Status	Finished
Started	Saturday, 1 November 2025, 6:12 PM
Completed	Saturday, 1 November 2025, 7:10 PM
Duration	58 mins 25 secs

Question **1**

Correct

In a betting game involving the roll of a dice, Sandeep gains Rs.X if an odd number turns up and he loses Rs.Y is an even number turns up. The numbers shown on the face of the dice in a certain number of games is passed as input. The values of X and Y are also passed as input. The program must print the net gain or loss as the output.

Input Format:

First line will contain the numbers shown on the face of the dice separated by one or more spaces.
Second line will contain the value of X
Third line will contain the value of Y

Output Format:

The net gain or loss (loss will be a negative value)

Sample Input/Output:

Example 1:

Input:

1 4 3
10
30

Output:

-10

Explanation:

He gains 20 rupees for 1 and 3 and loses 30 rupees for 4. Hence there is a net loss of 20-30 = -10

Example 2:

Input:

4 6 1 2 1
50
25

Output:

25

He gains 100 rupees for 1,1 and loses 75 rupees for 4,6,2. Hence there is a net gain of 100-75 = 25

For example:

Input	Result
1 4 3 10 30	-10
4 6 1 2 1 50 25	25

Answer: (penalty regime: 0 %)

```c
#include<stdio.h>
int main()
{
    int dice[100],n=0;
    int X,Y;
    int num;
    int gain=0,loss=0;
    while(scanf("%d",&num)==1){
        dice[n++]=num;
        if(getchar()=='\n')break;

    }
    scanf("%d",&X);
    scanf("%d",&Y);
    for(int i=0;i<n;i++){
        if(dice[i]%2==0)
          loss+=Y;
          else
          gain+=X;
    }
    printf("%d",gain-loss);
    return 0;
}
```

	Input	Expected	Got	
✓	1 4 3 10 30	-10	-10	✓
✓	4 6 1 2 1 50 25	25	25	✓

Passed all tests! ✓

Question **2**

Correct

Given a set of numbers where all other numbers are multiple of the smallest number, the program must find the count of the common factors C excluding 1.

Input Format:

First line will contain the integer value N representing how many numbers are passed as input.
Next N lines will have the numbers.

Output Format:

First line will contain the count of common factors C.

Constraints:

N will be from 2 to 20.

Sample Input/Output:

Example 1:

Input:

2
100
75

Output:

2

Explanation:

The common factors excluding 1 are 5,25. Hence output is 2

Example 2:

Input:

3
10
20
30

Output:

3

For example:

Input	Result
2 100 75	2
3 10 20 30	3

Answer: (penalty regime: 0 %)

```c
#include<stdio.h>
int gcd(int a,int b){
    while(b!=0){
        int temp=b;
        b=a%b;
        a=temp;
    }
    return a;
}
int main(){
    int N;
    scanf("%d",&N);
    int nums[20];
    for(int i=0;i<N;i++){
        scanf("%d",&nums[i]);
    }
    int common_gcd=nums[0];
    for(int i=1;i<N;i++){
        common_gcd=gcd(common_gcd,nums[i]);
    }
    int count=0;
    for(int i=2;i<=common_gcd;i++){
        if(common_gcd%i==0){
            count++;
        }
```

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

	Input	Expected	Got	
✓	2 100 75	2	2	✓
✓	3 10 20 30	3	3	✓

Passed all tests! ✓

Question **3**

Correct

Two whole numbers N1 and N2 are passed as input. The program must print the number of primes present between N1 and N2 (the range is inclusive of N1 and N2)

Input Format:

First line will contain the value of the first number N1
Second line will contain the value of the second number N2

Output Format:

First line will contain the count of prime numbers between N1 and N2

Sample Input/Output:

Example 1:

Input:

6142
6200

Output:

6

Explanation:

The prime numbers within the range 6142 to 6200 are 6143, 6151, 6163, 6173, 6197, 6199

Example 2:

Input:

38
70

Output:

7

Explanation:

The prime numbers within the range 38 to 70 are 41, 43, 47, 53, 59, 61, 67

For example:

Input	Result
6142 6200	6
38 70	7

Answer: (penalty regime: 0 %)

```c
#include<stdio.h>
#include<math.h>
int isPrime(int n){
    if(n<2)
    return 0;
    for(int i=2;i<=sqrt(n);i++){
        if(n%i==0)
        return 0;
    }
    return 1;
}
int main(){
    int N1,N2;
    scanf("%d",&N1);
    scanf("%d",&N2);
    int count=0;
    for(int i=N1;i<=N2;i++){
        if(isPrime(i))
        count++;
    }
    printf("%d",count);
    return 0;
}
```

	Input	Expected	Got	
✓	6142 6200	6	6	✓
✓	38 70	7	7	✓

Passed all tests! ✓

Question **4**

Correct

An integer value N is passed as the input. The program must print the first N terms in the Fibonacci sequence.

Input Format:

The first line denotes the value of N.

Output Format:

The first N terms in the Fibonacci sequence (with each term separated by a space)

Boundary Conditions:

3 <= N <= 50

Example Input/Output 1:

Input:

5

Output:

0 1 1 2 3

Example Input/Output 2:

Input:

10

Output:

0 1 1 2 3 5 8 13 21 34

For example:

Input	Result
5	0 1 1 2 3
10	0 1 1 2 3 5 8 13 21 34

Answer: (penalty regime: 0 %)

```
1 #include<stdio.h>
```

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

	Input	Expected	Got	
✓	5	0 1 1 2 3	0 1 1 2 3	✓
✓	10	0 1 1 2 3 5 8 13 21 34	0 1 1 2 3 5 8 13 21 34	✓

Passed all tests! ✓

Question **5**

Correct

The runs scored by a cricket team in the first and second innings of N test cricket matches are passed as input. The program must print the average of first and second innings (with precision upto two decimal places).

Input Format:

The first line denotes the value of N.
Next N lines will contain the first and second innings score separated by a space.

Output Format:

The first line contains the average of first innings score.
The second line contains the average of second innings score.

Boundary Conditions:

2 <= N <= 20
The value of the runs will be from 0 to 1000.

Example Input/Output 1:

Input:

3
250 200
450 300
200 250

Output:

300.00
250.00

For example:

Input	Result
3 250 200	300.00 250.00

Input	Result
450 300 200 250	

Answer: (penalty regime: 0 %)

```c
#include<stdio.h>
int main(){
    int N,i;
    float firstSum=0,secondSum=0;
    int first,second;
    scanf("%d",&N);
    for(i=0;i<N;i++){
        scanf("%d%d",&first,&second);
        firstSum+=first;
        secondSum+=second;
    }
    printf("%.2f\n",firstSum/N);
    printf("%.2f",secondSum/N);
    return 0;
}
```

	Input	Expected	Got	
✓	3 250 200 450 300 200 250	300.00 250.00	300.00 250.00	✓

Passed all tests! ✓

Question **6**

Correct

A number N is passed as the input. The program must print the next immediate prime number.

Input Format:

The first line will contain N.

Output Format:

The first line will contain the integer value of next immediate prime number.

Boundary Conditions:

1 <= N <= 999999

Example Input/Output 1:

Input:

11

Output:

13

Example Input/Output 2:

Input:

2

Output:

3

For example:

Input	Result
11	13

Input	Result
2	3

Answer: (penalty regime: 0 %)

```c
#include<stdio.h>
#include<math.h>
int isPrime(int num){
    if(num<=1)
    return 0;
    for(int i=2;i<=sqrt(num);i++){
        if(num%i==0)
        return 0;
    }
    return 1;
}
int main(){
    int n;
    scanf("%d",&n);
    int next=n+1;
    while(!isPrime(next)){
        next++;
    }
    printf("%d",next);
    return 0;
}
```

	Input	Expected	Got	
✓	11	13	13	✓
✓	2	3	3	✓

Passed all tests! ✓

Question **7**

Correct

The program must accept two integers X and Y and print the odd integers between them.

Input Format:

The first line denotes the value of X.
The second line denotes the value of Y.

Output Format:

The first line contains the odd integers between X and Y separated by a space.

Boundary Conditions:

-999999 <= X <= 9999999
X < Y <= 9999999

Example Input/Output 1:

Input:

1
11

Output:

3 5 7 9

Example Input/Output 2:

Input:

24
30

Output:

25 27 29

For example:

Input	Result
1 11	3 5 7 9

Input	Result
24 30	25 27 29

Answer: (penalty regime: 0 %)

```c
#include<stdio.h>
int main(){
    int x,y;
    scanf("%d",&x);
    scanf("%d",&y);
    for(int i=x+1;i<y;i++){
        if(i%2!=0){
            printf("%d",i);
            if(i+2<y){
                printf(" ");
            }
        }
    }
    return 0;
}
```

	Input	Expected	Got	
✓	1 11	3 5 7 9	3 5 7 9	✓
✓	24 30	25 27 29	25 27 29	✓

Passed all tests! ✓

Question **8**

Correct

An integer value N is passed as the input. The program must print YES if N is prime number. Else the program must print NO.

Input Format:

The first line denotes the value of N.

Output Format:

YES or NO based on if N is a prime number or not. (The OUTPUT is CASE SENSITIVE).

Boundary Conditions:

2 <= N <= 9999999

Example Input/Output 1:

Input:

19

Output:

YES

Example Input/Output 2:

Input:

189210

Output:

NO

For example:

Input	Result
19	YES
189210	NO

Answer: (penalty regime: 0 %)

```
1 #include<stdio.h>
```

```c
#include<stdio.h>
#include<math.h>
int main(){
    int n,i,isPrime=1;
    scanf("%d",&n);
    if(n<2){
        printf("NO");
        return 0;
    }
        for(i=2;i<=sqrt(n);i++){
            if(n%i==0){
                isPrime=0;
                break;
            }
        }
        if(isPrime)
        printf("YES");
        else
        printf("NO");
        return 0;
}
```

	Input	Expected	Got	
✓	19	YES	YES	✓
✓	189210	NO	NO	✓

Passed all tests! ✓

Question **9**

Correct

The program must accept N integers and print the second largest value among the N integers.

Input Format:

The first line denotes the value of N.
Next N lines will contain the N integer values.

Output Format:

The first line contains the second largest integer.

Boundary Conditions:

2 <= N <= 100
The value of the integers will be from -999999 to 999999.

Example Input/Output 1:

Input:

3
100
2200
345

Output:

345

Example Input/Output 2:

Input:

6
-23
-256
-87
-90
-11019
-2

Output:

-23

For example:

Input	Result
3 100 2200 345	345
6 -23 -256 -87 -90 -11019 -2	-23

Answer: (penalty regime: 0 %)

```c
#include<stdio.h>
int main(){
    int n;
    scanf("%d",&n);
    long long num,largest,secondLargest;
    scanf("%lld",&num);
    largest=num;
    secondLargest=-9999999999LL;
    for(int i=1;i<n;i++){
        scanf("%lld",&num);
        if(num>largest){
            secondLargest=largest;
            largest=num;
        }
    else if(num>secondLargest&&num<largest){
        secondLargest=num;
    }
    }
    printf("%lld",secondLargest);
    return 0;
}
```

	Input	Expected	Got	
✓	3 100	345	345	✓

	Input	Expected	Got	
	2200 345			
✓	6 -23 -256 -87 -90 -11019 -2	-23	-23	✓

Passed all tests! ✓