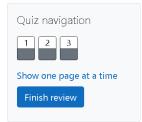
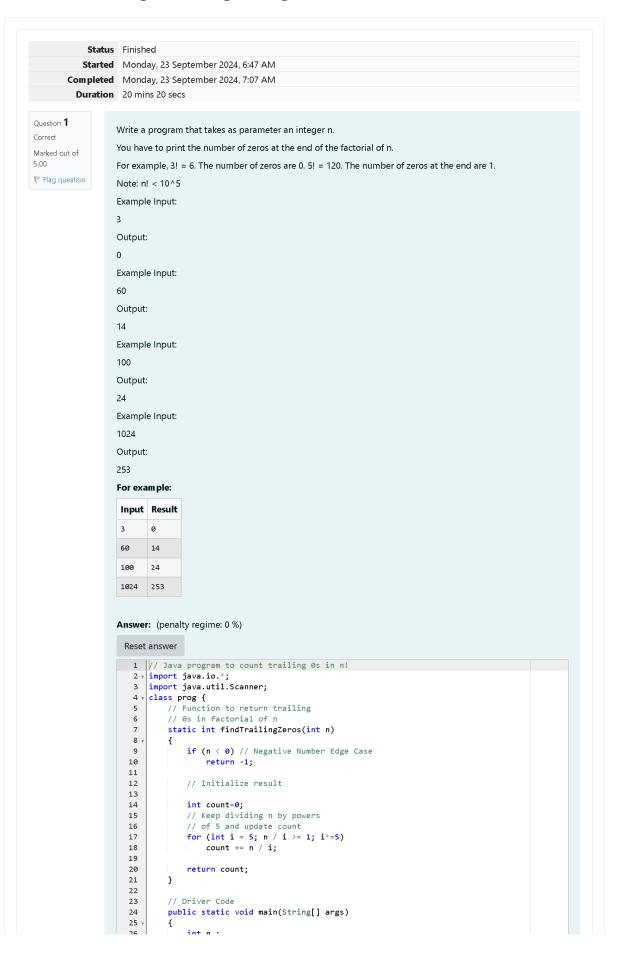
CS23333-Object Oriented Programming Using Java-2023





```
27
            Scanner sc= new Scanner(System.in);
            System.out.println();
28
29
             n=sc.nextInt();
            System.out.println(findTrailingZeros(n));
30
31
        }
   }
32
33
```

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

Passed all tests! 🗸

Question 2 Correct Marked out of

▼ 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:

19

Output:

No

Example input:

64

Output:

Maybe

For example:

Input	Result	
9 5	Yes	
1 9	No	
6 4	Maybe	

Answer: (penalty regime: 0 %)

```
1 v import java.util.Scanner;
2 v public class movie{
3 .
       public static void main(String[] args){
          Scanner scn= new Scanner(System.in);
           int ap=scn.nextInt();
           int cr=scn.nextInt();
           if (ap<2||cr<2){
               System.out.print("No");
```

```
| Input | Expected | Got | |
| ✓ 9 5 | Yes | Yes | ✓ |
| ✓ 1 9 | No | No | ✓ |
| ✓ 6 4 | Maybe | Maybe | ✓ |
| 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:

121312141213121

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 %)

```
1 | import java.util.Scanner;
 2 v public class sequence{
 3 1
         public static void main(String args[]){
 4
             Scanner scn=new Scanner(System.in);
              int n=scn.nextInt();
             String result=pattern(n);
             System.out.print(result);
 8
         public static String pattern(int n){
9 .
             if(n==1){
return "1";
10
11
12
             String prev=pattern(n-1);
return prev+' '+n+' '+prev;
13
14
         }
15
16 }
```

	Input	Expected Got	
~	1	1	~
~	2	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