

**PROBLEM OF THE
MONTH**



January, 2017

MATHEMATICS

5 points:

The triangle ABC is defined by coordinates of its vertices $A=(3,5)$, $B=(10,7)$, $C=(8,11)$. Find its area.

Hint: Draw this triangle as the triangle inscribed into the rectangle with vertex coordinates $(3,5)$, $(10,5)$, $(10,11)$, $(3,11)$.

10 points:

The pentagon ABCDE is defined by coordinates of its five vertices on a plane. It is known that all coordinates are integer numbers. Is it possible that the area of the pentagon is equal to $2017/5$?

Hint: Calculate the area of the triangle with vertices $(0,0)$, (m,n) , (M,N) . What values can this area take if m,n,M,N are integers?

PHYSICS

5 pt. Ivar has 50 identical incandescent mini light bulbs, each having constant resistance of 10 Ohms, designed for the current not exceeding 0.35 Amps. He wants to make the brightest possible single-strand Christmas String Lighting Set, powered by the US standard AC 110 Volts outlet. How many bulbs should Ivar use? What will be the value of current in each of the bulbs?

Hint: Find the maximum voltage drop that each bulb may have, and find the minimum number of bulbs connected in series, such that the voltage on each bulb is below that maximum. That will be the optimal string . Why?

10 pt. Ingrid has 350 identical incandescent mini light bulbs, each having constant resistance of 10 Ohms, designed for the current not exceeding 0.35 Amps. She wants to make the brightest possible multi-strand Christmas String Lighting Set, powered by the US standard AC 110 Volts outlet. How many bulbs should Ingrid use and how should she connect these bulbs? What will be the total power of this set?

Hint: Find the maximum voltage drop that each bulb may have, and find the minimum number of bulbs connected in series, such that the voltage on each bulb is below that maximum. That will be an optimal single string . Why? Now you can take several of these string and connect them in parallel.

CHEMISTRY

5 points:

The good thing about YouTube is that everybody can upload their video there. The bad thing about YouTube is ... the same. Please, watch this video: <https://www.youtube.com/watch?v=FofPjj7v414> (starting from 0:46) and explain what mistake the authors made.

Hint:

This video contains several minor inaccuracies, but there is one absolutely blatant error there. Watch the video starting from 0:40 till 1:05.

10 points:

In the "Taboo" (a British drama TV series, 2016-17) the procedure is described for the detection of the traces of arsenic in biological specimens (in that particular case, it was the content of a stomach of the poisoned man). According to the movie, the specimen is mixed with "potassium oxide, calcium oxide, and nitric acid". Watch the movie and tell if this method can really detect arsenic, and if not, what should be changed to make the detection of arsenic possible.

Hint:

In this movie, the apparatus for arsenic detection and the detection procedure are described correctly, but the chemicals they used are totally weird. Try to tell why it is not possible to convert arsenic in a volatile form using these chemicals?

BIOLOGY

5 points:

In deserts it rains only several days a year, usually during "winter" season. How do plants adapt to such conditions? Please give examples.

10 points:

Suppose there is a gene variant which encodes for a defective protein. It is known that some people with this gene variant have A serious disorder, caused by the defective protein. Other people with that same variant are seemingly healthy. Please give as many explanations as you can for how this is possible.

COMPUTER SCIENCE

- You can write and compile your code here:
<http://www.tutorialspoint.com/codingground.htm>
- Your program should be written in Java or Python
- No GUI should be used in your program: eg., easygui in Python. All problems in POM require only text input and output. GUI usage complicates solution validation, for which we are also using *codingground* site. Solutions with GUI will have points deducted or won't receive any points at all.
- Please make sure that the code compiles and runs on
<http://www.tutorialspoint.com/codingground.htm> before submitting it.
- Any input data specified in the problem should be supplied as user input, not hard-coded into the text of the program.
- Submit the problem in a plain text file, such as .txt, .dat, etc.
No .pdf, .doc, .docx, etc!

Introduction

We would like to introduce recursion -- a powerful tool to approach algorithm problems.

Example 1

For example, say you want to write a function that computes 10! ("!" stands for factorial, and is defined as $10! = 10 * 9 * 8 * \dots * 2 * 1$). A traditional way is to do that is like this:

```
def factorialSimple(n):
    total = 1
    while n > 1:
        total = total * n
        n = n - 1
    return total
```

Here is how you do it with recursion:

```
def factorial(n):
    if n == 1:
        return 1
    else:
        return n*factorial(n-1)
```

Recursive algorithms are more difficult to understand than regular ones, but can be much more powerful. Let's demonstrate how to understand what a recursive algorithm does.

What happens when you execute command `factorial(1)`? The function returns 1.

What happens if you execute `factorial(2)`? The function will return `2*factorial(1)`, which is of course equal to 2.

What happens if you execute `factorial(3)`? The function will return `3*factorial(2)`, which we already know is the same as `3*2*factorial(1)`, equal to 6. Do you see why this works?

Now, what happens if you execute `factorial(-1)`? The function will return `-1 * factorial(-2)`, where `factorial(-2)` will be evaluated as `-2 * factorial(-3)`, where `factorial(-3)` will be evaluated as `-3 * factorial(-4)` . . . And so this recursion will continue forever (or, rather, until your compiler/interpreter will give up and say "maximum recursion depth exceeded")

Example 2

Compute x to the power of n:

```
def power(x, n):  
    if n == 0:  
        return 1  
    else:  
        return x * power(x, n-1)
```

5 points:

Write a recursive function `bumps(n)` that produces the following output:

`bumps(1)` :

.

`bumps(2)` :

.

..

.

`bumps(3)` :

.

..

.

...

.

..
.
bumps (4) :
.
..
.
...
.
..
.
....
.
..
.
...
.
..
.

Etc for any positive integer n .

10 points:

Find the maximum number in an array of integers using recursion.