

**PROBLEM OF THE  
MONTH**



**September, 2016**

### **MATHEMATICS**

**5 points:** 16 participants have signed up for the Sigma ping pong tournament. How many ways are there to pair these 16 participants in the first round?

**10 points:** Prove that the product of any  $m$  subsequent integer numbers is divisible by  $m!$

## PHYSICS

**5 points:**



Two very massive walls are moving toward each other with identical speeds  $V$ . A light box is initially resting on a frictionless surface between the two walls as shown. Find the speed of the block after three elastic collisions with the walls.

**10 points:**

A spring is hanging from the ceiling of an elevator. A technician hung a 1 kg weight onto the spring, and after it reaches equilibrium, measured that the spring stretched by 10 cm. The technician then pressed the 50th floor button, and the elevator suddenly started moving up with the constant acceleration of  $1 \text{ m/s}^2$  and continues to move with the same acceleration. What will be the maximum additional stretching of the spring (assume that all processes are completely elastic)?

## CHEMISTRY

### 5 points:

Once upon a time, in a small tavern in Port-Royal one old pirate John Neusilber was telling a fascinating story of his recent enterprise, when he, along with a small crew of his friends, managed to capture a huge Spanish galleon. There they found several heavy chests full of golden coins. John was boasting how brave his crew was, and how cowardly the crew of the Spanish ship surrendered.

“Believe me, this story is absolutely real. I even have some coins with me,” John said and pulled out a small sack with coins from his pocket.

“Something is very suspicious in this story,” another pirate, Prudentio, said. “Why were there almost no guards on this ship, why did they surrender so easily? I am not sure these coins are golden. Let’s ask Alberto.”

Alberto was a pirate too, but he started his career as an alchemist. After many years of tireless attempts to convert lead to gold, he eventually realized that to obtain gold from lead one has to load a lead bullet into musket’s barrel and to find someone who has no loaded musket, but who has a wallet loaded with golden coins. In other words, he left Europe, went to Caribbean and became a pirate.

Alberto took one coin and looked at it. “Interesting” - he said. “Looks like gold, but... Can I have one? I’ll tell for sure by tomorrow.”

After coming to his room, where he had a small alchemist laboratory (apparently, he was still hoping he would be capable of inventing the method of obtaining gold without bloodletting), Alberto took a bottle with *Spiritus of Niter* (modern chemists call it “nitric acid”). Alberto initially planned to prepare *Aqua Regia* (a mixture of nitric and hydrochloric acids), but at the very last moment he decided to take *Spiritus of Niter* alone. He poured it in a glass and dropped the coin there. To his surprise, a violent reaction ensued, and the coin dissolved completely. The resulting solution had a green-blue color. “A-ha, not gold,” - Alberto said, “Let’s see what else is there besides Venus”. He poured the solution in a dish and started to heat gently until the solution evaporated completely, leaving a crystalline

residue. He then dissolved this residue in water and added a concentrated solution of table salt to it. A copious white precipitate formed immediately, which was not soluble neither in *Spiritus of Niter* nor in *Aqua Regia*. "Not bad", Alberto said, "not bad..."

Next evening Alberto announced his verdict to impatient pirates: "Prudentio was right, the coins are not golden. That is why there were almost no guards there. But I found something Spaniards themselves didn't probably know: these coins contain a lot of ....." "

What did Alberto mean under "Venus", and why did he decide the coins are not a complete junk?

**10 points:**

Below are fragments from a textbook written in some extinct language. The Cryptography department was capable of translating the text partially, but some words, symbols, and formulas remained unclear, because the cryptographer who did that work had almost zero knowledge of Chemistry. Decipher the unclear parts of the text.

*Seogulc* is a white crystalline compound with a formula  $\Psi_{\pi} E_{\xi} \mathcal{P}_{\pi}$ . It is non-toxic, soluble in water, and has a sweet taste. Its formula can also be written as  $(\Psi E_{\xi} \mathcal{P})_{\pi}$ , and that is why **Seogulc**, as well as other compounds of that type are called **glewarhyd** (a word composed of "glew",  $\Psi$ , and **arhyd**,  $E_{\xi} \mathcal{P}$ ).

.....

In free form, **glew**, or  $\Psi$ , is a solid that burns in gaseous  $\mathcal{P}_{\xi}$  to produce  $\Psi \mathcal{P}_{\xi}$  (a gas) according to the equation:



.....

$P_{\zeta}$  is a gas. Its mixture with gaseous  $\mathcal{P}_{\zeta}$  burns violently to yield **arhyd**, and the equation of this reaction is:



At room temperature, **arhyd** is a colorless transparent liquid. It reacts violently with some active metals (calcium, sodium), and, in a presence of gaseous  $\mathcal{P}_{\zeta}$ , causes corrosion of many other metals (e.g., iron, copper). It is used as a solvent for many inorganic reactions.

Explain what “seogulc”, “arhyd”, and “glew” mean. Explain the meaning of the symbols  $\Psi$ ,  $\mathcal{P}$ ,  $P$ ,  $\mathcal{L}$ ,  $\zeta$ , and  $\pi$ .

## **BIOLOGY**

### **5 points:**

It is known that in all living organisms three nucleotides are used to encode each amino acid. Does it mean that to encode the peptides with the total length of  $N$  amino acids the DNA segment must be at least  $3N$  nucleotides long? Are there any exceptions to this rule, and if yes, what are the benefits and disadvantages of that?

### **10 points:**

The gastric brooding frog (*Rheobatrachus silus*), extinct since 1981, swallowed its eggs and then brooded about twenty tadpoles for 6-7 weeks in its stomach. Ultimately, the mother "give birth" by burping up her froglets. If the species evolved from a more conventional type of frog, as seems likely, propose a possible mechanism that lead to the development of this "method of birth" and identify at least three changes that would have had to take place in order to make it 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, or as a zip archive of such files.

**No .pdf, .doc, .docx, etc!**

### Introduction: Multidimensional Arrays

*(If you already know what single-dimensional and multidimensional arrays are, you are welcome to skip straight to the problems below)*

As you know, in computer languages, values representing quantities, IDs, text, and so on, are stored in *variables*:  $a = 1$ ,  $b = 3.14$ ,  $c = \text{"Hello World!"}$ . If we want to store the year's precipitation by day in inches, it would be inconvenient to store the data in 365 variables:  $\text{day0} = 0.0$ ,  $\text{day1} = 0.5\dots$  Instead, we use a *one-dimensional array*: We *allocate* (meaning *reserve*) space for 365 variables in one long block of memory, and call the block **dailyPrecipitation**. To access the variable for each day, we write **dailyPrecipitation[i]**, where  $i$  goes from 0 to 364. So to write or read the value for day 54, we write **dailyPrecipitation[54]**, which means "look in the 54th block of the array". Behind the scenes, when the computer reads the expression **dailyPrecipitation[54]**, it goes to the beginning of this block of memory, then moves 54 blocks over, and reads the value there. *Be careful: although in most languages, the first block is numbered 0, there are some (rare) exceptions where labeling starts at 1.*

We can extend the notion of one-dimensional arrays to two dimensions. From the perspective of writing code, it's easy. Say, if you want to record the players' positions on a checkerboard, and denote each square as  $0 \rightarrow$  empty,  $1 \rightarrow$  black piece,  $2 \rightarrow$  red piece, then you create an  $8 \times 8$  *two dimensional array* of integers, call it, say, **checkerboard**.

Now, to write or read the state of square A8 (i.e. 1,8), we just write **checkerboard[0][7]**. Behind the scenes, things are slightly more complicated than in one dimensional case: Computer memory is single-dimensional, so it can only reserve 1-dimensional arrays. One way that a computer can create the 8x8 2D array **checkerboard[8][8]**, is to allocate  $8*8 = 64$  blocks, and then map the 2D array onto these 64 blocks like this:  
**checkerboard[n][m]** = look in the  $(8*n + m)$  block from the beginning of **checkerboard**. Naturally, it is possible to make arrays of any dimension this way.

Different languages have different syntax for arrays. Look up the array documentation for the language of your choice to learn how to implement them in your code.

### 5 points:

If numbers 1 through 9 are placed in the 3x3 grid so that sums of numbers in each row, column and two major diagonals are 15, they form a magic square. An example of magic square is:

```
2 7 6
9 5 1
4 3 8
```

Write a program where you receive first 2 rows of a square from standard input. Then your program needs to figure out whether third row can be added so that a magic square is formed. If it's possible, you should output the resulting magic square. If not possible, the program should state so. Use two-dimensional array (see *Introduction*) in your solution.

### 10 points:

#### Problem:

Your program receives from standard input an "image" of an arbitrary shape cut from an NxN square of quad-ruled paper (cutout follows grid lines). First the dimension N of the square is given. Then N lines of text are provided, where "-" is standing for "cutout", and "x" is standing for the shape. For example:

```

(0,0) (5,0)
-x-----
-xxx--
-xxx--
-xxx--
---x--
-----
(0,5) (5,5)

```

(Note: coordinates are provided here for clarity, they are not included in the input).

Your program should:

1. Find the center of mass of the shape. For the purpose of the calculation of the center of mass, presume that each little square "x" of the shape has the same mass. In the example above the center of mass is at (2, 2).

Note: it is possible for the center of mass to be located between the coordinate lines. For example, the following shape

```

----
-xx-
-xx-
----

```

has a center of mass located at (1.5, 1.5). Your program should take this into consideration.

2. Determine if the shape is contiguous (i.e. one piece). For the shape to be contiguous its constituent little squares should touch each other horizontally, vertically or diagonally. For example, the following shape:

```

----
-x--
--x-
----

```

is contiguous, but this one:

----

-x--

----

-xx-

is not.