Dashboard / My courses / CS23333-OOPUJ-2023 / Lab-02-Flow Control Statements / Lab-02-Logic Building

Status	Finished
Started	Thursday, 3 October 2024, 9:55 PM
Completed	Thursday, 3 October 2024, 10:01 PM
Duration	6 mins 7 secs

Question **1**

Correct

Marked out of 5.00

You have recently seen a motivational sports movie and want to start exercising regularly. Your coach tells you that it is important to get up early in the morning to exercise. She sets up a schedule for you:

On weekdays (Monday - Friday), you have to get up at 5:00. On weekends (Saturday & Sunday), you can wake up at 6:00. However, if you are on vacation, then you can get up at 7:00 on weekdays and 9:00 on weekends.

Write a program to print the time you should get up.

Input Format

Input containing an integer and a boolean value.

The integer tells you the day it is (1-Sunday, 2-Monday, 3-Tuesday, 4-Wednesday, 5-Thursday, 6-Friday, 7-Saturday). The boolean is true if you are on vacation and false if you’re not on vacation.

You have to print the time you should get up.

Example Input:

1 false

Output:

6:00

Example Input:

5 false

Output:

5:00

Example Input:

1 true

Output:

9:00

For example:

Input	Result
`1 false`	`6:00`
`5 false`	`5:00`
`1 true`	`9:00`

Answer: (penalty regime: 0 %)

```java
import java.util.Scanner;

public class WakeUpTime {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        int day = scanner.nextInt();
        boolean onVacation = scanner.nextBoolean();
        String wakeUpTime;
        if (onVacation) {
            if (day == 1 || day == 7) { // Sunday or Saturday
                wakeUpTime = "9:00";
            } else { // Monday to Friday
                wakeUpTime = "7:00";
            }
        } else {
            if (day == 1) { // Sunday
```

```
                wakeUpTime = "6:00";
            } else if (day >= 2 && day <= 6) { // Monday to Friday
                wakeUpTime = "5:00";
            } else { // Saturday
                wakeUpTime = "6:00";
            }
        }
        System.out.println(wakeUpTime);
    }
}


```

	Input	Expected	Got	
✓	1 false	6:00	6:00	✓
✓	5 false	5:00	5:00	✓
✓	1 true	9:00	9:00	✓

Passed all tests! ✓

Question **2**

Correct

Marked out of 5.00

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.*;
public class main{
    public static void main(String arg[]){
        Scanner obj=new Scanner(System.in);
        int a=obj.nextInt();
        String b="1";
        for(int i=2;i<=a;i++){
            b+=" "+i+" "+b;
        }
        System.out.println(b);
    }
}
```

	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! ✓

Question **3**

Correct

Marked out of 5.00

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 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;
        int count=0;
        // Initialize result


        // Keep dividing n by powers
        // of 5 and update count
        // 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 result=findTrailingZeros(n);
        System.out.println(result);
    }
}
```

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

Passed all tests! ✓

◄ Lab-02-MCQ

Jump to...

Lab-03-MCQ ►