CS23333-Object Oriented Programming Using Java-2023

Quiz navigation

1 2 3

Show one page at a time

Finish review

Status	Finished
Started	Saturday, 5 October 2024, 1:34 PM
Completed	Saturday, 5 October 2024, 7:41 PM
Duration	6 hours 6 mins

Question **1**

Correct

Marked out of 5.00

Flag question

You and your friend are movie fans and want to predict if the movie is going to be a hit!

The movie's success formula depends on 2 parameters:

the acting power of the actor (range 0 to 10)

the critic's rating of the movie (range 0 to 10)

The movie is a hit if the acting power is excellent (more than 8) or the rating is excellent (more than 8). This holds true except if either the acting power is poor (less than 2) or rating is poor (less than 2), then the movie is a flop. Otherwise the movie is average.

Write a program that takes 2 integers:

the first integer is the acting power

second integer is the critic's rating.

You have to print Yes if the movie is a hit, Maybe if the movie is average and No if the movie is flop.

Example input:

9 5

Output:

Yes

Example input:

1 9

Output:

No

Example input:

6 4

Output:

Maybe

For example:

Input	Result
9 5	Yes
1 9	No
6 4	Maybe

Answer: (penalty regime: 0 %)

```
import java.util.Scanner;
public class main{
    public static void main(String args[]){
        Scanner s=new Scanner(System.in);
        int a=s.nextInt();
        int b=s.nextInt();
        if(a<2||b<2){
                System.out.println("No");
        }else if(a>8 || b>8){
            System.out.println("Yes");
        }
        else{
            System.out.println("Maybe");
        }
    }
}
```

	Input	Expected	Got	
	9 5	Yes	Yes	

	Input	Expected	Got	
	1 9	No	No	
	6 4	Maybe	Maybe	

Passed all tests!

Question **2**

Correct

Marked out of 5.00

Flag question

Write a program that takes as parameter an integer n.

You have to print the number of zeros at the end of the factorial of n.

For example, 3! = 6. The number of zeros are 0. 5! = 120. The number of zeros at the end are 1.

Note: n! < 10^5

Example Input:

3

Output:

0

Example Input:

60

Output:

14

Example Input:

100

Output:

24

Example Input:

1024

Output:

253

For example:

Input	Result
3	0
60	14
100	24
1024	253

Answer: (penalty regime: 0 %)

Reset answer

```java
// Java program to count trailing 0s in n!
import java.io.*;
import java.util.Scanner;
class prog {
    // Function to return trailing
    // 0s in factorial of n
    static int findTrailingZeros(int n)
    {
        if (n < 0) // Negative Number Edge Case
            return -1;

        // Initialize result

        int count=0;
        // Keep dividing n by powers
        // of 5 and update count
        for (int i = 5; n / i >= 1;  i*=5   ){
            count += n / i;
}
        return count;
    }

    // Driver Code
    public static void main(String[] args)
    {
        Scanner sc= new Scanner(System.in);
        int n=sc.nextInt();
        int res=findTrailingZeros(n);
        System.out.println(res);
    }
}
```

	Input	Expected	Got	
	3	0	0	
	60	14	14	
	100	24	24	
	1024	253	253	

Passed all tests!

Question **3**

Correct

Marked out of 5.00

Flag question

Consider the following sequence:

1st term: 1

2nd term: 1 2 1

3rd term: 1 2 1 3 1 2 1

4th term: 1 2 1 3 1 2 1 4 1 2 1 3 1 2 1

And so on. Write a program that takes as parameter an integer n and prints the nth terms of this sequence.

Example Input:

1

Output:

1

Example Input:

4

Output:

1 2 1 3 1 2 1 4 1 2 1 3 1 2 1

For example:

Input	Result
1	1
2	1 2 1
3	1 2 1 3 1 2 1
4	1 2 1 3 1 2 1 4 1 2 1 3 1 2 1

Answer: (penalty regime: 0 %)

```
import java.util.Scanner;
public class main{
    public static void main(String args[]){
        Scanner s=new Scanner(System.in);
        int a=s.nextInt();
        String currentTerm="1";
        for(int i=2;i<=a;i++){
            currentTerm=currentTerm+" "+i+" "+currentTerm;
        }
        System.out.println(currentTerm);
    }
}
```

	Input	Expected	Got	
	1	1	1	
	2	1 2 1	1 2 1	
	3	1 2 1 3 1 2 1	1 2 1 3 1 2 1	
	4	1 2 1 3 1 2 1 4 1 2 1 3 1 2 1	1 2 1 3 1 2 1 4 1 2 1 3 1 2 1	

Passed all tests!

Finish review