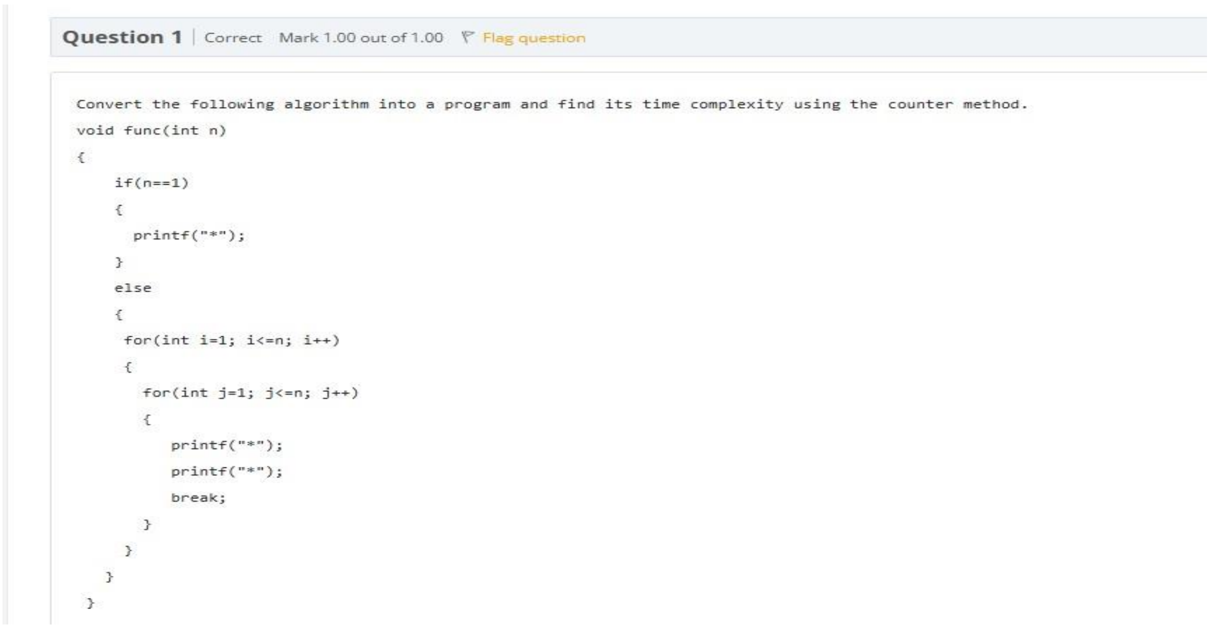
	Input	Expected	Got	
✓	9	12	12	✓
✓	4	9	9	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

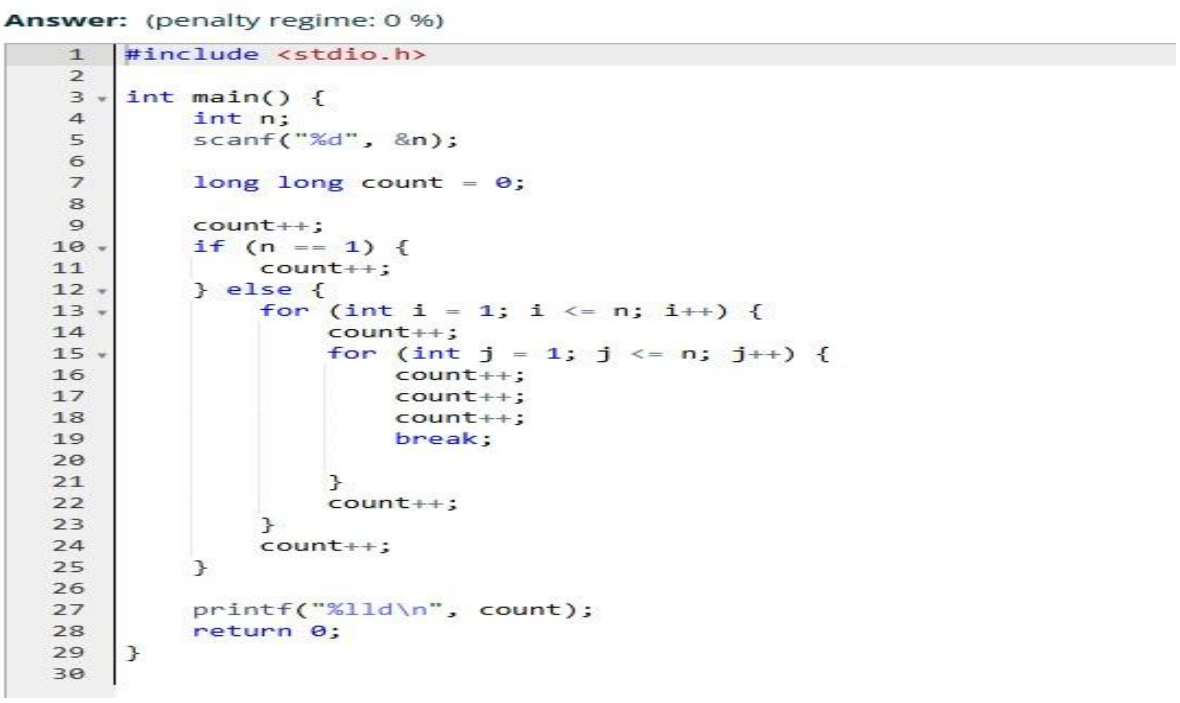
Q2:



Question 1 | Correct Mark 1.00 out of 1.00 Flag question

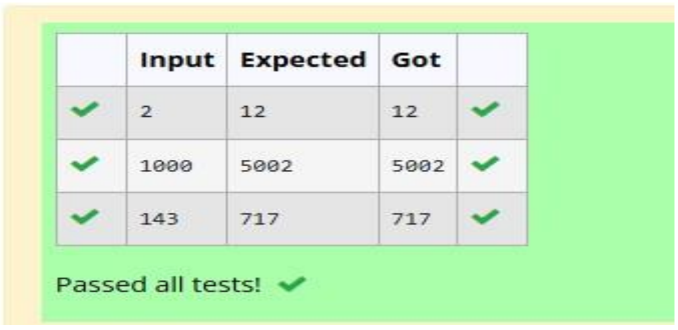
Convert the following algorithm into a program and find its time complexity using the counter method.

```
void func(int n)
{
    if(n==1)
    {
      printf("*");
    }
    else
    {
     for(int i=1; i<=n; i++)
     {
       for(int j=1; j<=n; j++)
       {
          printf("*");
          printf("*");
          break;
       }
     }
   }
 }
```



Answer: (penalty regime: 0 %)

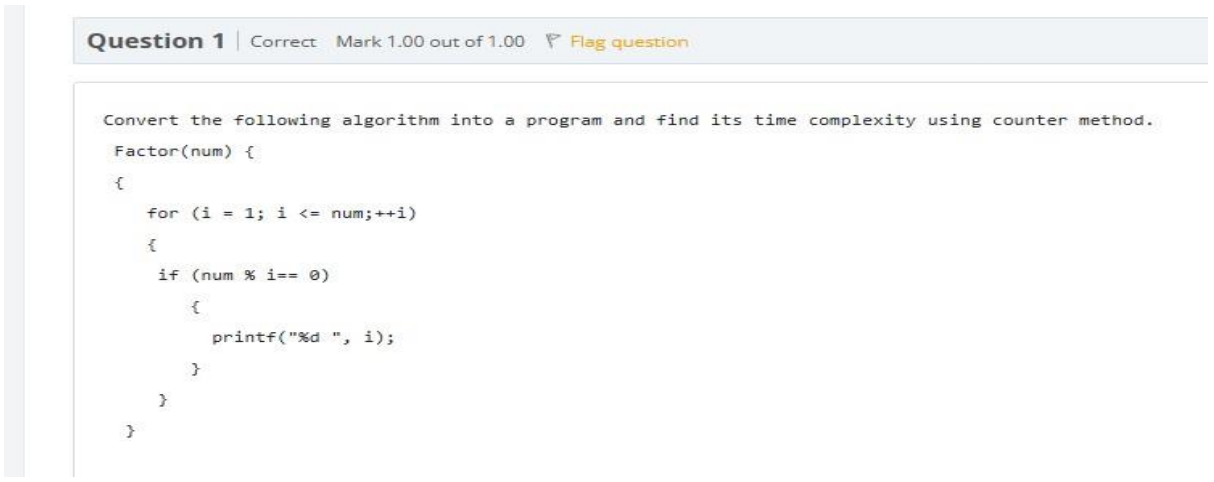
```c
#include <stdio.h>

int main() {
    int n;
    scanf("%d", &n);

    long long count = 0;

    count++;
    if (n == 1) {
        count++;
    } else {
        for (int i = 1; i <= n; i++) {
            count++;
            for (int j = 1; j <= n; j++) {
                count++;
                count++;
                count++;
                break;

            }
            count++;
        }
        count++;
    }

    printf("%lld\n", count);
    return 0;
}

```



	Input	Expected	Got	
✓	2	12	12	✓
✓	1000	5002	5002	✓
✓	143	717	717	✓

Passed all tests! ✓

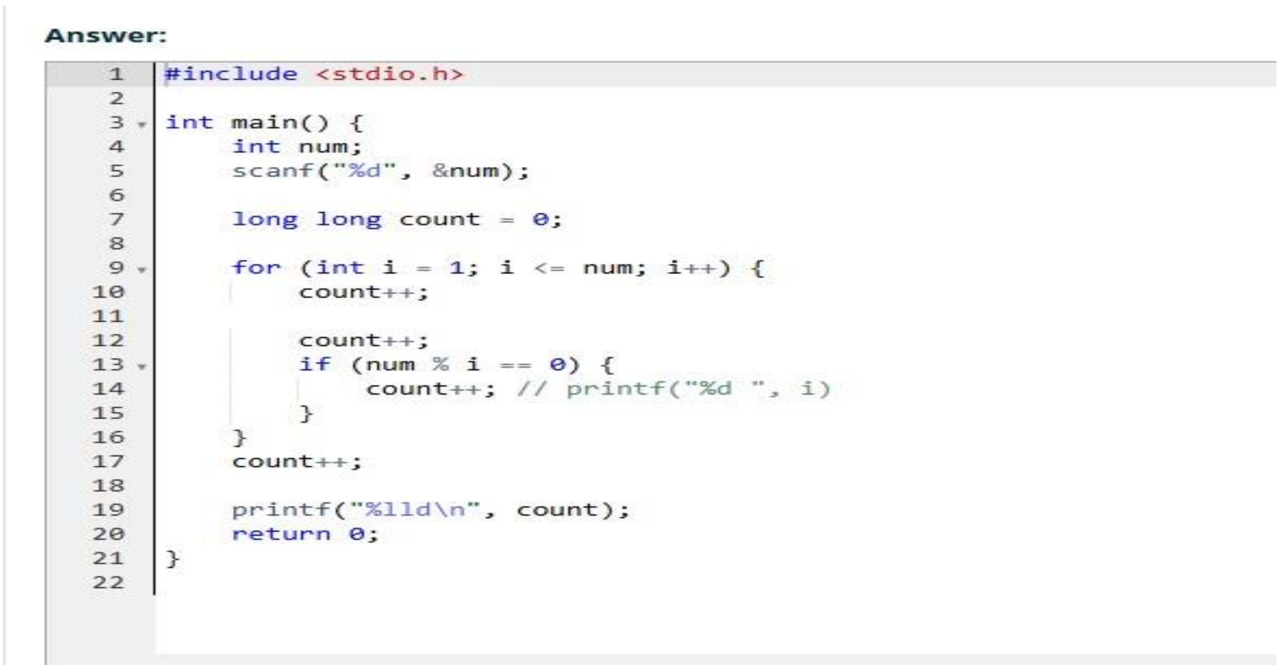
Q3:



Question 1 | Correct Mark 1.00 out of 1.00 Flag question

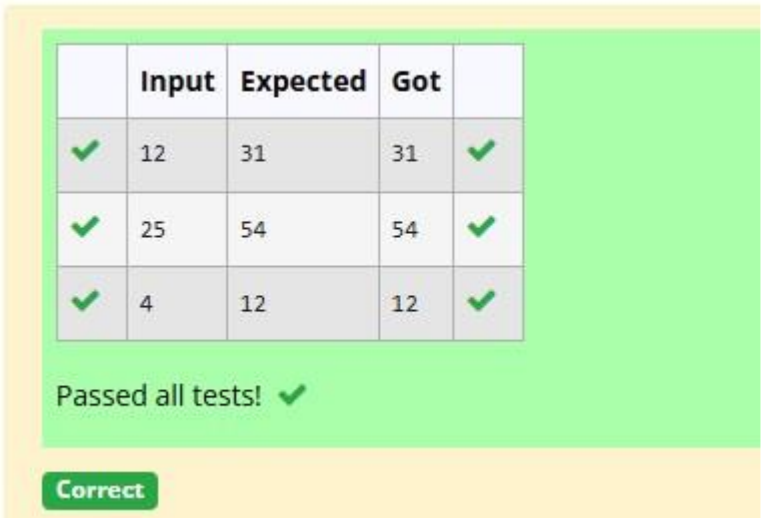
Convert the following algorithm into a program and find its time complexity using counter method.

```
Factor(num) {
{
   for (i = 1; i <= num;++i)
   {
    if (num % i== 0)
       {
         printf("%d ", i);
       }
    }
 }
```



Answer:

```c
#include <stdio.h>

int main() {
    int num;
    scanf("%d", &num);

    long long count = 0;

    for (int i = 1; i <= num; i++) {
        count++;

        count++;
        if (num % i == 0) {
            count++; // printf("%d ", i)
        }
    }
    count++;

    printf("%lld\n", count);
    return 0;
}
```

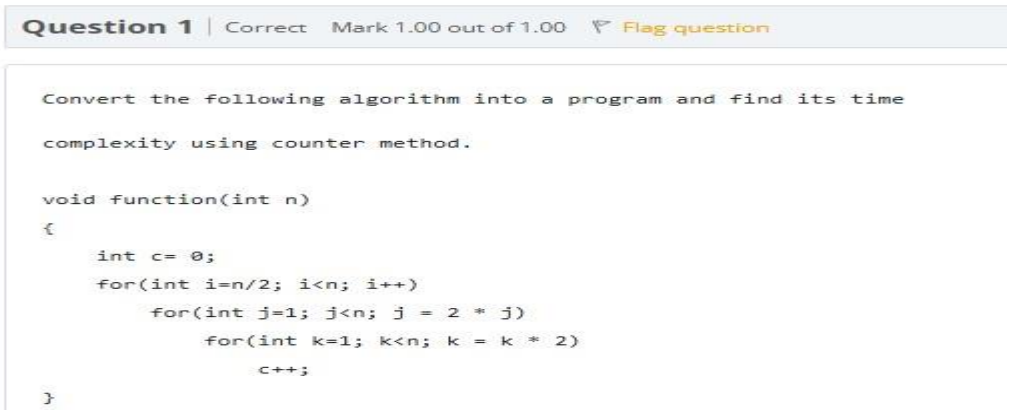


	Input	Expected	Got	
✓	12	31	31	✓
✓	25	54	54	✓
✓	4	12	12	✓

Passed all tests! ✓

Correct

Q4:

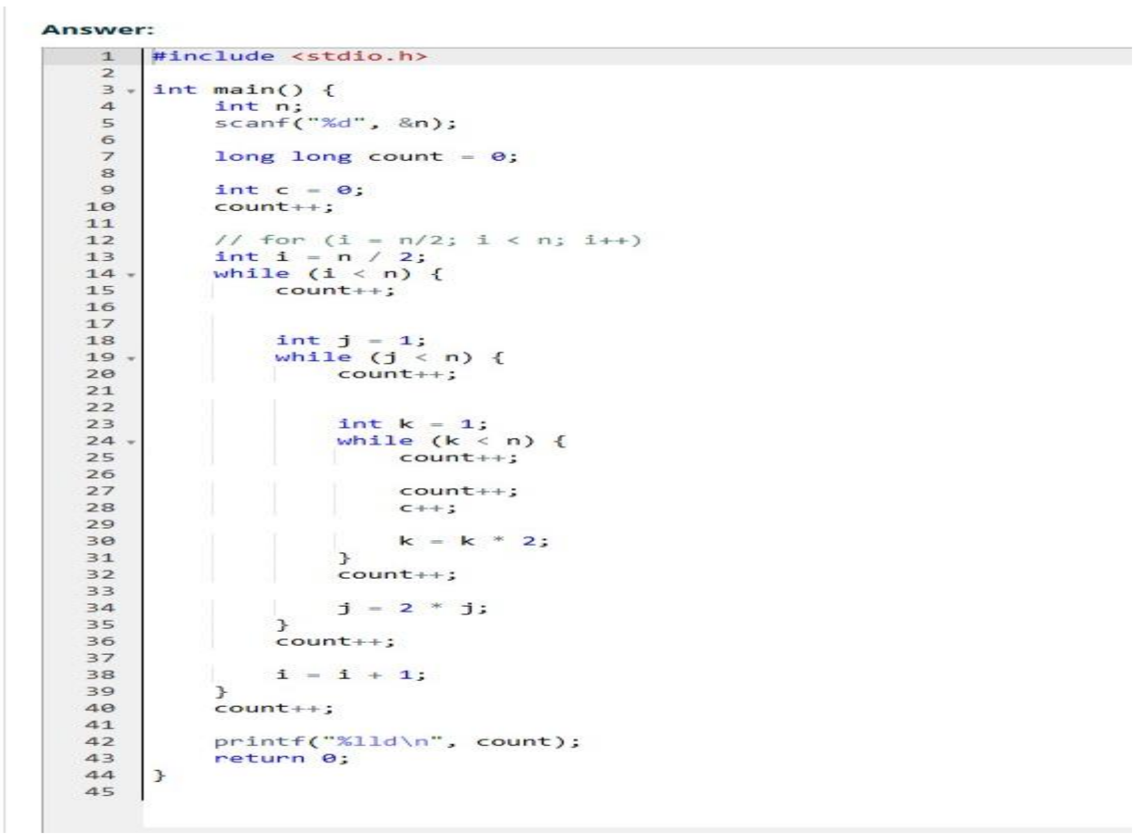


Question 1 | Correct Mark 1.00 out of 1.00 Flag question

Convert the following algorithm into a program and find its time

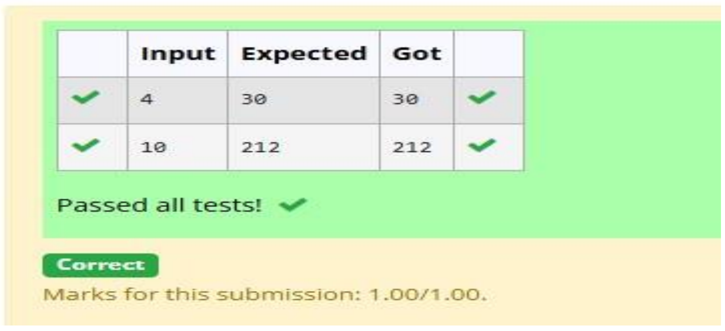
complexity using counter method.

```
void function(int n)
{
     int c= 0;
     for(int i=n/2; i<n; i++)
          for(int j=1; j<n; j = 2 * j)
               for(int k=1; k<n; k = k * 2)
                    c++;
}
```



Answer:

```
#include <stdio.h>

int main() {
    int n;
    scanf("%d", &n);

    long long count = 0;

    int c = 0;
    count++;

    // for (i = n/2; i < n; i++)
    int i = n / 2;
    while (i < n) {
        count++;


        int j = 1;
        while (j < n) {
            count++;


            int k = 1;
            while (k < n) {
                count++;

                count++;
                c++;

                k = k * 2;
            }
            count++;

            j = 2 * j;
        }
        count++;

        i = i + 1;
    }
    count++;

    printf("%lld\n", count);
    return 0;
}

```



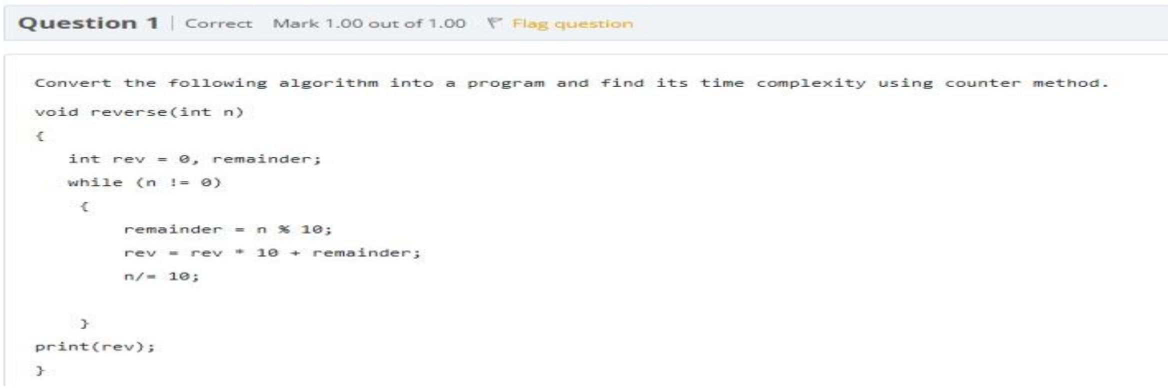
	Input	Expected	Got	
✓	4	30	30	✓
✓	10	212	212	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

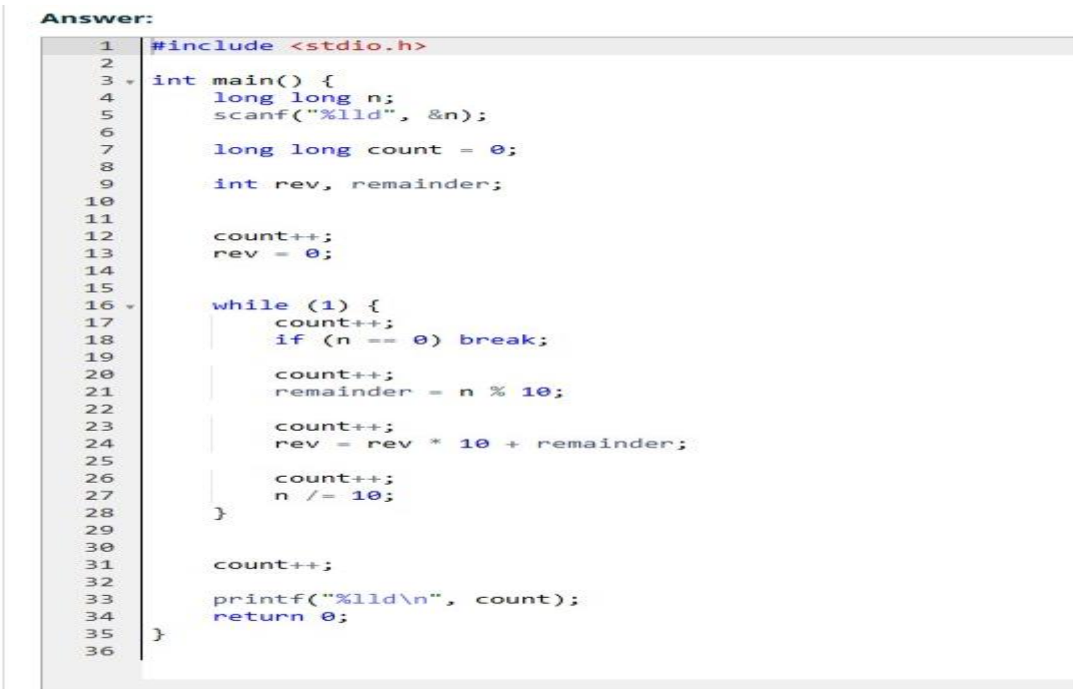
Q5:



Question 1 | Correct Mark 1.00 out of 1.00 Flag question

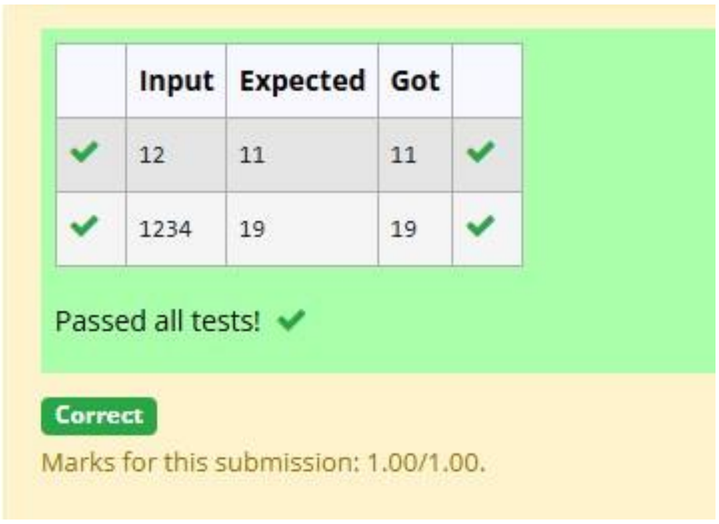
Convert the following algorithm into a program and find its time complexity using counter method.

```
void reverse(int n)
{
   int rev = 0, remainder;
   while (n != 0)
    {
        remainder = n % 10;
        rev = rev * 10 + remainder;
        n/= 10;

    }
print(rev);
}
```



Answer:

```
#include <stdio.h>

int main() {
     long long n;
     scanf("%lld", &n);

     long long count = 0;

     int rev, remainder;


     count++;
     rev = 0;


     while (1) {
         count++;
         if (n == 0) break;

         count++;
         remainder = n % 10;

         count++;
         rev = rev * 10 + remainder;

         count++;
         n /= 10;
     }


     count++;

     printf("%lld\n", count);
     return 0;
}
```



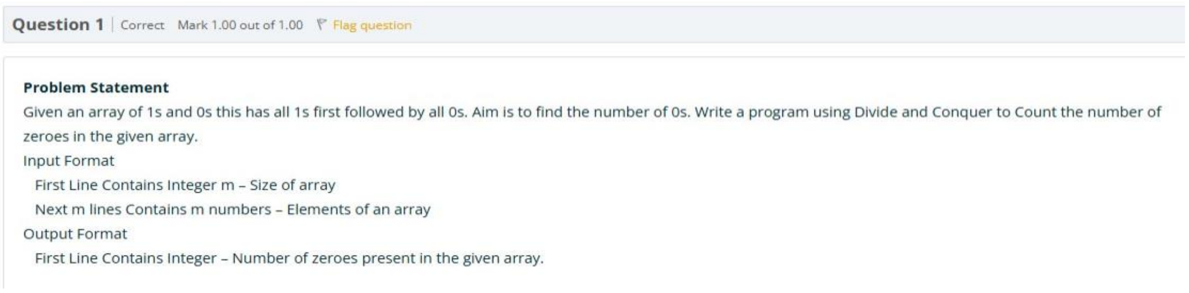
	Input	Expected	Got	
✓	12	11	11	✓
✓	1234	19	19	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

3.Divide And Conquer Q1:


Question 1 | Correct Mark 1.00 out of 1.00 Flag question

Problem Statement

Given an array of 1s and 0s this has all 1s first followed by all 0s. Aim is to find the number of 0s. Write a program using Divide and Conquer to Count the number of zeroes in the given array.

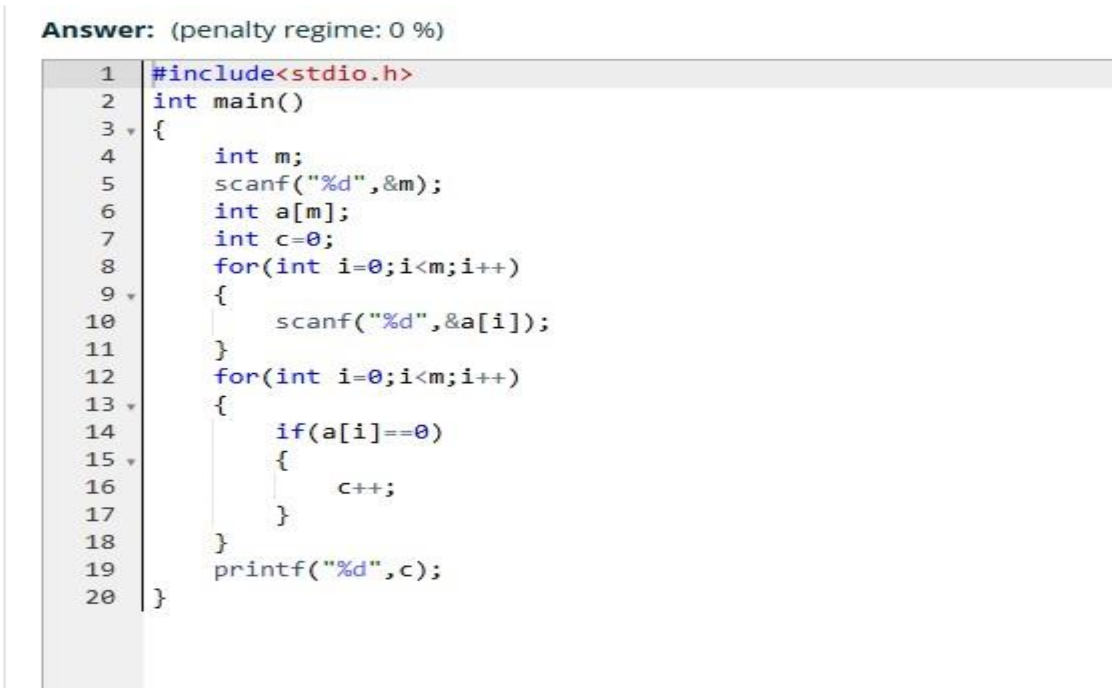
Input Format

First Line Contains Integer m – Size of array

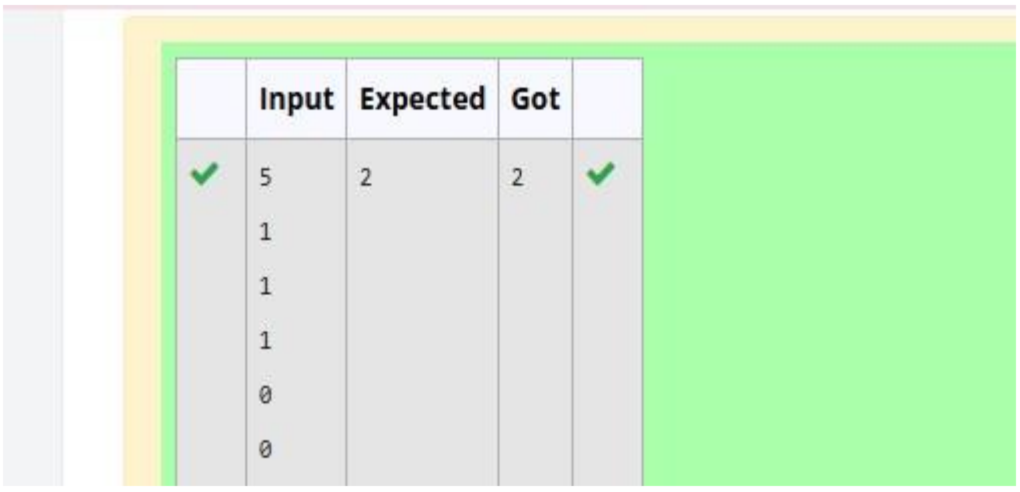
Next m lines Contains m numbers – Elements of an array

Output Format

First Line Contains Integer – Number of zeroes present in the given array.

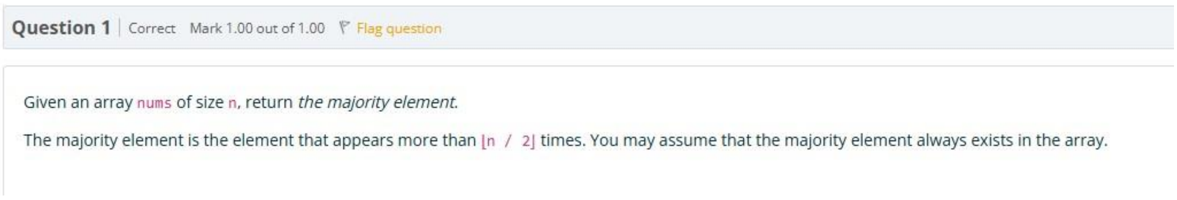

Answer: (penalty regime: 0 %)

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



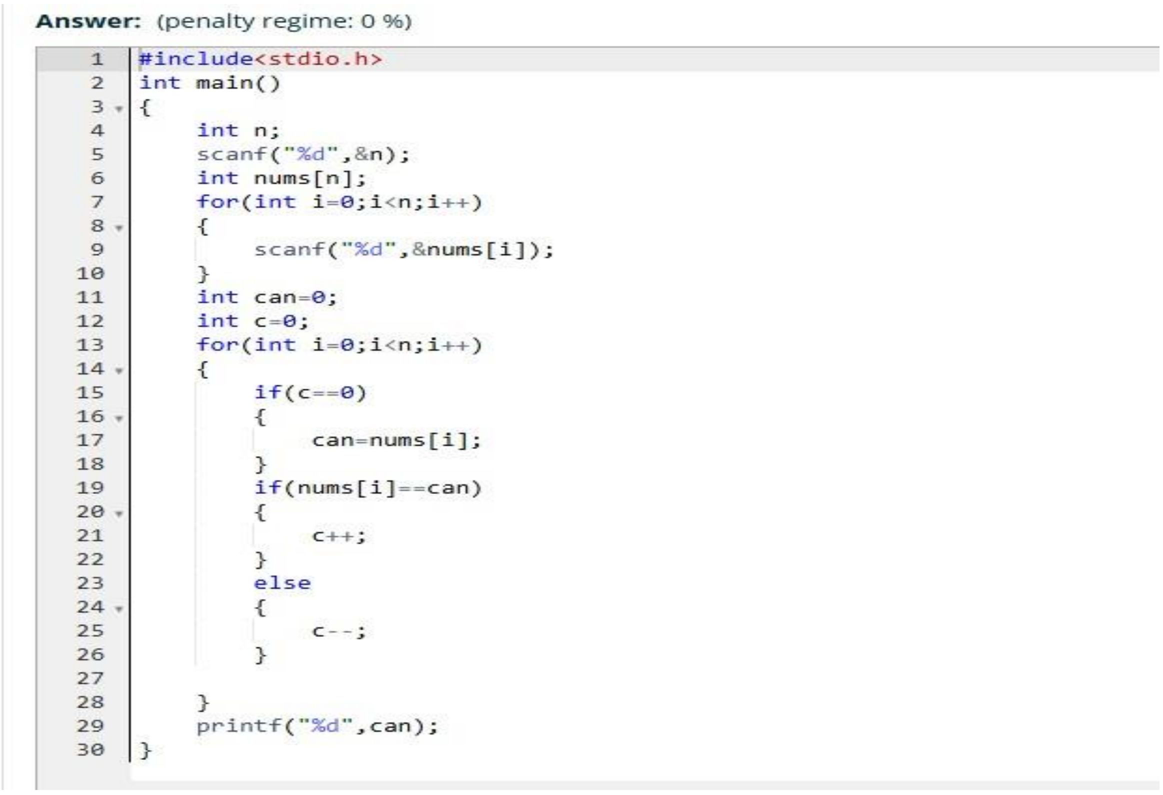
	Input	Expected	Got	
✓	5 1 1 1 0 0	2	2	✓

Q2:

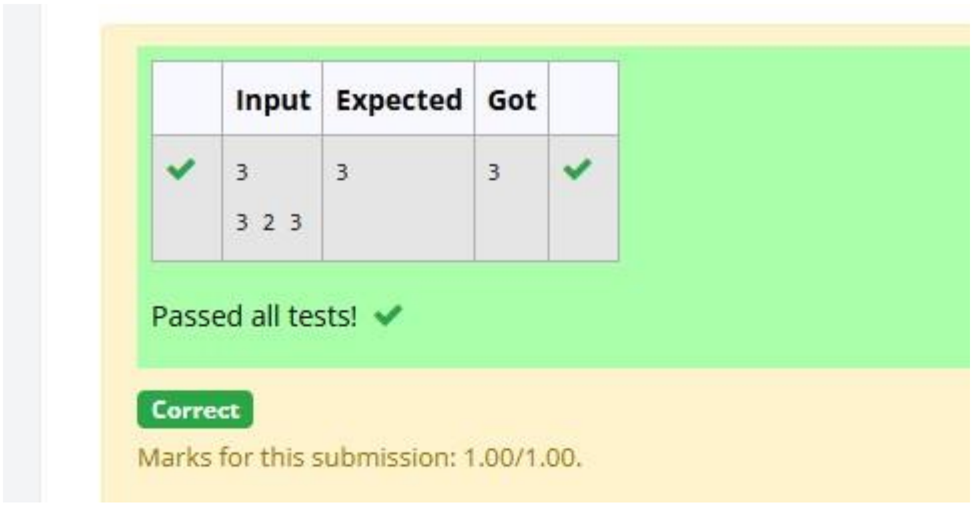

Question 1 | Correct Mark 1.00 out of 1.00 Flag question

Given an array nums of size n, return *the majority element.*

The majority element is the element that appears more than ⌊n / 2⌋ times. You may assume that the majority element always exists in the array.


Answer: (penalty regime: 0 %)

```
#include<stdio.h>
int main()
{
    int n;
    scanf("%d",&n);
    int nums[n];
    for(int i=0;i<n;i++)
    {
        scanf("%d",&nums[i]);
    }
    int can=0;
    int c=0;
    for(int i=0;i<n;i++)
    {
        if(c==0)
        {
            can=nums[i];
        }
        if(nums[i]==can)
        {
            c++;
        }
        else
        {
            c--;
        }

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



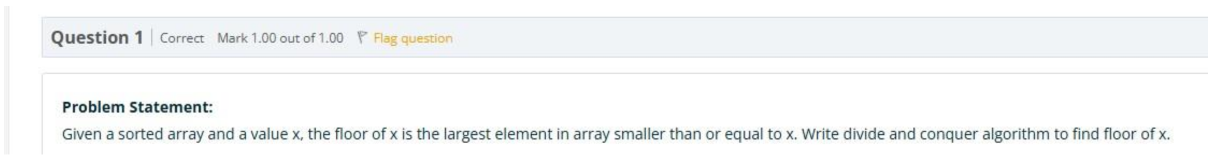
	Input	Expected	Got	
✓	3 3 2 3	3	3	✓

Passed all tests! ✓

Correct

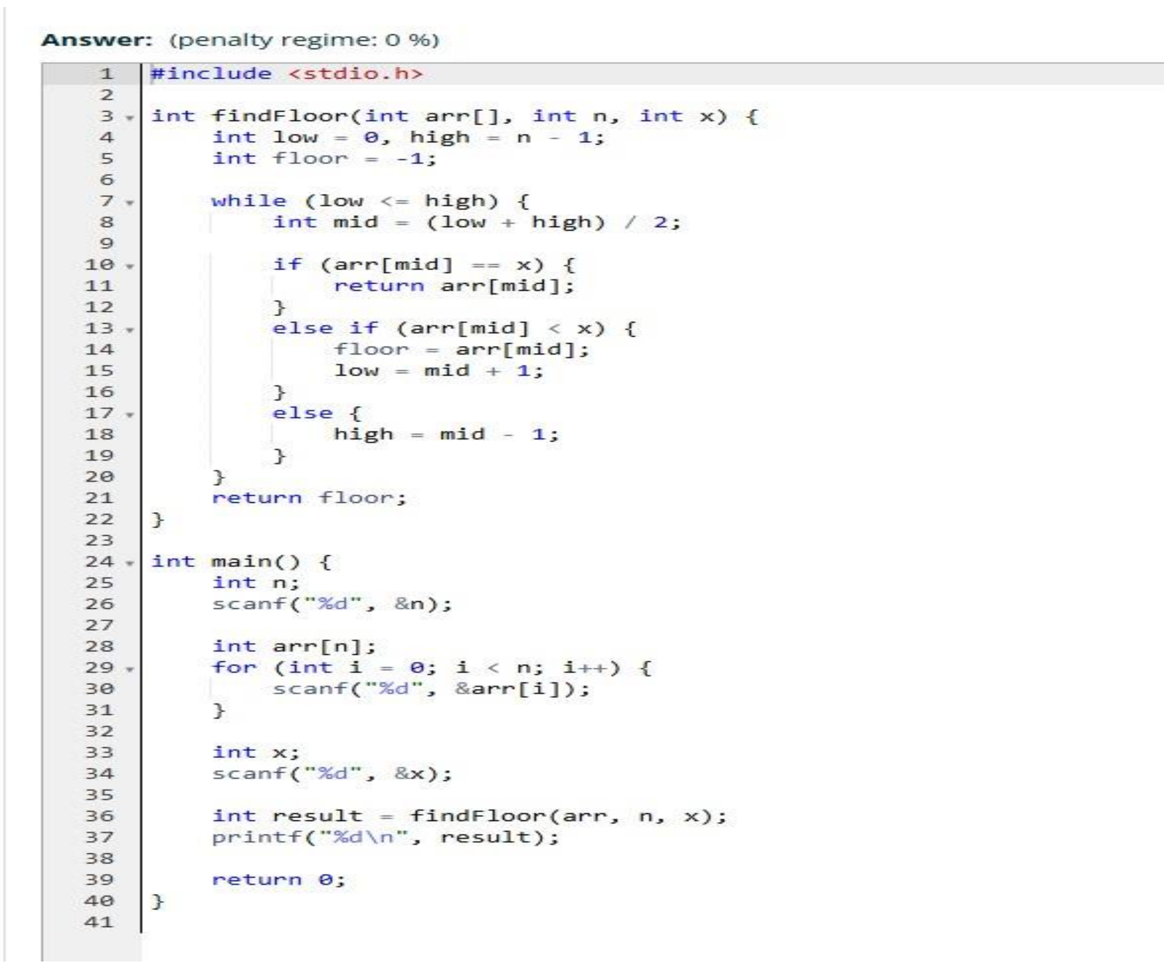
Marks for this submission: 1.00/1.00.

Q3:

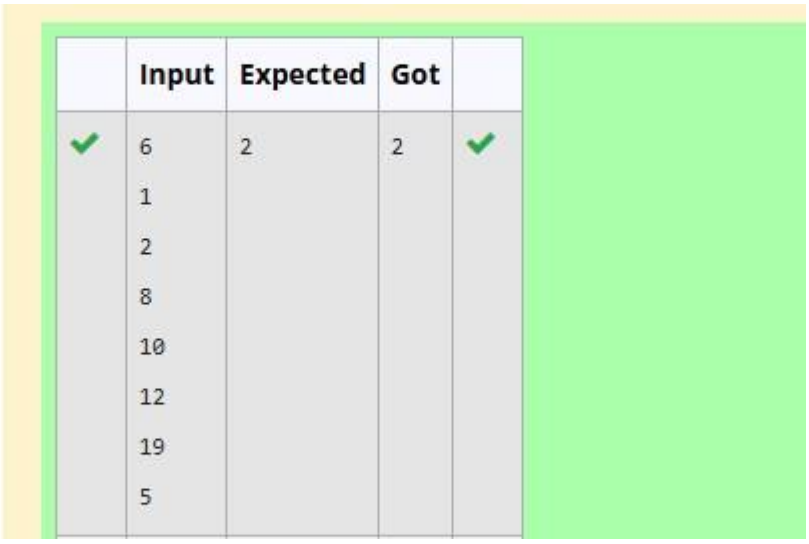

Question 1 | Correct Mark 1.00 out of 1.00 Flag question

Problem Statement:

Given a sorted array and a value x, the floor of x is the largest element in array smaller than or equal to x. Write divide and conquer algorithm to find floor of x.

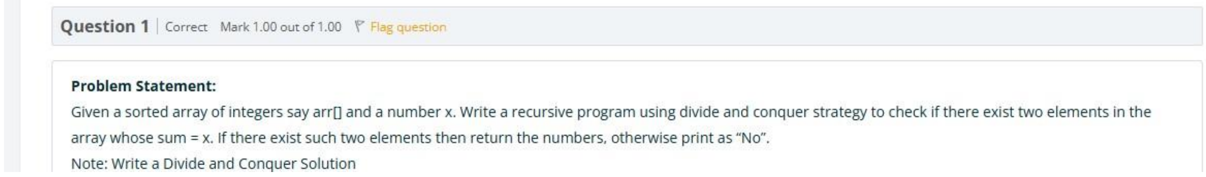

Answer: (penalty regime: 0 %)

```c
#include <stdio.h>

int findFloor(int arr[], int n, int x) {
    int low = 0, high = n - 1;
    int floor = -1;

    while (low <= high) {
        int mid = (low + high) / 2;

        if (arr[mid] == x) {
            return arr[mid];
        }
        else if (arr[mid] < x) {
            floor = arr[mid];
            low = mid + 1;
        }
        else {
            high = mid - 1;
        }
    }
    return floor;
}

int main() {
    int n;
    scanf("%d", &n);

    int arr[n];
    for (int i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }

    int x;
    scanf("%d", &x);

    int result = findFloor(arr, n, x);
    printf("%d\n", result);

    return 0;
}
```



	Input	Expected	Got	
✓	6	2	2	✓
	1			
	2			
	8			
	10			
	12			
	19			
	5			

Q4:

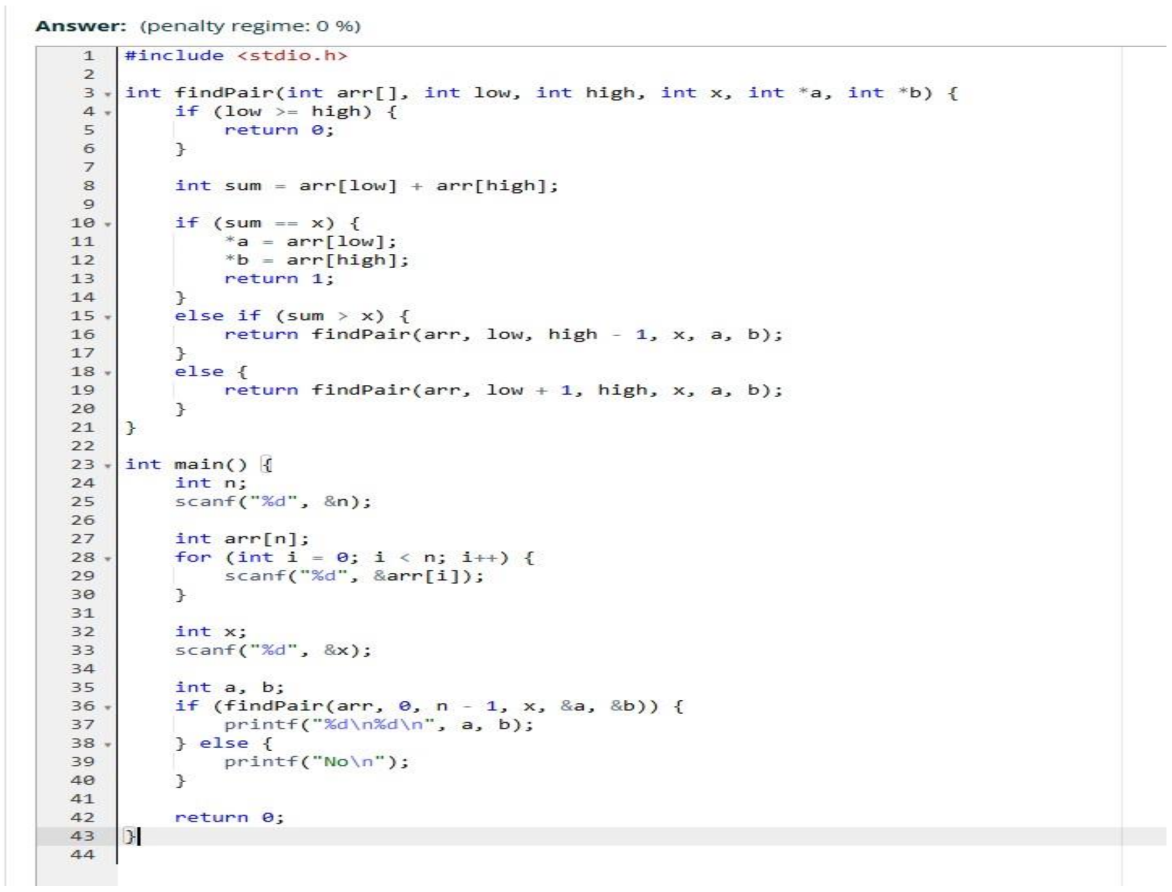


Question 1 | Correct Mark 1.00 out of 1.00 Flag question

Problem Statement:

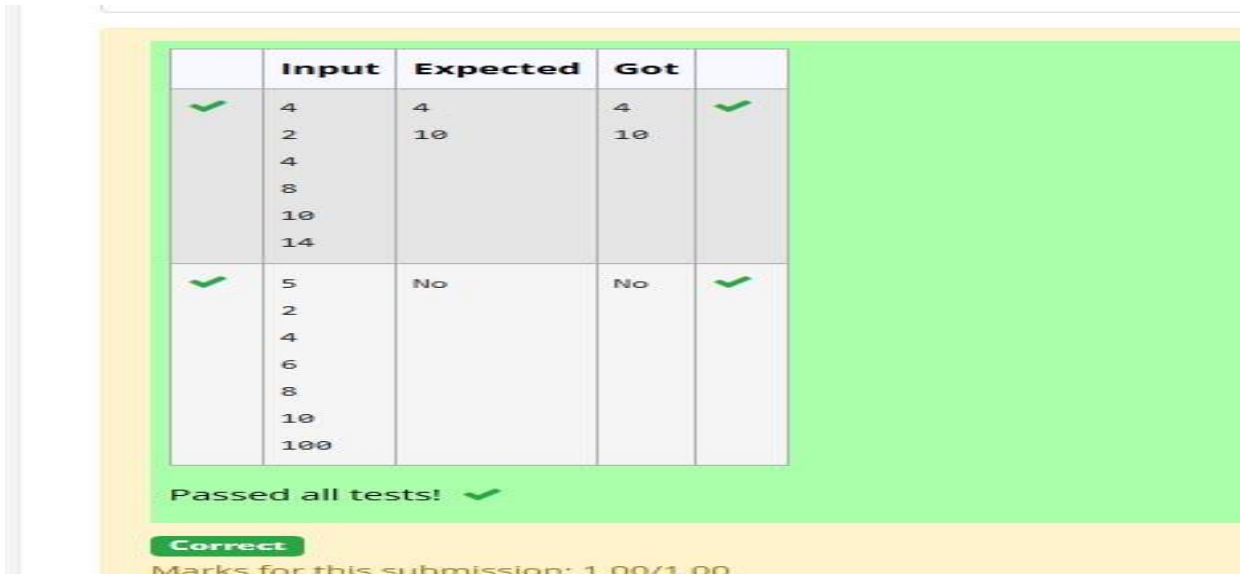
Given a sorted array of integers say arr[] and a number x. Write a recursive program using divide and conquer strategy to check if there exist two elements in the array whose sum = x. If there exist such two elements then return the numbers, otherwise print as "No".

Note: Write a Divide and Conquer Solution



Answer: (penalty regime: 0 %)

```c
#include <stdio.h>

int findPair(int arr[], int low, int high, int x, int *a, int *b) {
    if (low >= high) {
        return 0;
    }

    int sum = arr[low] + arr[high];

    if (sum == x) {
        *a = arr[low];
        *b = arr[high];
        return 1;
    }
    else if (sum > x) {
        return findPair(arr, low, high - 1, x, a, b);
    }
    else {
        return findPair(arr, low + 1, high, x, a, b);
    }
}

int main() {
    int n;
    scanf("%d", &n);

    int arr[n];
    for (int i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }

    int x;
    scanf("%d", &x);

    int a, b;
    if (findPair(arr, 0, n - 1, x, &a, &b)) {
        printf("%d\n%d\n", a, b);
    } else {
        printf("No\n");
    }

    return 0;
}

```



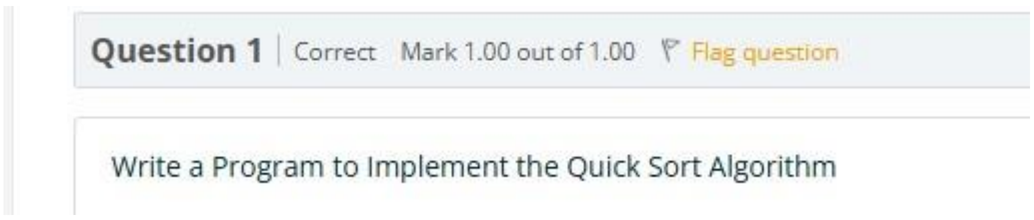
	Input	Expected	Got	
✓	4 2 4 8 10 14	4 10	4 10	✓
✓	5 2 4 6 8 10 100	No	No	✓

Passed all tests! ✓

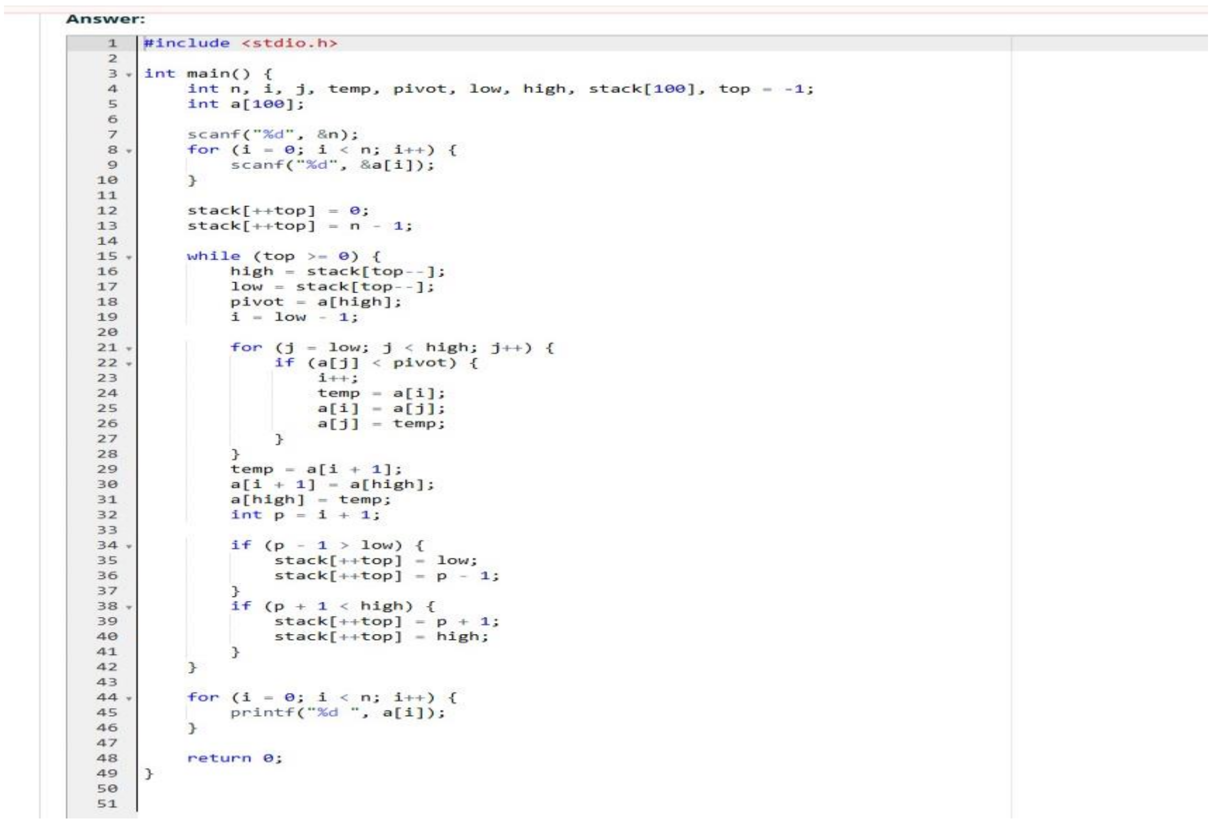
Correct

Marks for this submission: 1.00/1.00

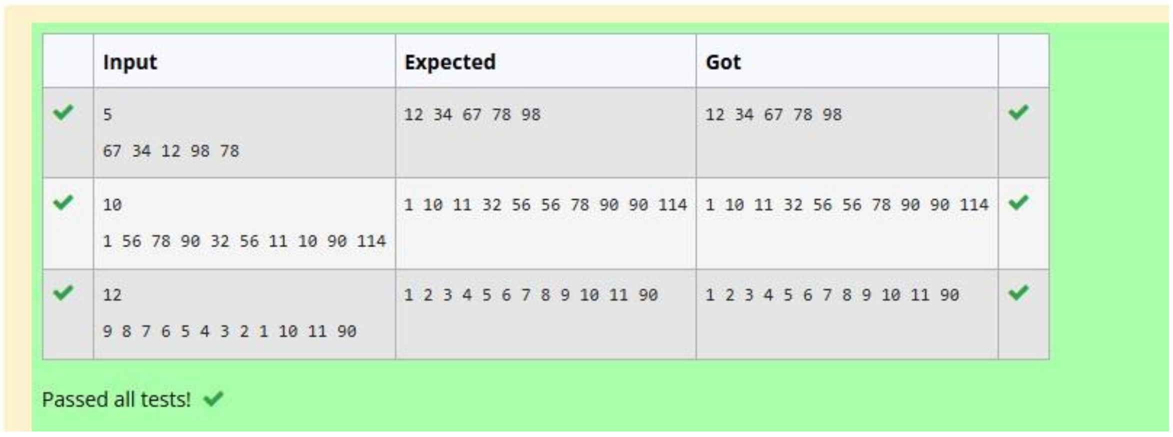
Q5:


Question 1 | Correct Mark 1.00 out of 1.00 Flag question

Write a Program to Implement the Quick Sort Algorithm


Answer:

```
#include <stdio.h>

int main() {
    int n, i, j, temp, pivot, low, high, stack[100], top = -1;
    int a[100];

    scanf("%d", &n);
    for (i = 0; i < n; i++) {
        scanf("%d", &a[i]);
    }

    stack[++top] = 0;
    stack[++top] = n - 1;

    while (top >= 0) {
        high = stack[top--];
        low = stack[top--];
        pivot = a[high];
        i = low - 1;

        for (j = low; j < high; j++) {
            if (a[j] < pivot) {
                i++;
                temp = a[i];
                a[i] = a[j];
                a[j] = temp;
            }
        }
        temp = a[i + 1];
        a[i + 1] = a[high];
        a[high] = temp;
        int p = i + 1;

        if (p - 1 > low) {
            stack[++top] = low;
            stack[++top] = p - 1;
        }
        if (p + 1 < high) {
            stack[++top] = p + 1;
            stack[++top] = high;
        }
    }

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

    return 0;
}
```

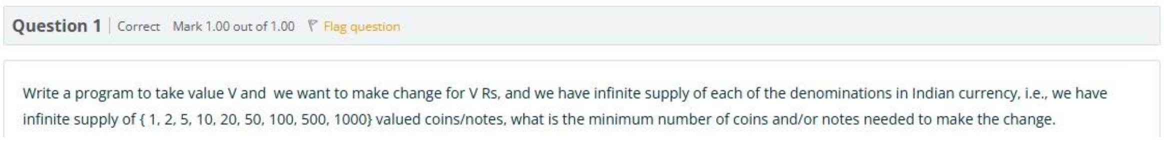


	Input	Expected	Got	
✓	5 67 34 12 98 78	12 34 67 78 98	12 34 67 78 98	✓
✓	10 1 56 78 90 32 56 11 10 90 114	1 10 11 32 56 56 78 90 90 114	1 10 11 32 56 56 78 90 90 114	✓
✓	12 9 8 7 6 5 4 3 2 1 10 11 90	1 2 3 4 5 6 7 8 9 10 11 90	1 2 3 4 5 6 7 8 9 10 11 90	✓

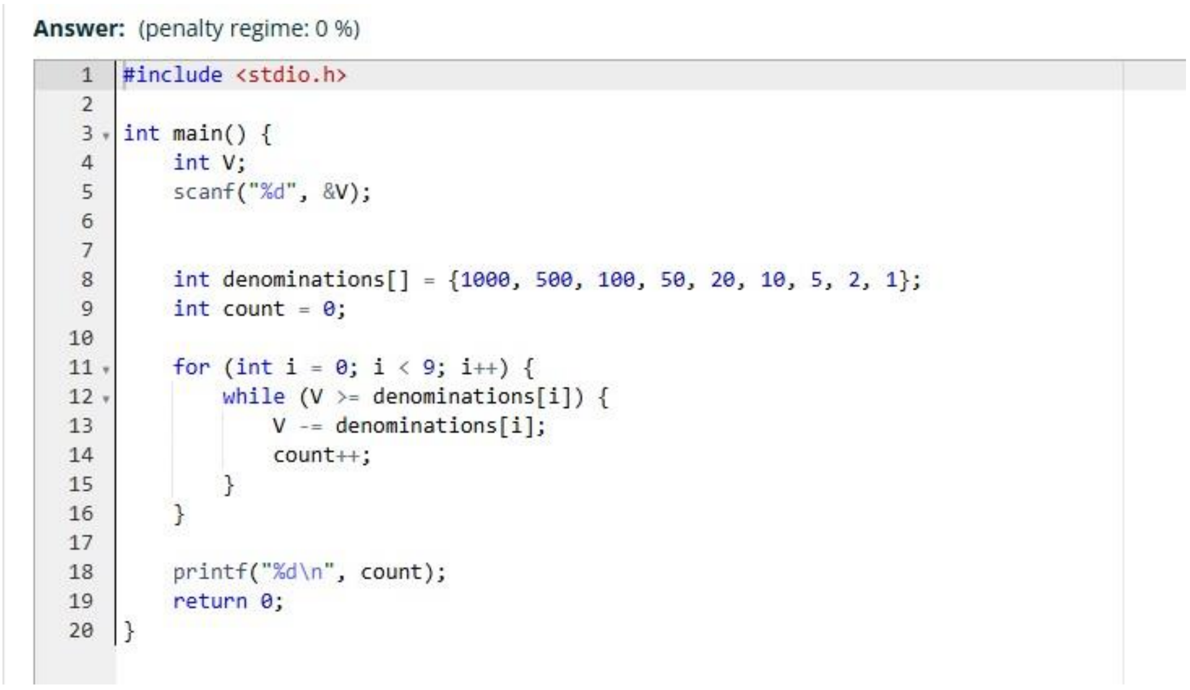
Passed all tests! ✓

4.Greedy Algorithms

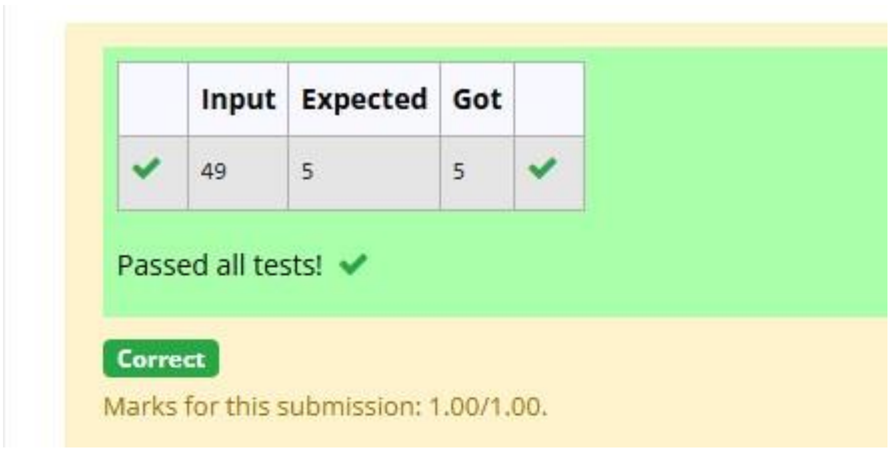
Q1:


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 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.


Answer: (penalty regime: 0 %)

```c
#include <stdio.h>

int main() {
    int V;
    scanf("%d", &V);


    int denominations[] = {1000, 500, 100, 50, 20, 10, 5, 2, 1};
    int count = 0;

    for (int i = 0; i < 9; i++) {
        while (V >= denominations[i]) {
            V -= denominations[i];
            count++;
        }
    }

    printf("%d\n", count);
    return 0;
}
```



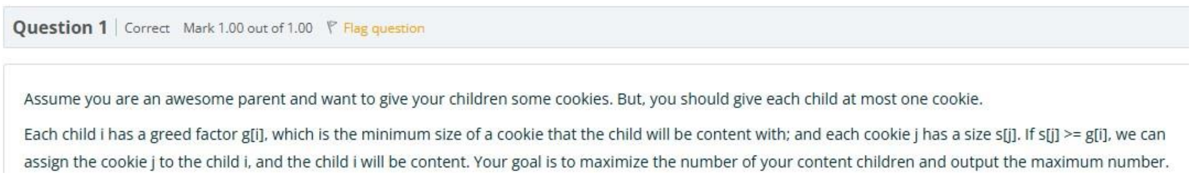
	Input	Expected	Got	
✓	49	5	5	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

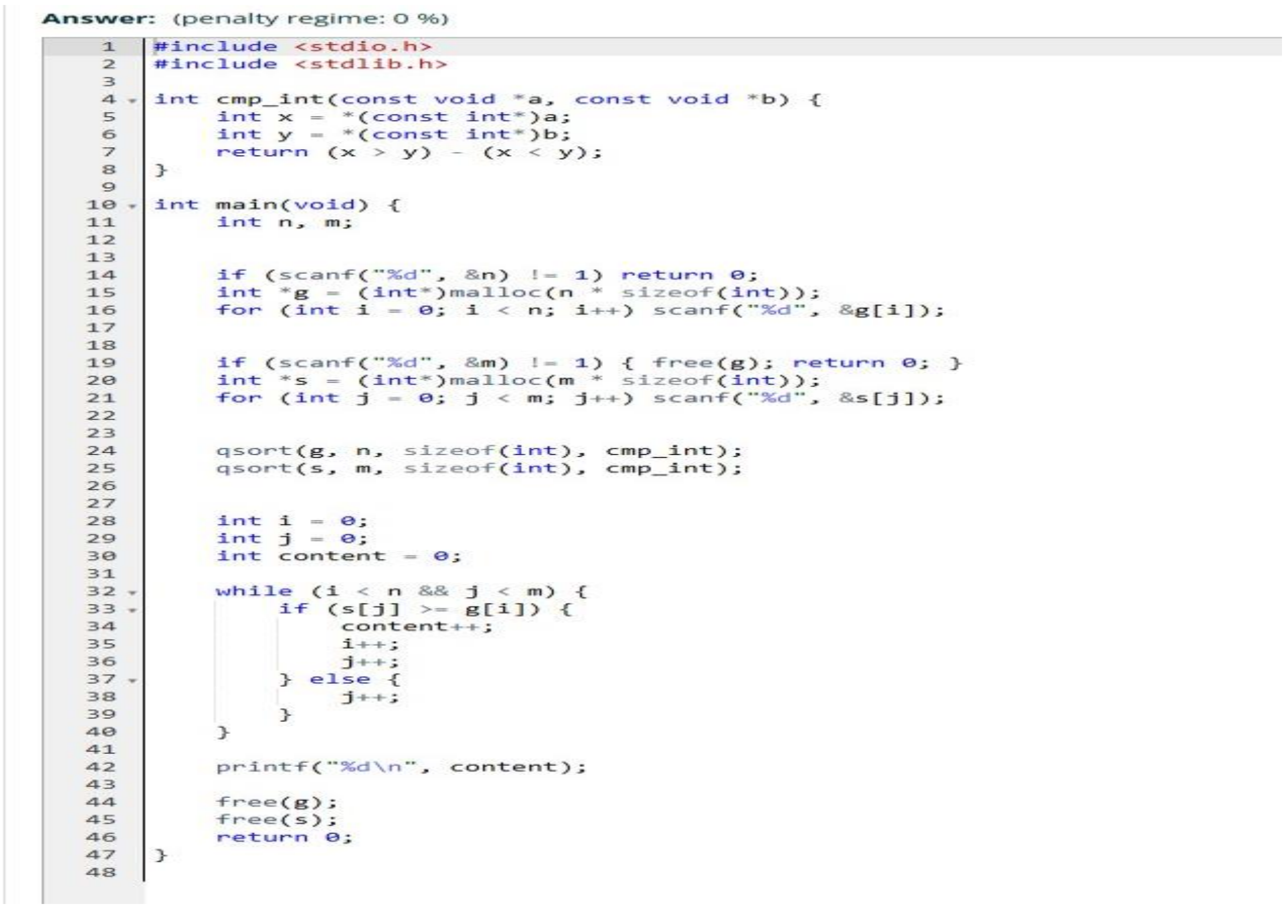
Q2:



Question 1 | Correct Mark 1.00 out of 1.00 Flag question

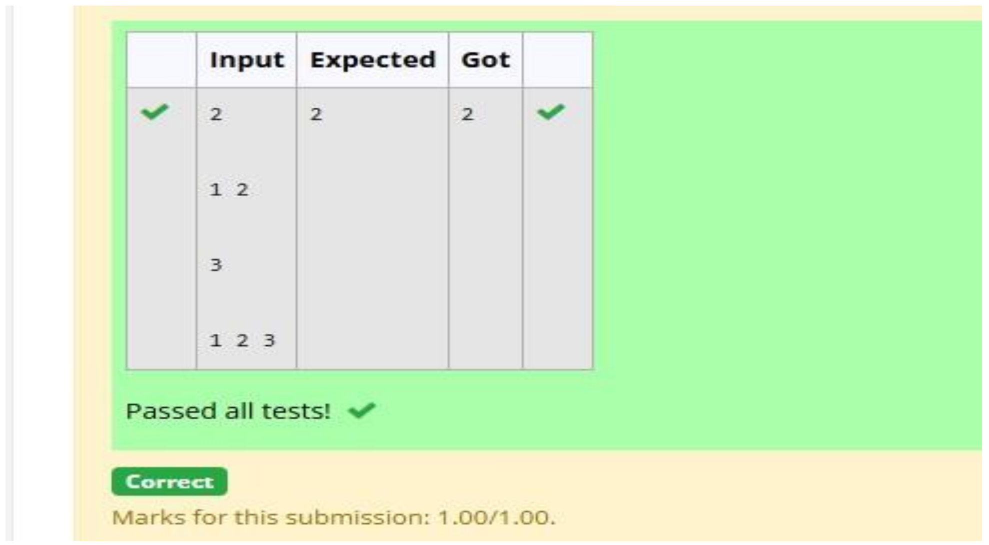
Assume you are an awesome parent and want to give your children some cookies. But, you should give each child at most one cookie.

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] >= 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.



Answer: (penalty regime: 0 %)

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

int cmp_int(const void *a, const void *b) {
    int x = *(const int*)a;
    int y = *(const int*)b;
    return (x > y) - (x < y);
}

int main(void) {
    int n, m;


    if (scanf("%d", &n) != 1) return 0;
    int *g = (int*)malloc(n * sizeof(int));
    for (int i = 0; i < n; i++) scanf("%d", &g[i]);


    if (scanf("%d", &m) != 1) { free(g); return 0; }
    int *s = (int*)malloc(m * sizeof(int));
    for (int j = 0; j < m; j++) scanf("%d", &s[j]);


    qsort(g, n, sizeof(int), cmp_int);
    qsort(s, m, sizeof(int), cmp_int);


    int i = 0;
    int j = 0;
    int content = 0;

    while (i < n && j < m) {
        if (s[j] >= g[i]) {
            content++;
            i++;
            j++;
        } else {
            j++;
        }
    }

    printf("%d\n", content);

    free(g);
    free(s);
    return 0;
}
```



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

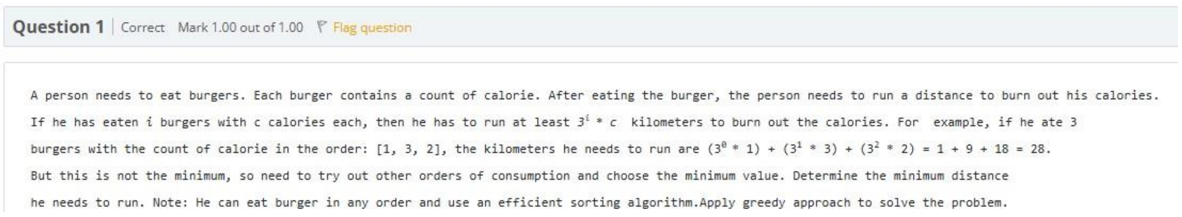
Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

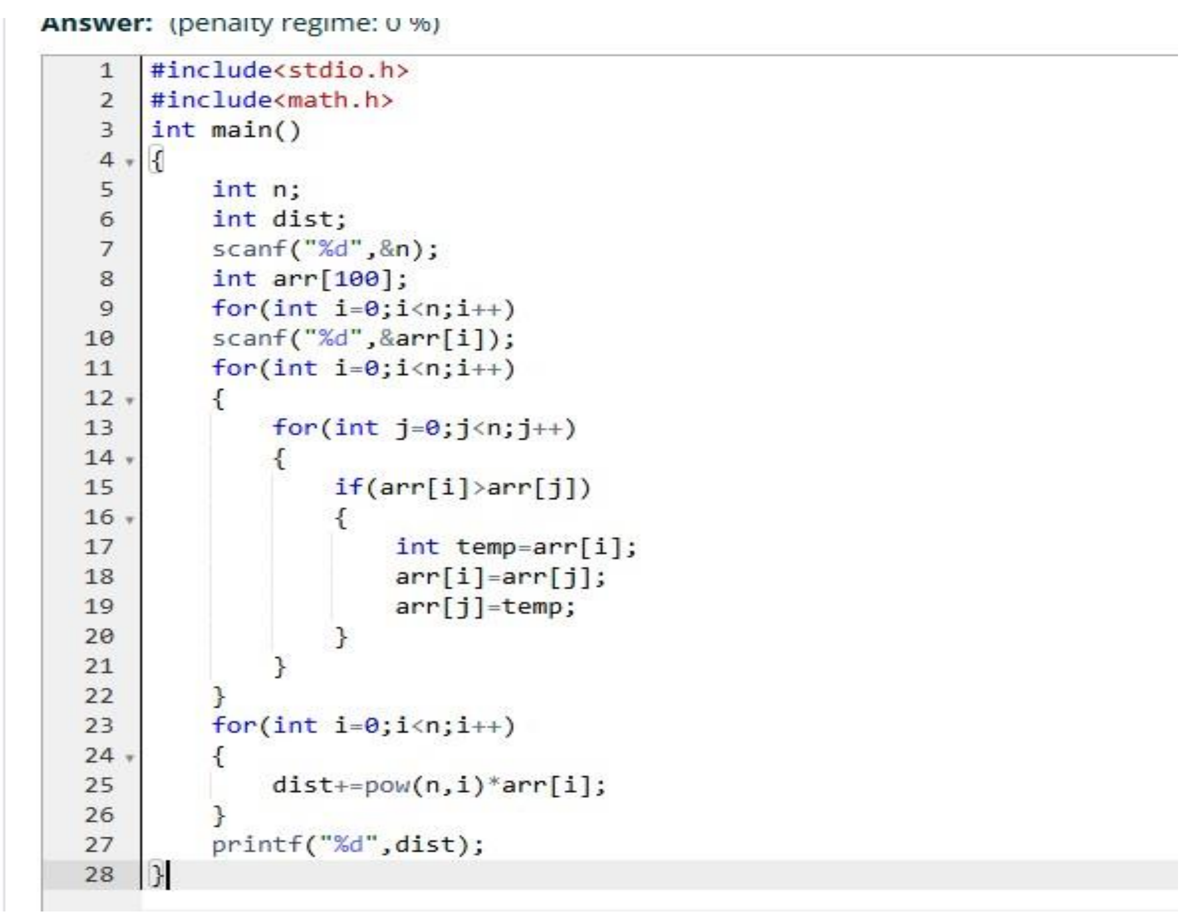
Q3:

Q4:



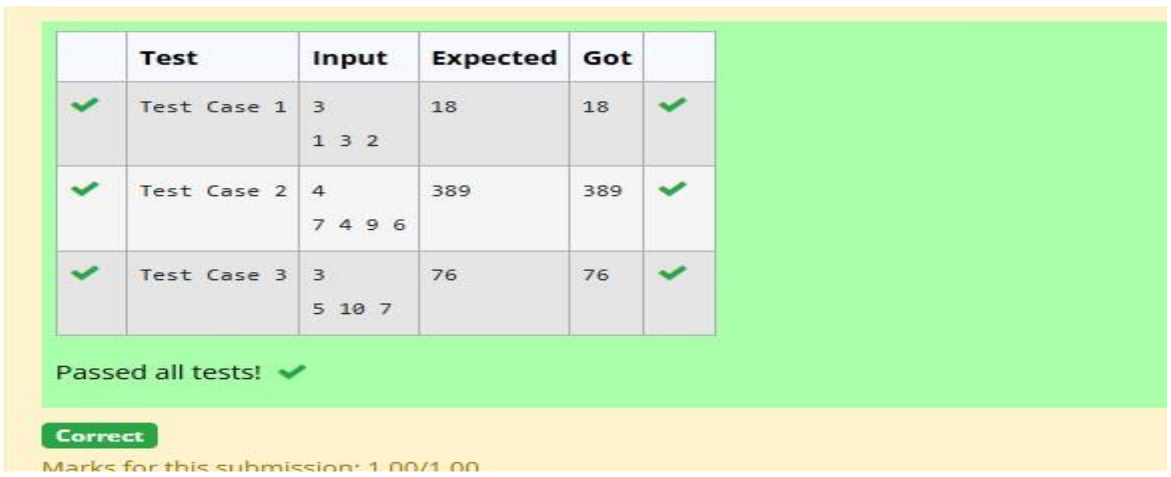
Question 1 | Correct Mark 1.00 out of 1.00 Flag question

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



Answer: (penalty regime: 0 %)

```c
#include<stdio.h>
#include<math.h>
int main()
{
    int n;
    int dist;
    scanf("%d",&n);
    int arr[100];
    for(int i=0;i<n;i++)
    scanf("%d",&arr[i]);
    for(int i=0;i<n;i++)
    {
        for(int j=0;j<n;j++)
        {
            if(arr[i]>arr[j])
            {
                int temp=arr[i];
                arr[i]=arr[j];
                arr[j]=temp;
            }
        }
    }
    for(int i=0;i<n;i++)
    {
        dist+=pow(n,i)*arr[i];
    }
    printf("%d",dist);
}
```

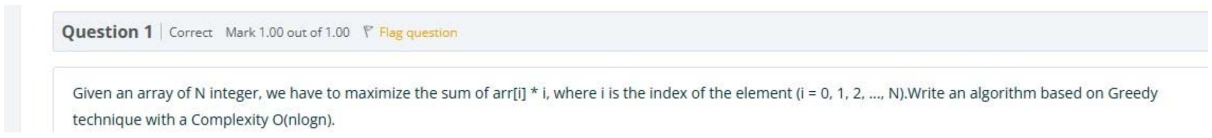


	Test	Input	Expected	Got	
✓	Test Case 1	3 1 3 2	18	18	✓
✓	Test Case 2	4 7 4 9 6	389	389	✓
✓	Test Case 3	3 5 10 7	76	76	✓

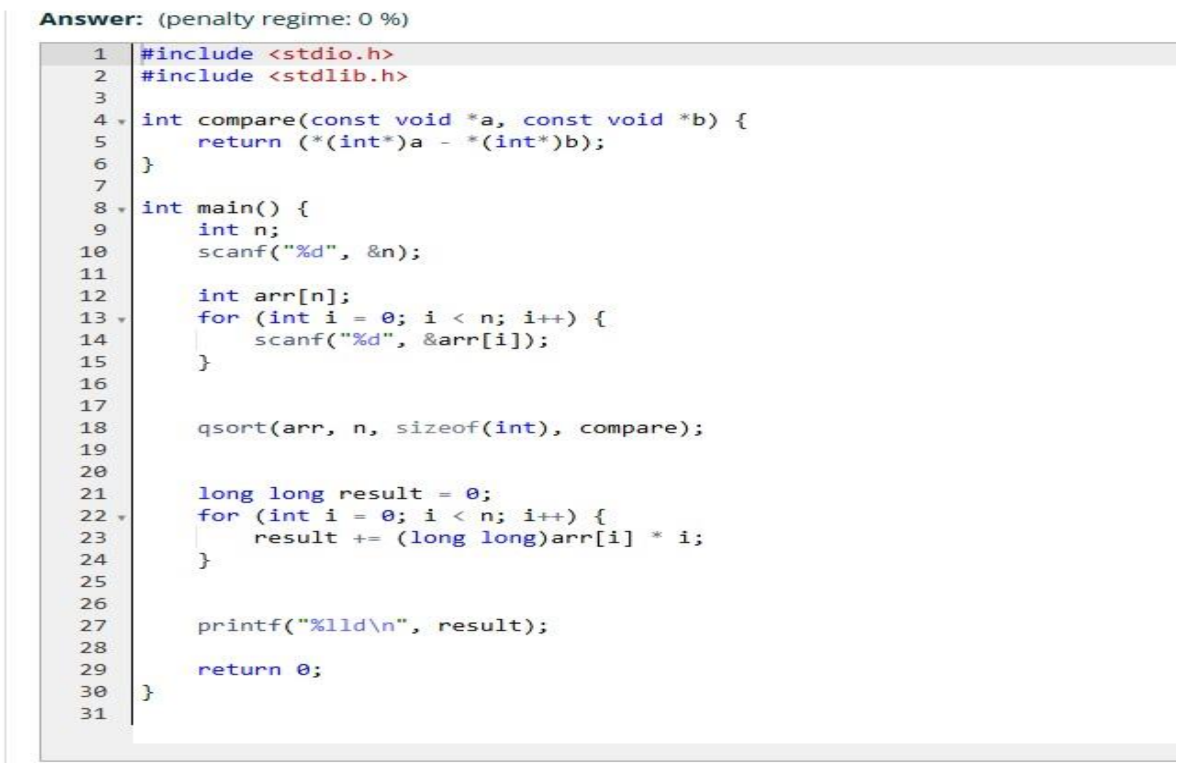
Passed all tests! ✓

Correct

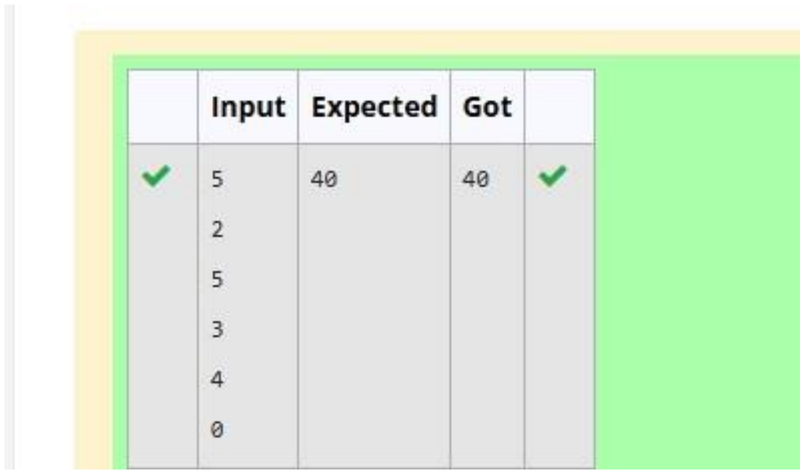
Marks for this submission: 1.00/1.00


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 based on Greedy technique with a Complexity O(nlogn).

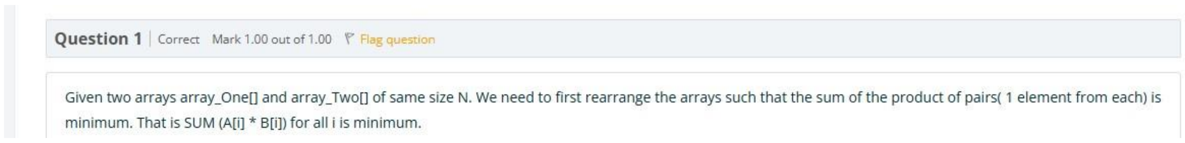

Answer: (penalty regime: 0 %)

```c
#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[n];
    for (int i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }


    qsort(arr, n, sizeof(int), compare);


    long long result = 0;
    for (int i = 0; i < n; i++) {
        result += (long long)arr[i] * i;
    }


    printf("%lld\n", result);

    return 0;
}

```



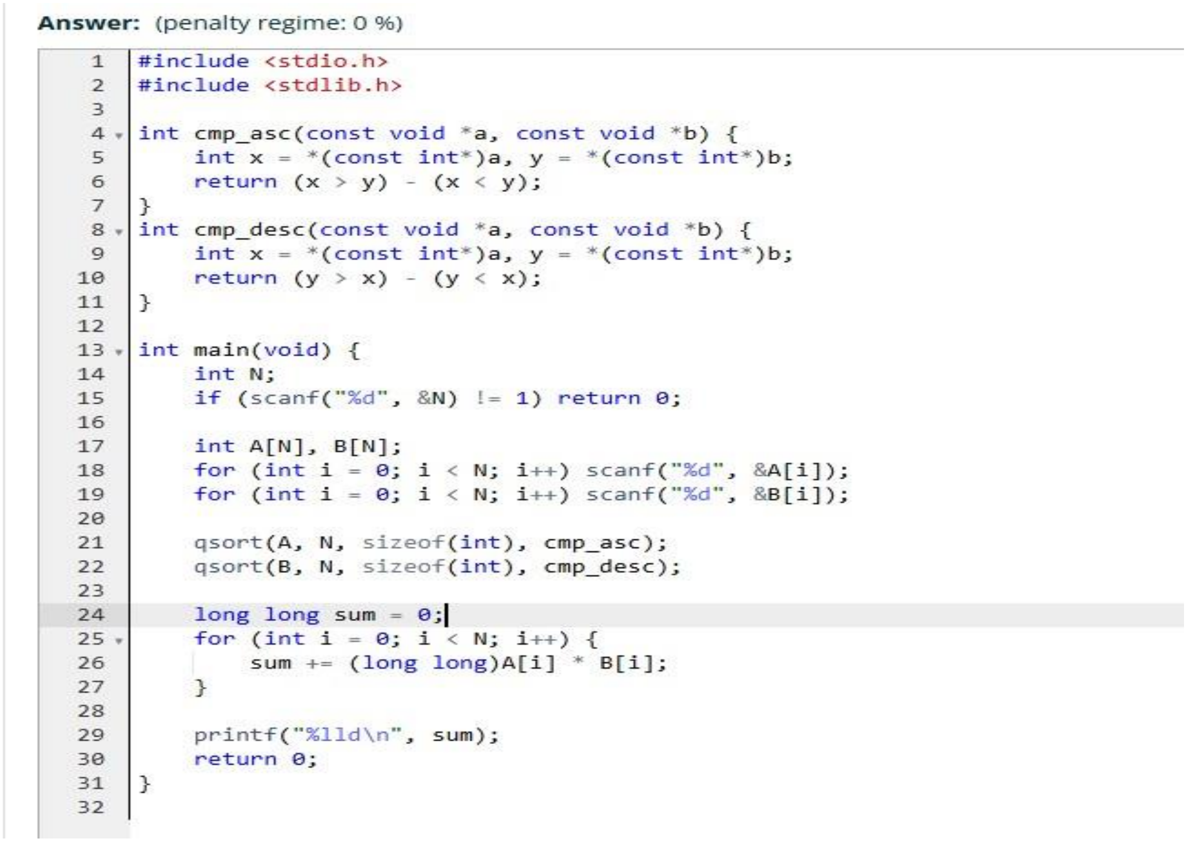
	Input	Expected	Got	
✓	5 2 5 3 4 0	40	40	✓

Q5:



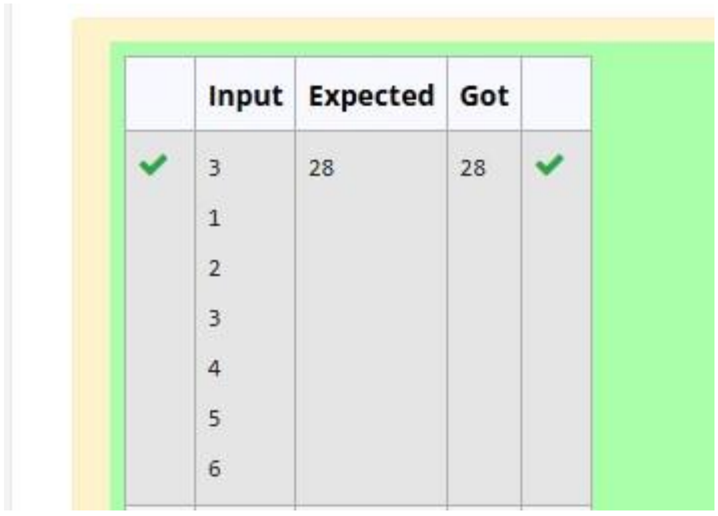
Question 1 | Correct Mark 1.00 out of 1.00 Flag question

Given two arrays array_One[] and array_Two[] of same size N. We need to first rearrange the arrays such that the sum of the product of pairs(1 element from each) is minimum. That is SUM (A[i] * B[i]) for all i is minimum.



Answer: (penalty regime: 0 %)

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

int cmp_asc(const void *a, const void *b) {
    int x = *(const int*)a, y = *(const int*)b;
    return (x > y) - (x < y);
}
int cmp_desc(const void *a, const void *b) {
    int x = *(const int*)a, y = *(const int*)b;
    return (y > x) - (y < x);
}

int main(void) {
    int N;
    if (scanf("%d", &N) != 1) return 0;

    int A[N], B[N];
    for (int i = 0; i < N; i++) scanf("%d", &A[i]);
    for (int i = 0; i < N; i++) scanf("%d", &B[i]);

    qsort(A, N, sizeof(int), cmp_asc);
    qsort(B, N, sizeof(int), cmp_desc);

    long long sum = 0;
    for (int i = 0; i < N; i++) {
        sum += (long long)A[i] * B[i];
    }

    printf("%lld\n", sum);
    return 0;
}

```

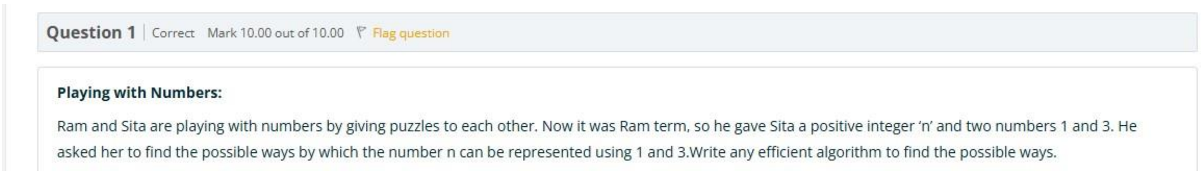


	Input	Expected	Got	
✓	3 1 2 3 4 5 6	28	28	✓

5.Dynamic Programming

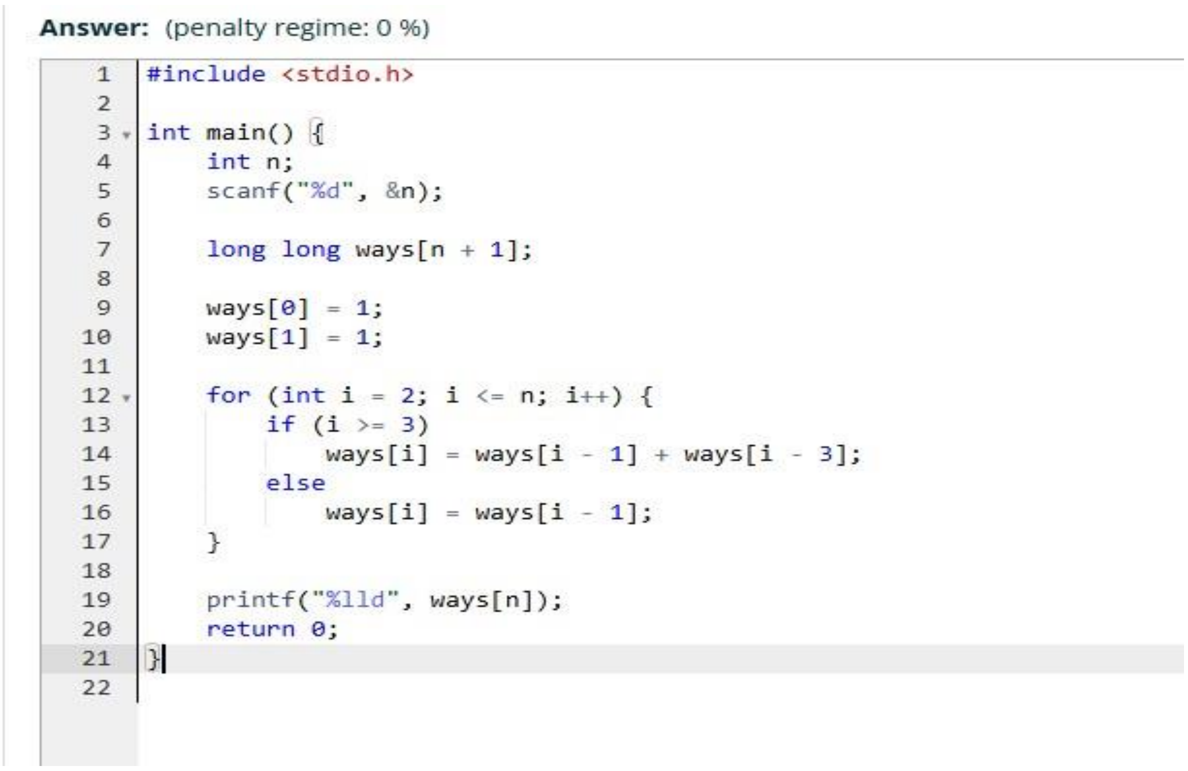
Q1:

Q2:

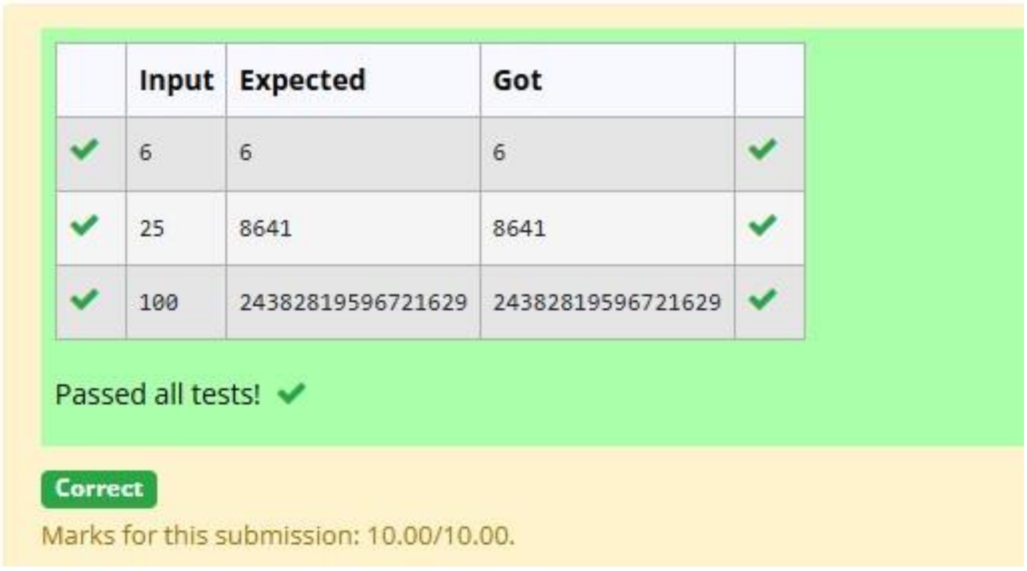

Question 1 | Correct Mark 10.00 out of 10.00 Flag question

Playing with Numbers:

Ram and Sita are playing with numbers by giving puzzles to each other. Now it was Ram term, so he gave Sita a positive integer 'n' and two numbers 1 and 3. He asked her to find the possible ways by which the number n can be represented using 1 and 3.Write any efficient algorithm to find the possible ways.


Answer: (penalty regime: 0 %)

```c
#include <stdio.h>

int main() {
    int n;
    scanf("%d", &n);

    long long ways[n + 1];

    ways[0] = 1;
    ways[1] = 1;

    for (int i = 2; i <= n; i++) {
        if (i >= 3)
            ways[i] = ways[i - 1] + ways[i - 3];
        else
            ways[i] = ways[i - 1];
    }

    printf("%lld", ways[n]);
    return 0;
}
```

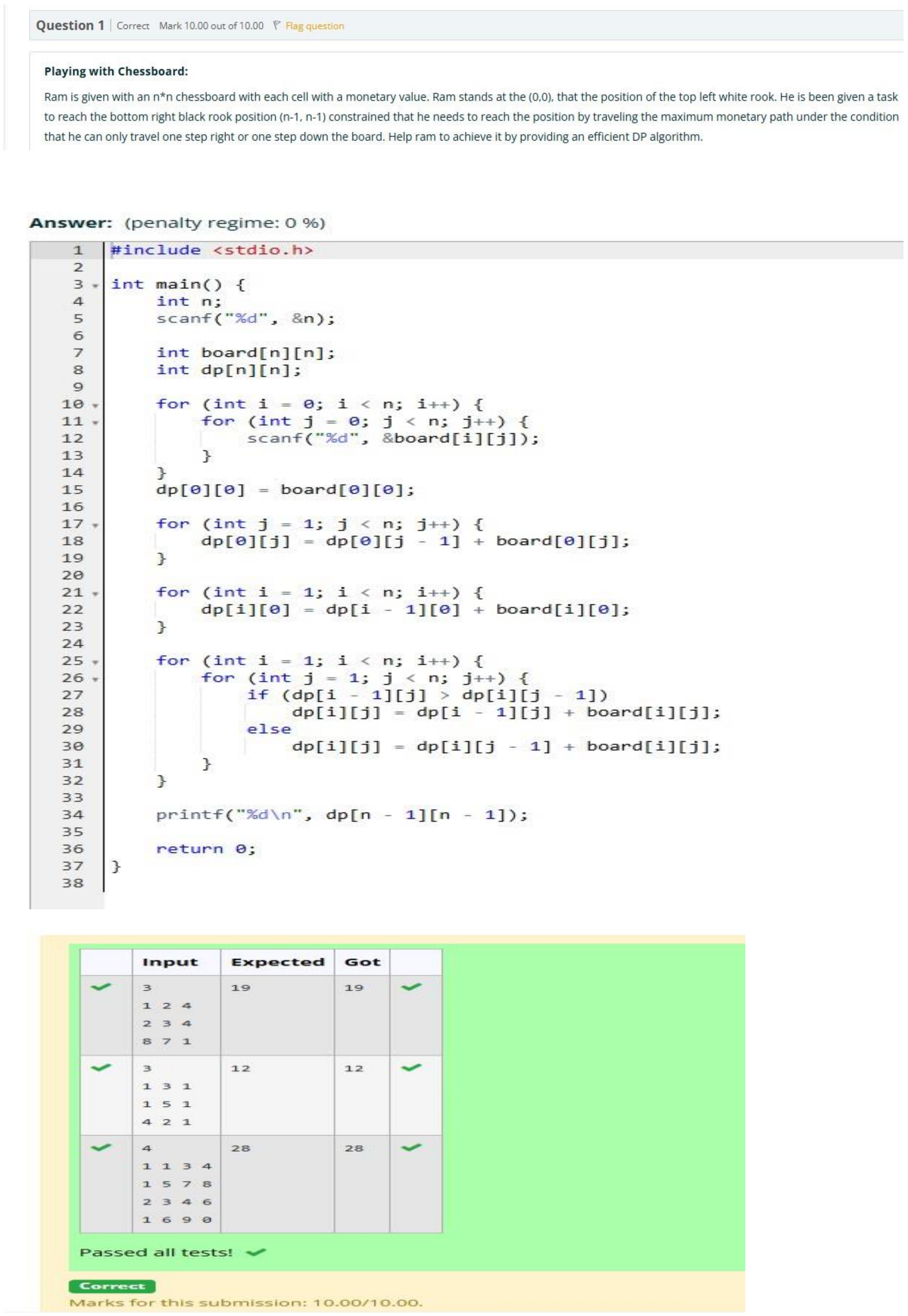


	Input	Expected	Got	
✓	6	6	6	✓
✓	25	8641	8641	✓
✓	100	24382819596721629	24382819596721629	✓

Passed all tests! ✓

Correct

Marks for this submission: 10.00/10.00.



Question 1 Correct Mark 10.00 out of 10.00 Flag question

Playing with Chessboard:

Ram is given with an n*n chessboard with each cell with a monetary value. Ram stands at the (0,0), that the position of the top left white rook. He is been given a task to reach the bottom right black rook position (n-1, n-1) constrained that he needs to reach the position by traveling the maximum monetary path under the condition that he can only travel one step right or one step down the board. Help ram to achieve it by providing an efficient DP algorithm.

Answer: (penalty regime: 0 %)

```c
#include <stdio.h>

int main() {
    int n;
    scanf("%d", &n);

    int board[n][n];
    int dp[n][n];

    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++) {
            scanf("%d", &board[i][j]);
        }
    }
    dp[0][0] = board[0][0];

    for (int j = 1; j < n; j++) {
        dp[0][j] = dp[0][j - 1] + board[0][j];
    }

    for (int i = 1; i < n; i++) {
        dp[i][0] = dp[i - 1][0] + board[i][0];
    }

    for (int i = 1; i < n; i++) {
        for (int j = 1; j < n; j++) {
            if (dp[i - 1][j] > dp[i][j - 1])
                dp[i][j] = dp[i - 1][j] + board[i][j];
            else
                dp[i][j] = dp[i][j - 1] + board[i][j];
        }
    }

    printf("%d\n", dp[n - 1][n - 1]);

    return 0;
}
```

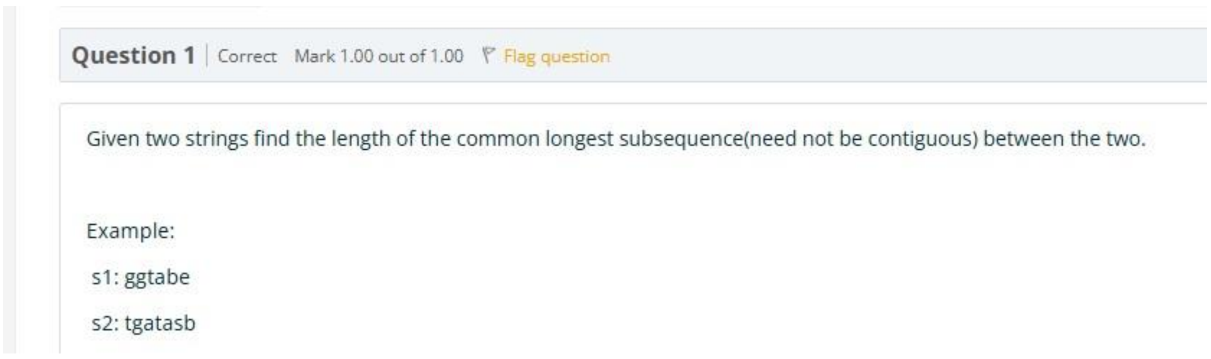
	Input	Expected	Got	
✓	3 1 2 4 2 3 4 8 7 1	19	19	✓
✓	3 1 3 1 1 5 1 4 2 1	12	12	✓
✓	4 1 1 3 4 1 5 7 8 2 3 4 6 1 6 9 0	28	28	✓

Passed all tests! ✓

Correct

Marks for this submission: 10.00/10.00.

Q3:

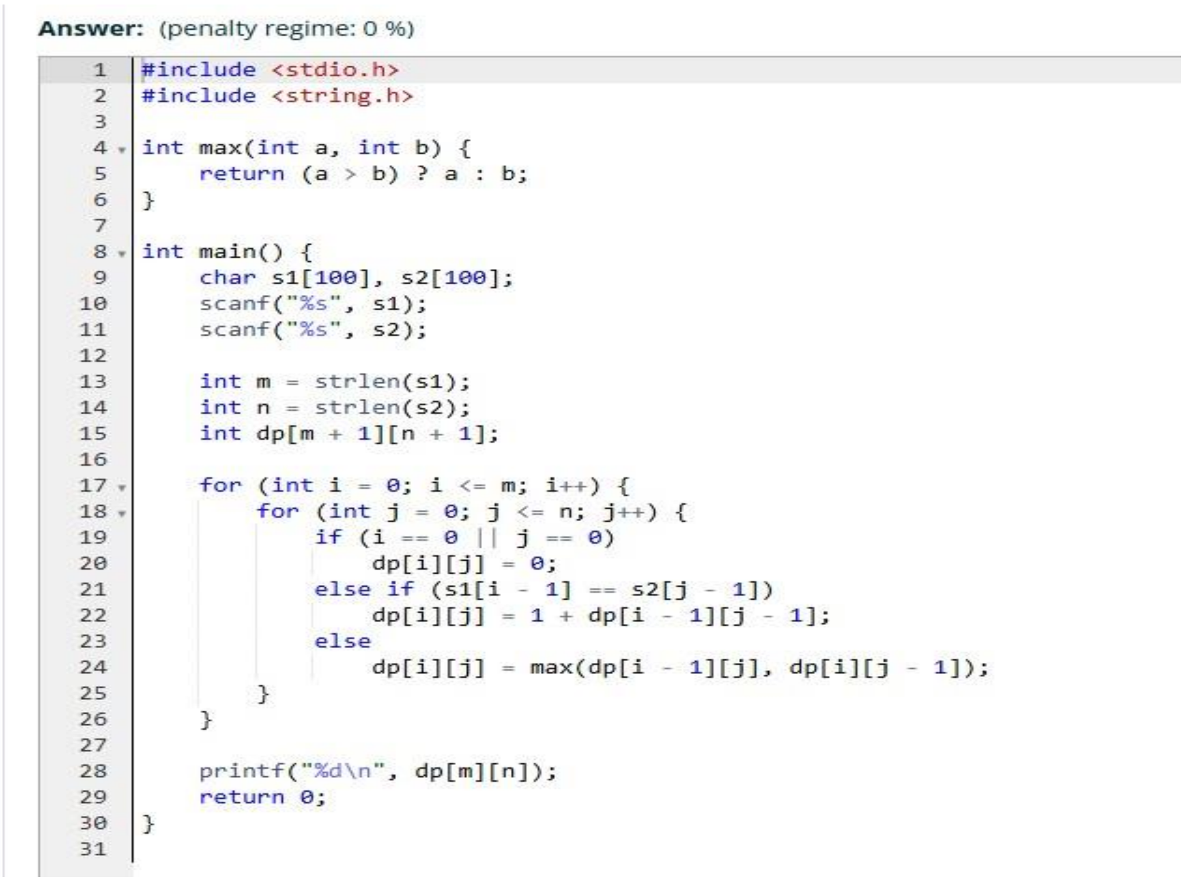

Question 1 | Correct Mark 1.00 out of 1.00 Flag question

Given two strings find the length of the common longest subsequence(need not be contiguous) between the two.

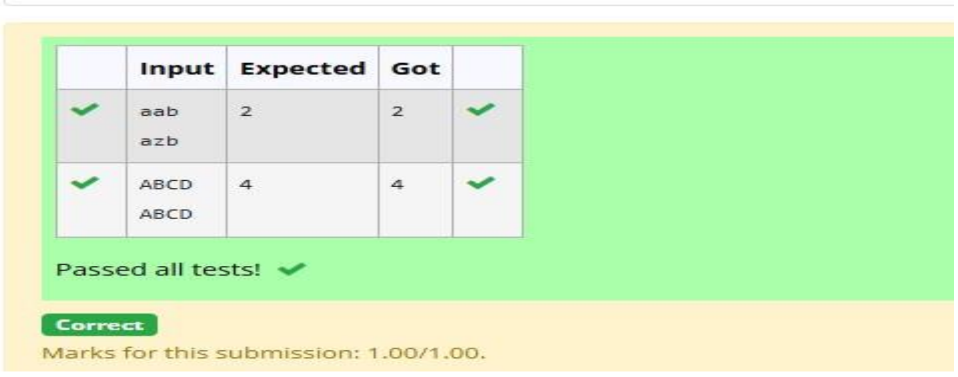
Example:

s1: ggtabe

s2: tgatasb


Answer: (penalty regime: 0 %)

```c
#include <stdio.h>
#include <string.h>

int max(int a, int b) {
    return (a > b) ? a : b;
}

int main() {
    char s1[100], s2[100];
    scanf("%s", s1);
    scanf("%s", s2);

    int m = strlen(s1);
    int n = strlen(s2);
    int dp[m + 1][n + 1];

    for (int i = 0; i <= m; i++) {
        for (int j = 0; j <= n; j++) {
            if (i == 0 || j == 0)
                dp[i][j] = 0;
            else if (s1[i - 1] == s2[j - 1])
                dp[i][j] = 1 + dp[i - 1][j - 1];
            else
                dp[i][j] = max(dp[i - 1][j], dp[i][j - 1]);
        }
    }

    printf("%d\n", dp[m][n]);
    return 0;
}
```

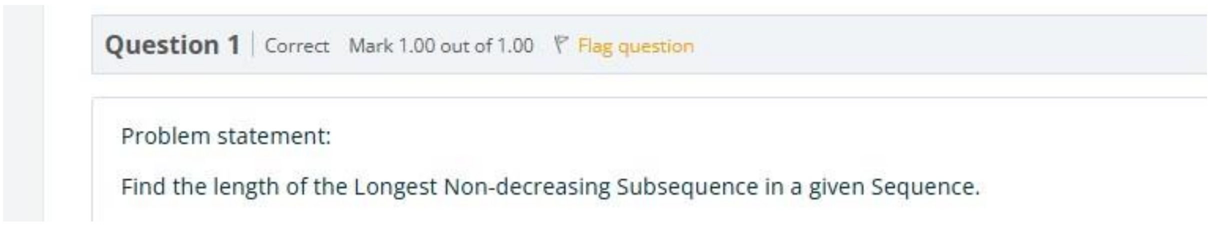

	Input	Expected	Got	
✓	aab azb	2	2	✓
✓	ABCD ABCD	4	4	✓

Passed all tests! ✓

Correct

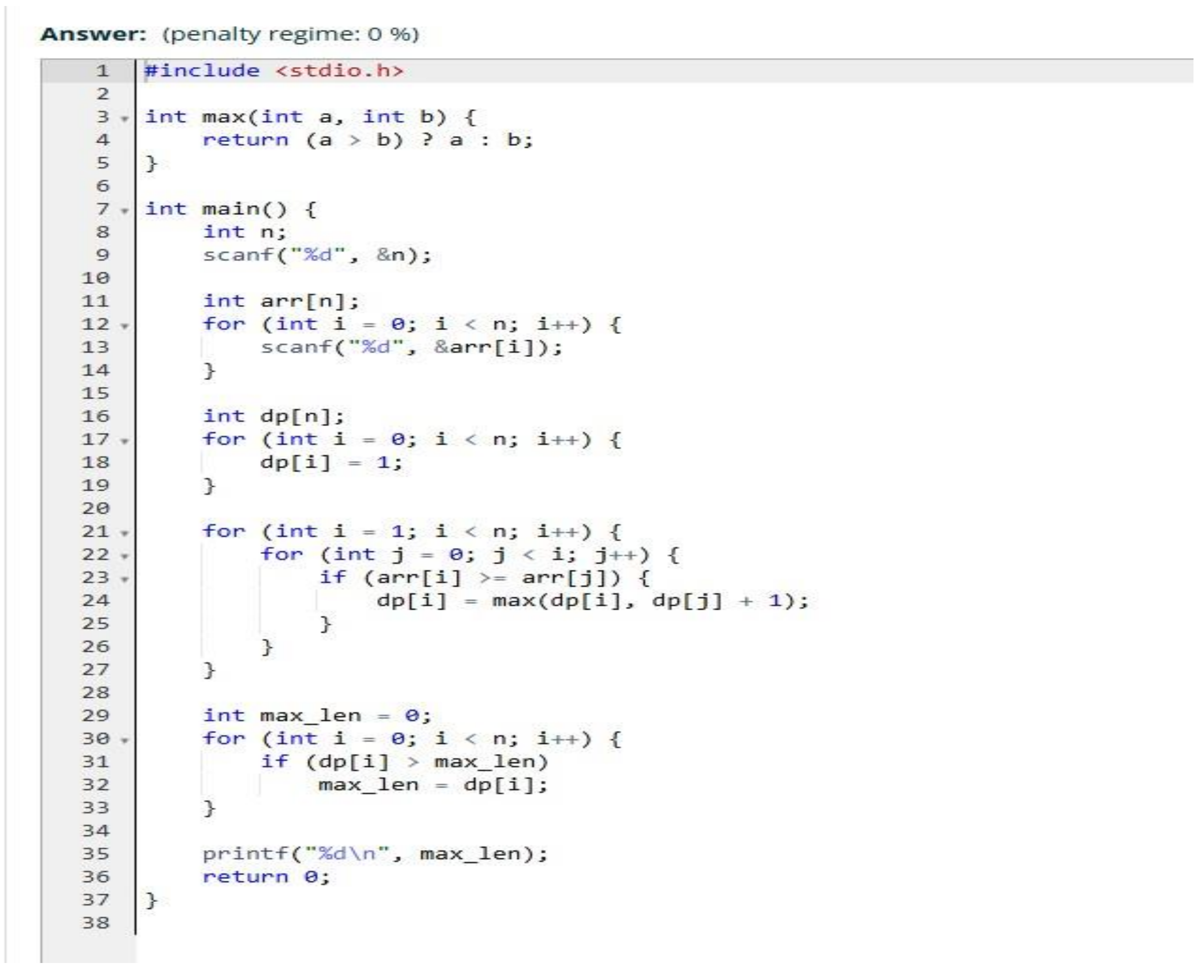
Marks for this submission: 1.00/1.00.

Q4:

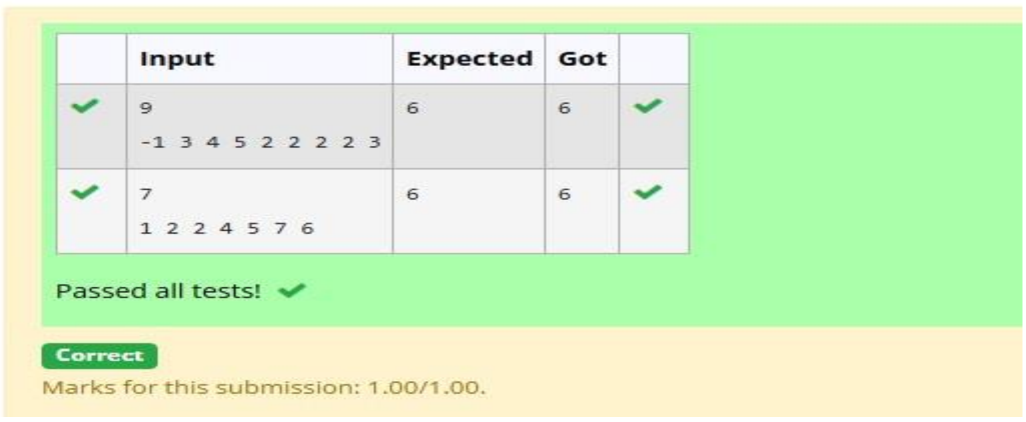

Question 1 | Correct Mark 1.00 out of 1.00 Flag question

Problem statement:

Find the length of the Longest Non-decreasing Subsequence in a given Sequence.


Answer: (penalty regime: 0 %)

```c
#include <stdio.h>

int max(int a, int b) {
    return (a > b) ? a : b;
}

int main() {
    int n;
    scanf("%d", &n);

    int arr[n];
    for (int i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }

    int dp[n];
    for (int i = 0; i < n; i++) {
        dp[i] = 1;
    }

    for (int i = 1; i < n; i++) {
        for (int j = 0; j < i; j++) {
            if (arr[i] >= arr[j]) {
                dp[i] = max(dp[i], dp[j] + 1);
            }
        }
    }

    int max_len = 0;
    for (int i = 0; i < n; i++) {
        if (dp[i] > max_len)
            max_len = dp[i];
    }

    printf("%d\n", max_len);
    return 0;
}
```



	Input	Expected	Got	
✓	9 -1 3 4 5 2 2 2 2 3	6	6	✓
✓	7 1 2 2 4 5 7 6	6	6	✓

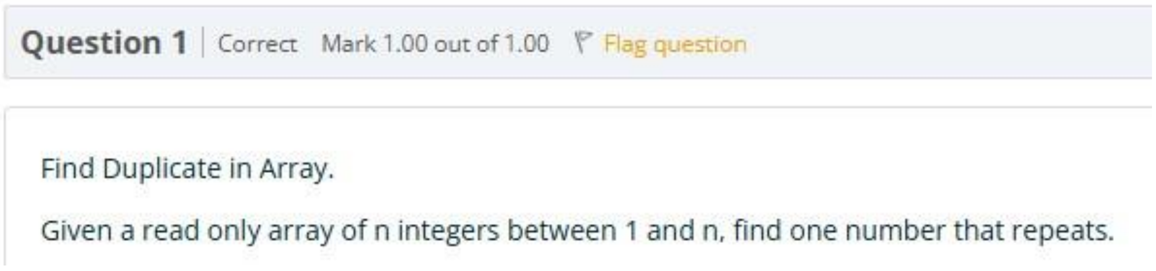
Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

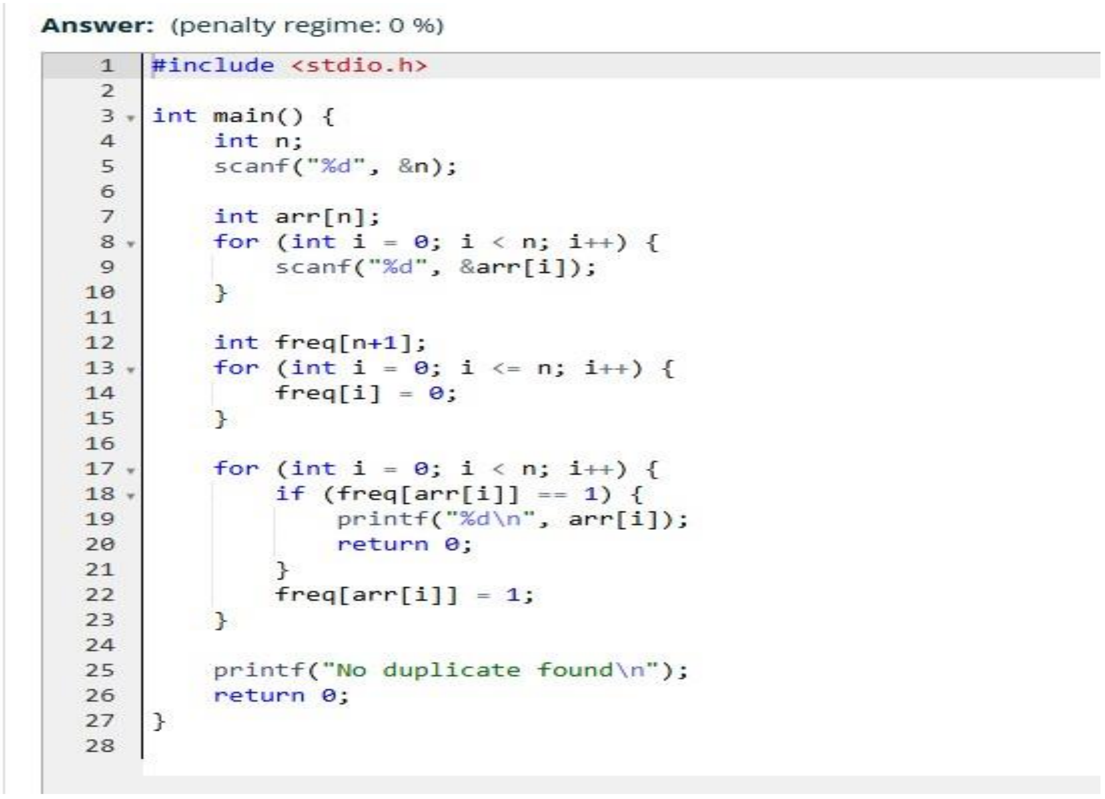
6.Competitive Programming

Q1:

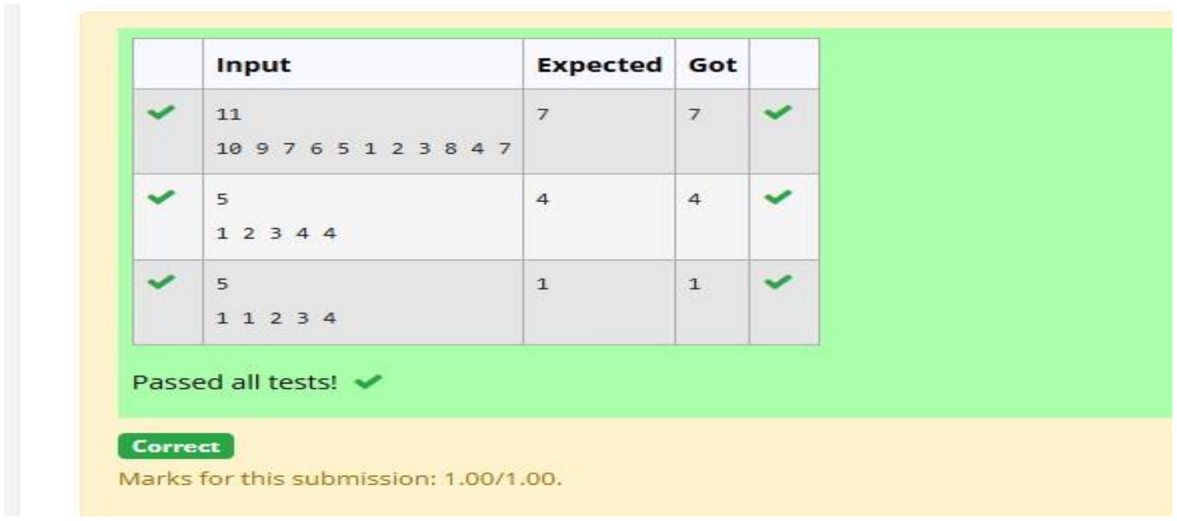

Question 1 | Correct Mark 1.00 out of 1.00 Flag question

Find Duplicate in Array.

Given a read only array of n integers between 1 and n, find one number that repeats.


Answer: (penalty regime: 0 %)

```c
#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 freq[n+1];
    for (int i = 0; i <= n; i++) {
        freq[i] = 0;
    }

    for (int i = 0; i < n; i++) {
        if (freq[arr[i]] == 1) {
            printf("%d\n", arr[i]);
            return 0;
        }
        freq[arr[i]] = 1;
    }

    printf("No duplicate found\n");
    return 0;
}

```



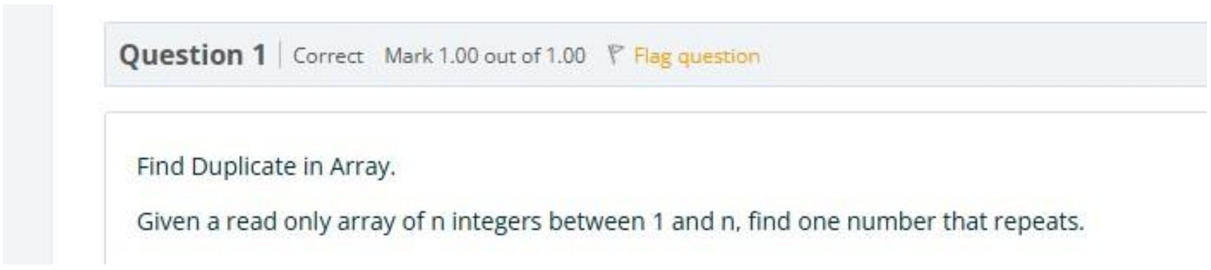
	Input	Expected	Got	
✓	11 10 9 7 6 5 1 2 3 8 4 7	7	7	✓
✓	5 1 2 3 4 4	4	4	✓
✓	5 1 1 2 3 4	1	1	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

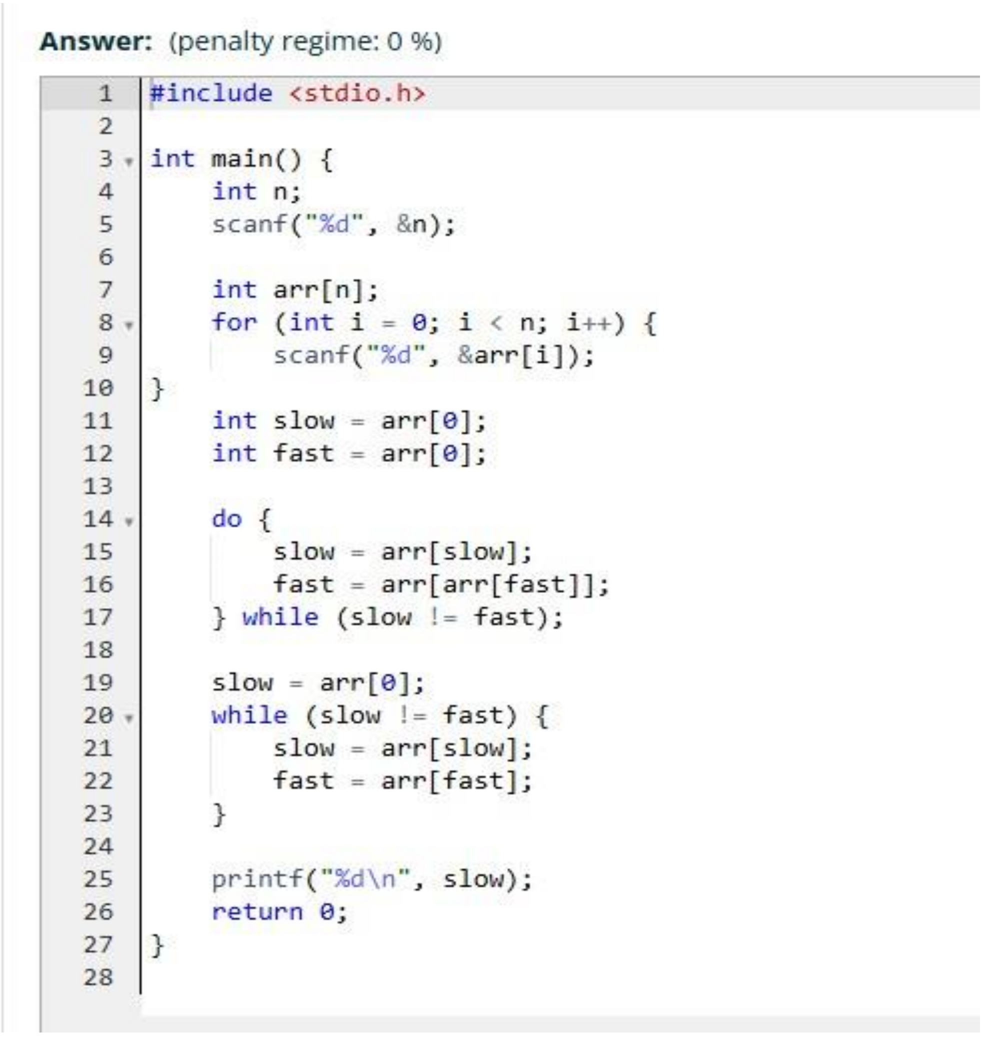
Q2:



Question 1 | Correct Mark 1.00 out of 1.00 Flag question

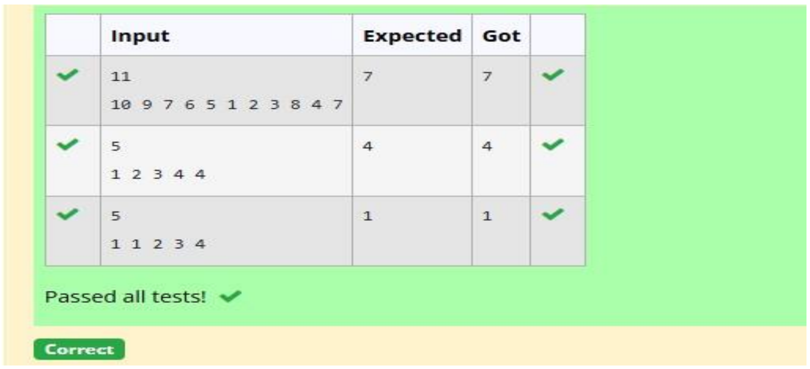
Find Duplicate in Array.

Given a read only array of n integers between 1 and n, find one number that repeats.



Answer: (penalty regime: 0 %)

```c
#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 slow = arr[0];
    int fast = arr[0];

    do {
        slow = arr[slow];
        fast = arr[arr[fast]];
    } while (slow != fast);

    slow = arr[0];
    while (slow != fast) {
        slow = arr[slow];
        fast = arr[fast];
    }

    printf("%d\n", slow);
    return 0;
}

```

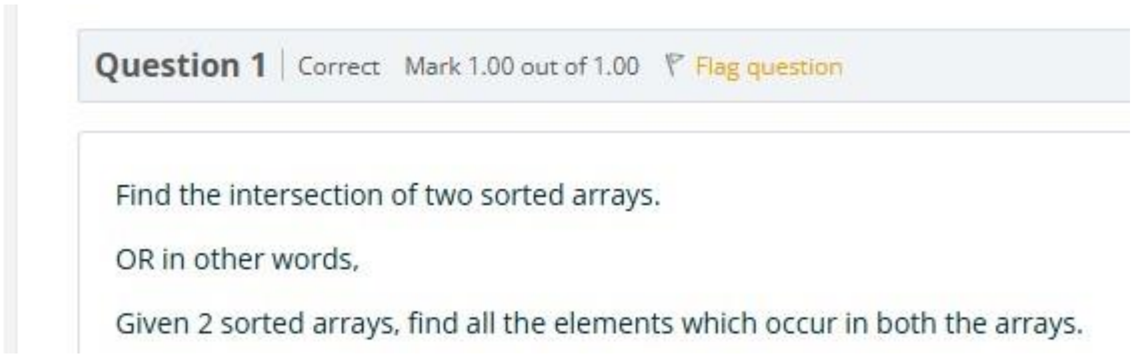


	Input	Expected	Got	
✓	11 10 9 7 6 5 1 2 3 8 4 7	7	7	✓
✓	5 1 2 3 4 4	4	4	✓
✓	5 1 1 2 3 4	1	1	✓

Passed all tests! ✓

Correct

Q3:

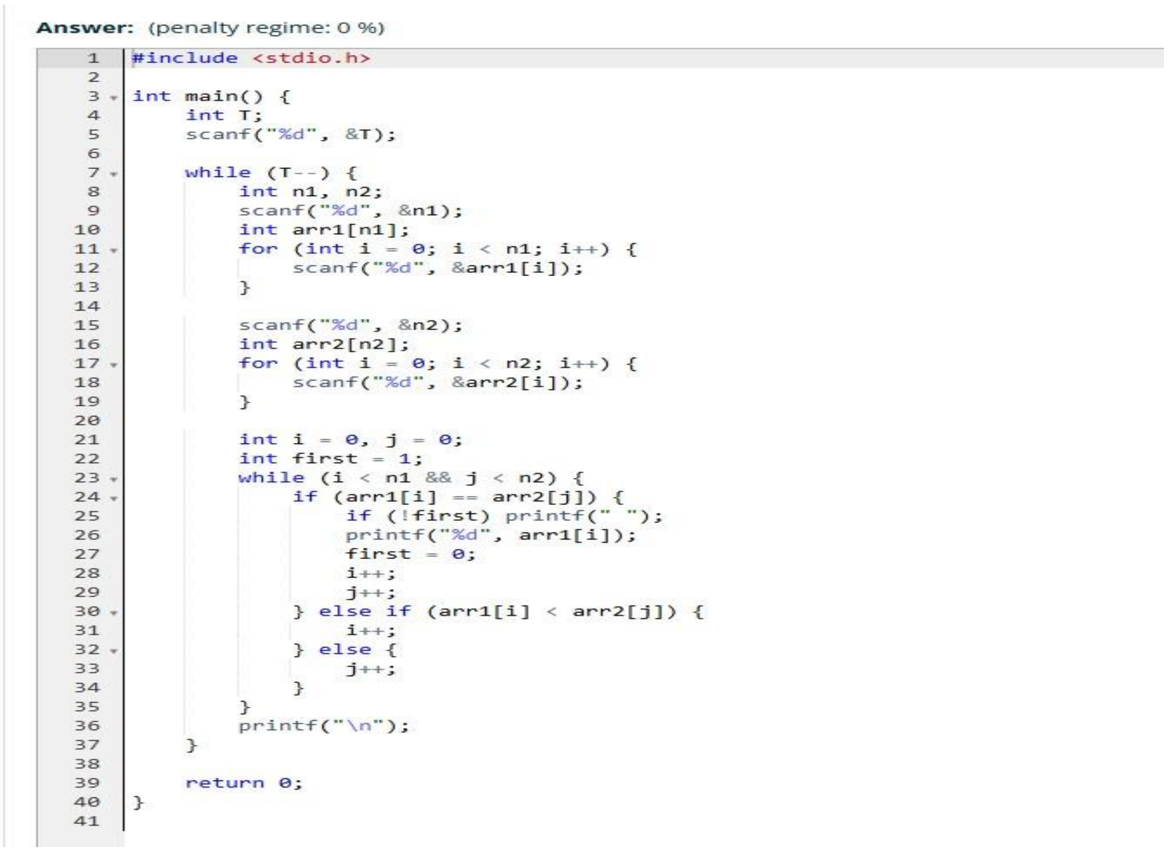


Question 1 | Correct Mark 1.00 out of 1.00 Flag question

Find the intersection of two sorted arrays.

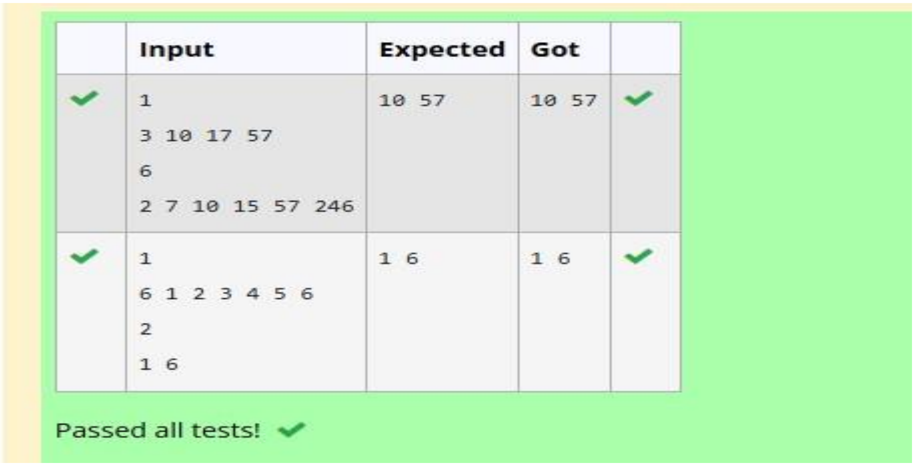
OR in other words,

Given 2 sorted arrays, find all the elements which occur in both the arrays.



Answer: (penalty regime: 0 %)

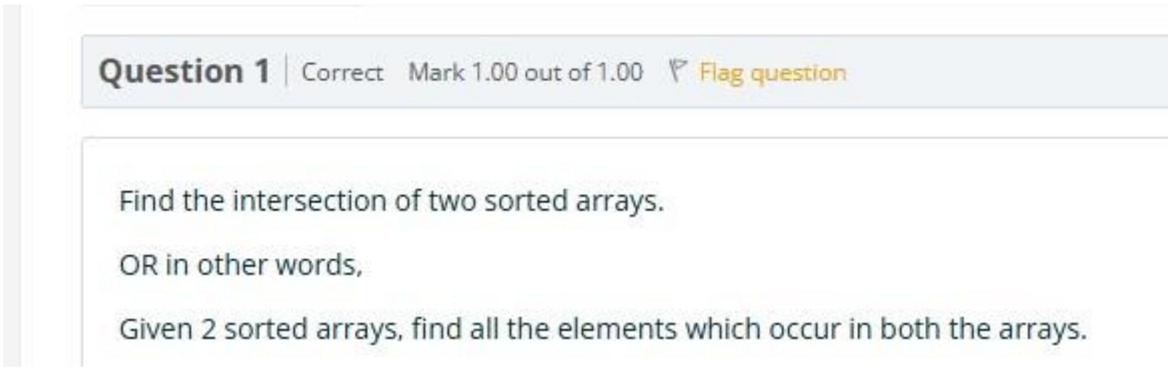
```c
#include <stdio.h>

int main() {
    int T;
    scanf("%d", &T);

    while (T--) {
        int n1, n2;
        scanf("%d", &n1);
        int arr1[n1];
        for (int i = 0; i < n1; i++) {
            scanf("%d", &arr1[i]);
        }

        scanf("%d", &n2);
        int arr2[n2];
        for (int i = 0; i < n2; i++) {
            scanf("%d", &arr2[i]);
        }

        int i = 0, j = 0;
        int first = 1;
        while (i < n1 && j < n2) {
            if (arr1[i] == arr2[j]) {
                if (!first) printf(" ");
                printf("%d", arr1[i]);
                first = 0;
                i++;
                j++;
            } else if (arr1[i] < arr2[j]) {
                i++;
            } else {
                j++;
            }
        }
        printf("\n");
    }

    return 0;
}
```



	Input	Expected	Got	
✓	1 3 10 17 57 6 2 7 10 15 57 246	10 57	10 57	✓
✓	1 6 1 2 3 4 5 6 2 1 6	1 6	1 6	✓

Passed all tests! ✓

Q4:

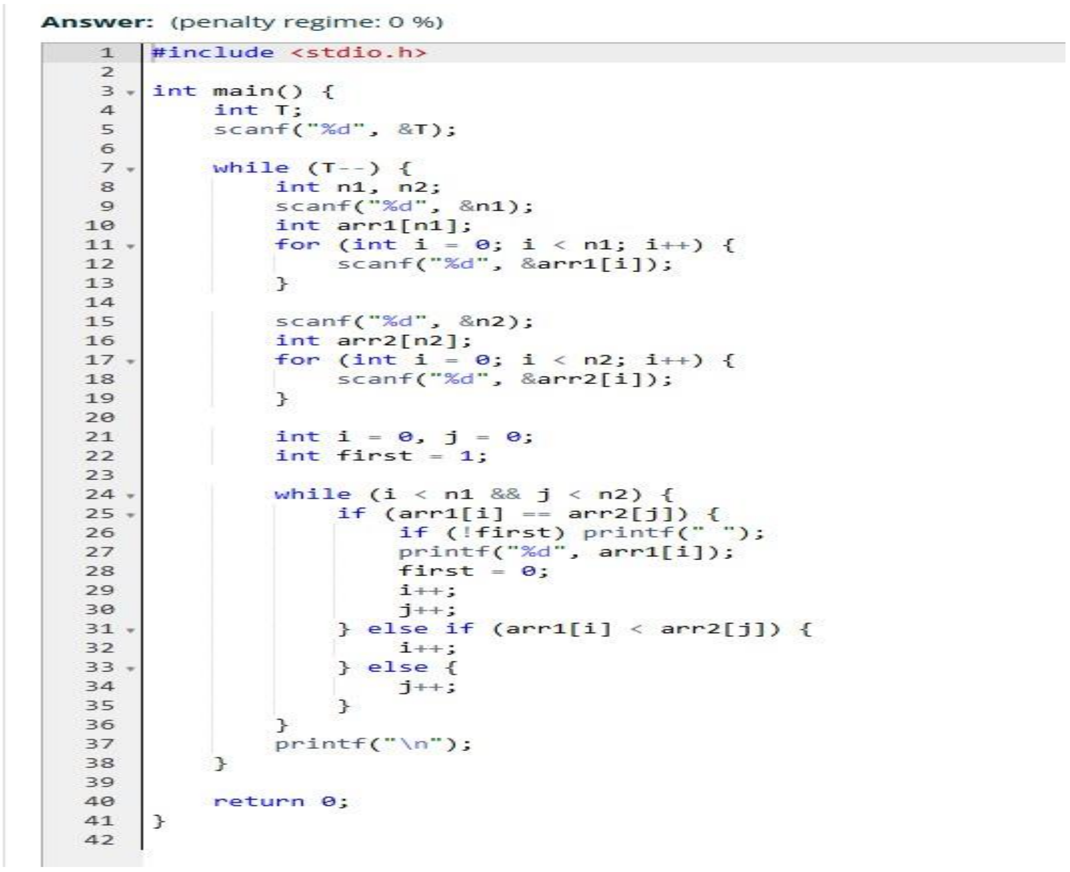


Question 1 | Correct Mark 1.00 out of 1.00 Flag question

Find the intersection of two sorted arrays.

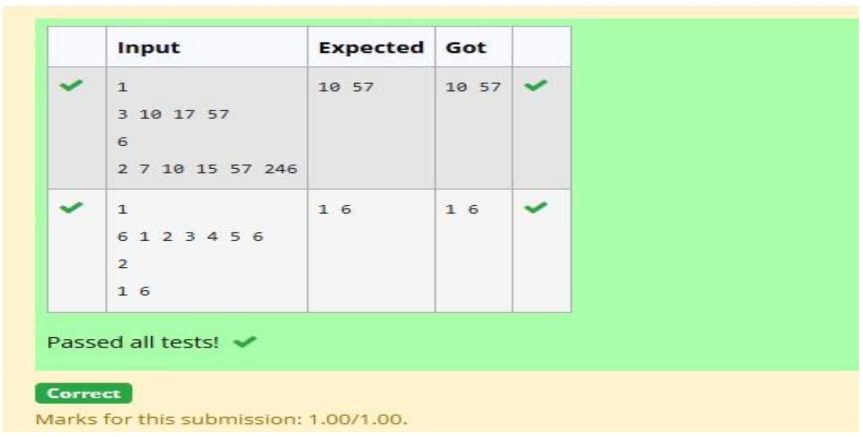
OR in other words,

Given 2 sorted arrays, find all the elements which occur in both the arrays.



Answer: (penalty regime: 0 %)

```c
#include <stdio.h>

int main() {
    int T;
    scanf("%d", &T);

    while (T--) {
        int n1, n2;
        scanf("%d", &n1);
        int arr1[n1];
        for (int i = 0; i < n1; i++) {
            scanf("%d", &arr1[i]);
        }

        scanf("%d", &n2);
        int arr2[n2];
        for (int i = 0; i < n2; i++) {
            scanf("%d", &arr2[i]);
        }

        int i = 0, j = 0;
        int first = 1;

        while (i < n1 && j < n2) {
            if (arr1[i] == arr2[j]) {
                if (!first) printf(" ");
                printf("%d", arr1[i]);
                first = 0;
                i++;
                j++;
            } else if (arr1[i] < arr2[j]) {
                i++;
            } else {
                j++;
            }
        }
        printf("\n");
    }

    return 0;
}
```



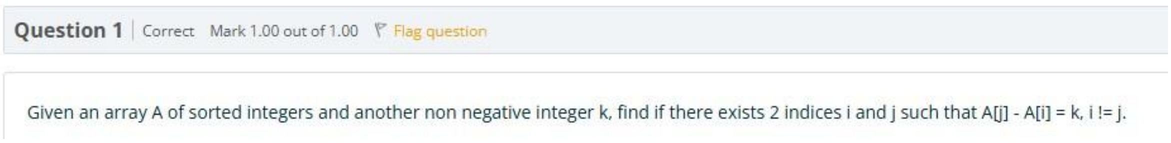
	Input	Expected	Got	
✓	1 3 10 17 57 6 2 7 10 15 57 246	10 57	10 57	✓
✓	1 6 1 2 3 4 5 6 2 1 6	1 6	1 6	✓

Passed all tests! ✓

Correct

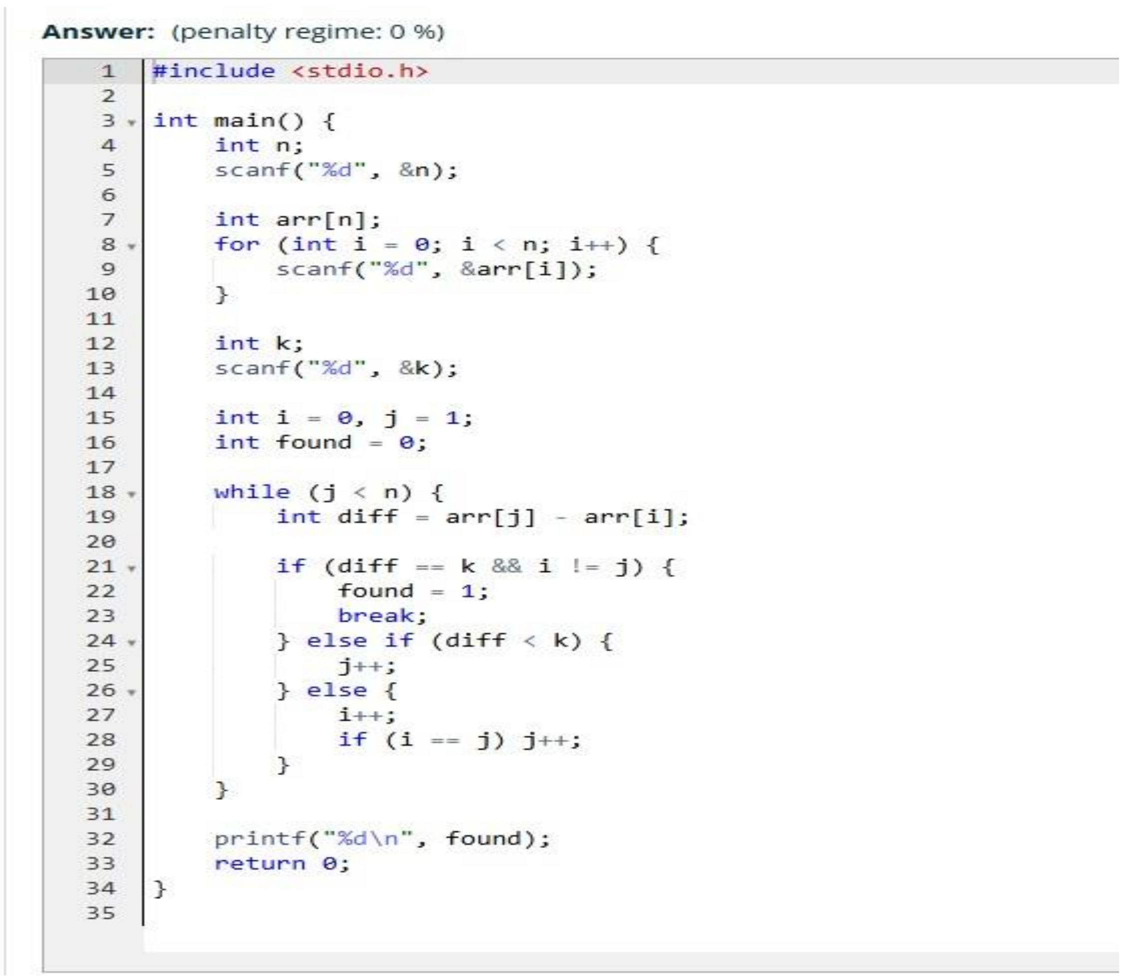
Marks for this submission: 1.00/1.00.

Q5:



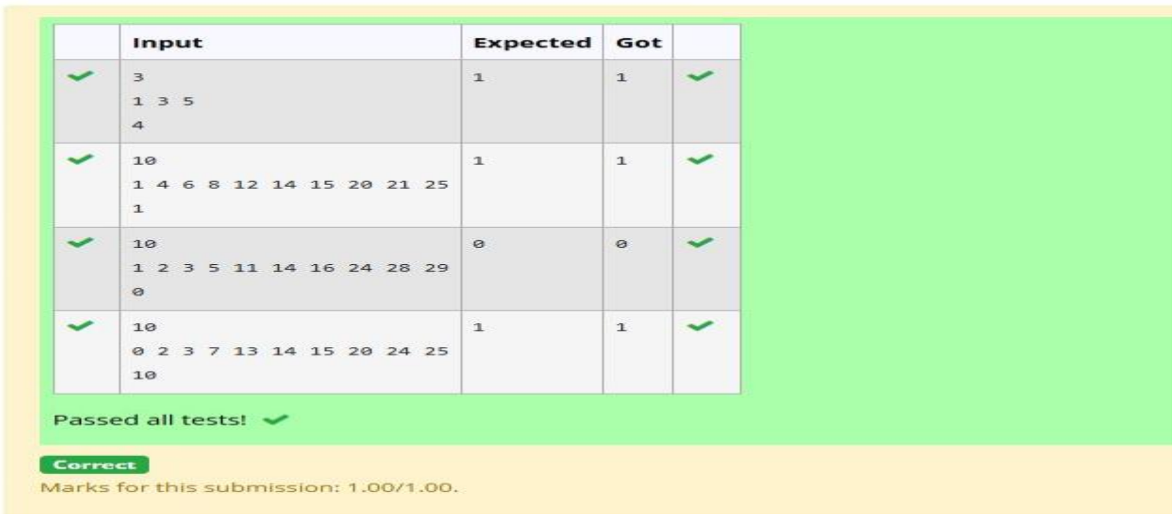
Question 1 | Correct Mark 1.00 out of 1.00 Flag question

Given an array A of sorted integers and another non negative integer k, find if there exists 2 indices i and j such that A[j] - A[i] = k, i != j.



Answer: (penalty regime: 0 %)

```c
#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 k;
    scanf("%d", &k);

    int i = 0, j = 1;
    int found = 0;

    while (j < n) {
        int diff = arr[j] - arr[i];

        if (diff == k && i != j) {
            found = 1;
            break;
        } else if (diff < k) {
            j++;
        } else {
            i++;
            if (i == j) j++;
        }
    }

    printf("%d\n", found);
    return 0;
}

```



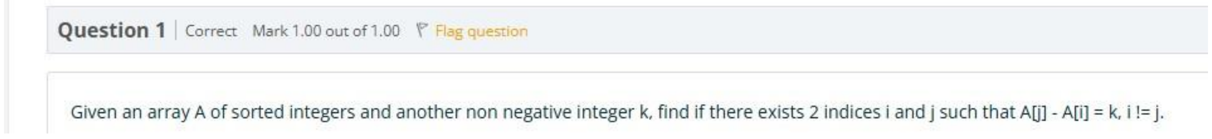
	Input	Expected	Got	
✓	3 1 3 5 4	1	1	✓
✓	10 1 4 6 8 12 14 15 20 21 25 1	1	1	✓
✓	10 1 2 3 5 11 14 16 24 28 29 0	0	0	✓
✓	10 0 2 3 7 13 14 15 20 24 25 10	1	1	✓

Passed all tests! ✓

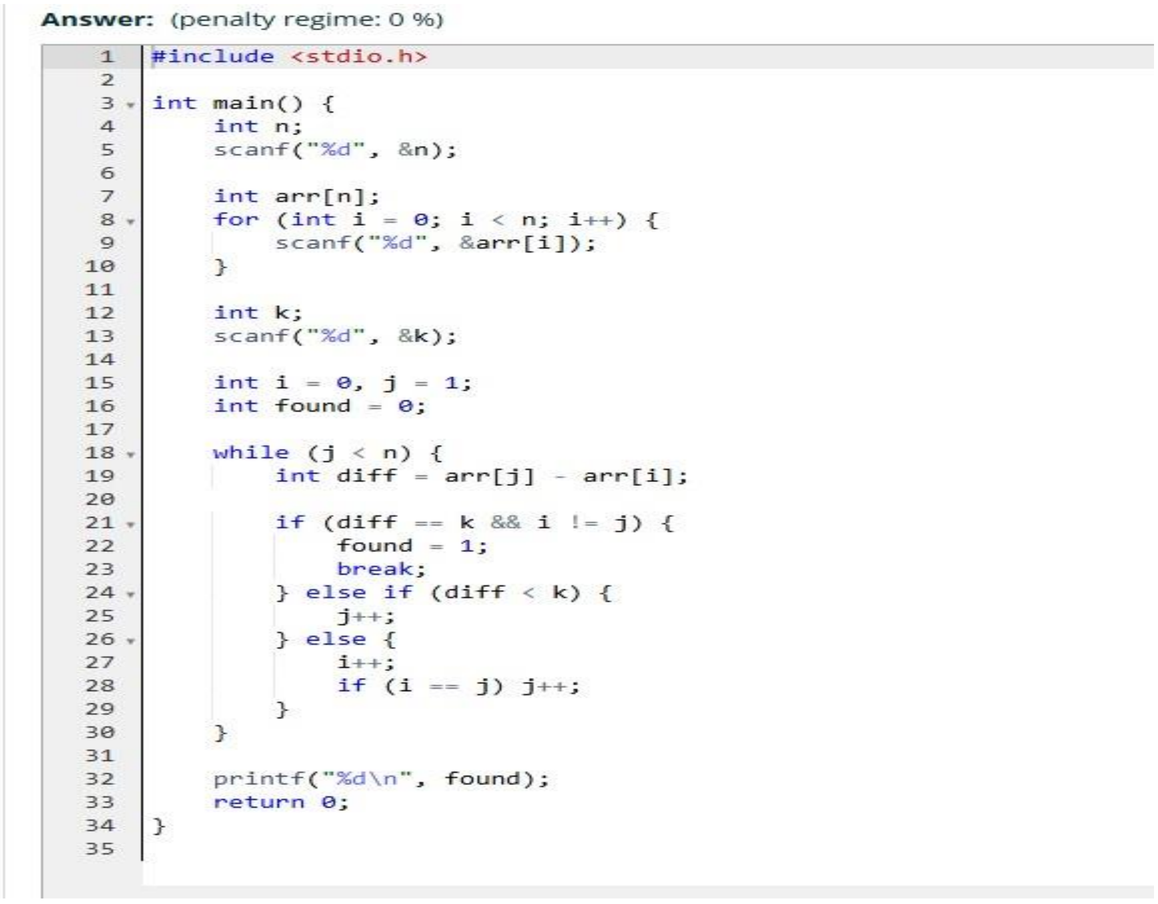
Correct

Marks for this submission: 1.00/1.00.

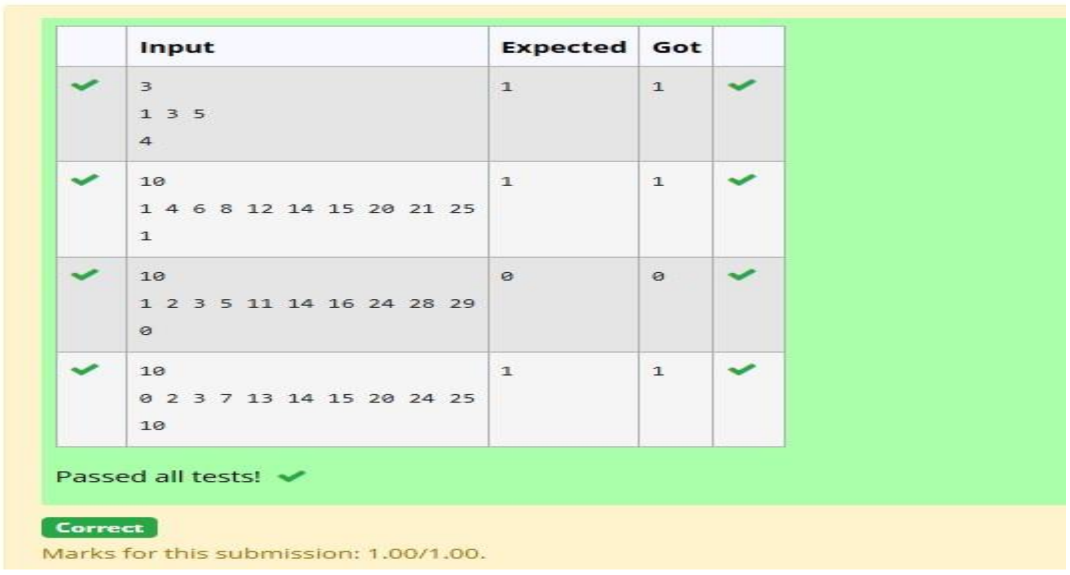
Q6:


Question 1 | Correct Mark 1.00 out of 1.00 Flag question

Given an array A of sorted integers and another non negative integer k, find if there exists 2 indices i and j such that A[j] - A[i] = k, i != j.


Answer: (penalty regime: 0 %)

```c
#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 k;
    scanf("%d", &k);

    int i = 0, j = 1;
    int found = 0;

    while (j < n) {
        int diff = arr[j] - arr[i];

        if (diff == k && i != j) {
            found = 1;
            break;
        } else if (diff < k) {
            j++;
        } else {
            i++;
            if (i == j) j++;
        }
    }

    printf("%d\n", found);
    return 0;
}
```



	Input	Expected	Got	
✓	3 1 3 5 4	1	1	✓
✓	10 1 4 6 8 12 14 15 20 21 25 1	1	1	✓
✓	10 1 2 3 5 11 14 16 24 28 29 0	0	0	✓
✓	10 0 2 3 7 13 14 15 20 24 25 10	1	1	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.