PROBLEM OF THE MONTH

December, 2019

2019 - 2020

MATHEMATICS

5 points:

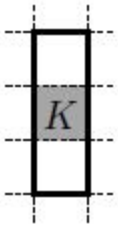
With a pen, Alex drew a rectangle on the lines in his quad ruled notebook. Using a pencil, he successfully divided the rectangle into 3x1 rectangles. Once again with a pen, he colored in the middle square of each 3x1 rectangle. He then erased all the penciled lines. Is it always possible to recover the initial division into 3x1 rectangles?

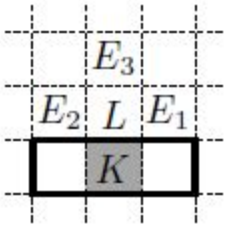
Hint: Let's assume it's not possible to recover the initial division into 3x1 rectangles. This means that there must be at least one colored square that represents a horizontal 3x1 rectangle in one possible division and a vertical rectangle in another. Let's take the highest such colored square. What does this mean for the (non-colored) square directly above it?

Answer: Yes

Solution:

Suppose There are two divisions of the large rectangle into 3x1 rectangles for which the colored squares match. Then, there must be at least one colored square that corresponds to a vertical rectangle in one division and a horizontal one in the other. Let's take the highest such colored square and call it *K*. Let's consider the square directly above that, which we'll call *L*. In the case that *K* is covered by a horizontal 3x1 rectangle, *L* must be part of a different 3x1 rectangle, so one of the squares adjacent to it that aren't *K* must be colored. But this is a contradiction because all of those squares are higher than *K*, but we assume *K* was the highest. Therefore, it's impossible to have one coloring correspond to more than one division into 3x1 rectangles, thus the original division is always recoverable.





Problem source: Moscow Olympiad from 2016

10 points:

In triangle ABC angle A is 60 degrees. Let BB_1 and CC_1 be two angle bisectors of this triangle.

a) (2 points) Prove that the quadrilateral AB_1IC_1, where I is the intersection of the angle bisectors, can be inscribed in a circle.

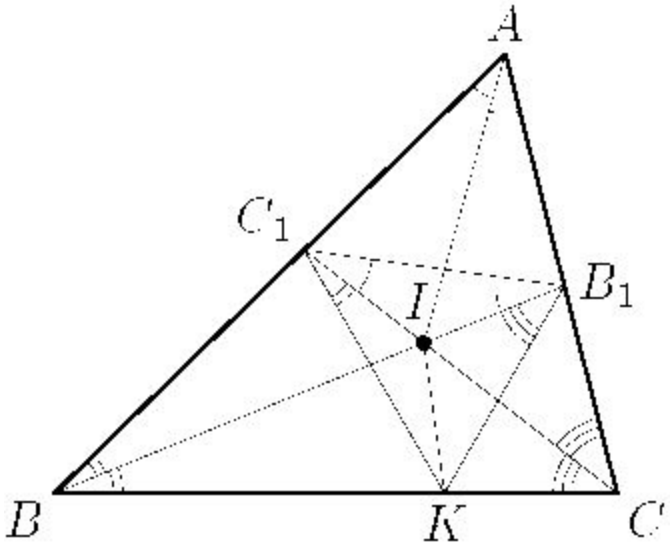
b) (8 points) Show that the point symmetric to A relative to the line B_1C_1 lies on the side BC.

Hint: Google theorems about inscribed angles in a circle.

Hint:

Answer: see the proof

Solution:



This solution frequently uses the fact that an angle inscribed in a circle equals half of the measure of the intercepted arc. Due to this, in order for a quadrilateral to be inscribed in a circle, its opposite angles must add up to 180 degrees.

Let point I be the intersection point of the angle bisectors of triangle ABC. Then $B_1IC_1 = \angle BIC = 180^o - \angle IBC - \angle ICB = 180^0 - (\angle ABC + \angle ACB) / 2 = 180^o - 60^o = 120^o = 180^o - \angle B_1AC_1$. Therefore, the quadrilateral AB_1IC_1 can be inscribed in a circle. From here, $\angle AC_1B_1 = \angle AIB_1 = \angle ABI + \angle BAI = (\angle A + \angle B) / 2$ (because $\angle AIB_1$ — is external in ΔABI), and analogously $\angle AB_1C_1 = (\angle A + \angle C) / 2$. Let the circumscribed circle around triangle BC_1I intersect the line BC at point K (it's easy to see that this point can't fall on the continuation of side BC). Then $\angle IKC = 180^o - \angle BKI = \angle BC_1I = 180^o - \angle AC_1I = \angle AB_1I = 180^o - \angle IB_1C$, so the quadrilateral

IB_1CK can also be inscribed in a circle. Finally, because the quadrilaterals AB_1IC_1 , BC_1IK и $CKIB_1$ are inscribed, we have $\angle KC_1B_1 = \angle KC_1I + \angle IC_1B_1 = \angle KBI + \angle IAB_1 = (\angle B + \angle A) / 2 = \angle AC_1B_1$, and analogously $\angle KB_1C_1 = \angle KB_1I + \angle IB_1C_1 = (\angle C + \angle A) / 2 = \angle AB_1C_1$. Therefore, triangles AB_1C_1 и KB_1C_1 are congruent from side B_1C_1 and two of its adjacent angles (angle-side-angle congruence law). This means they are symmetrical relative to B_1C_1 , therefore points A and K are also symmetrical. Because point K lies on BC , the solution is complete.

PHYSICS

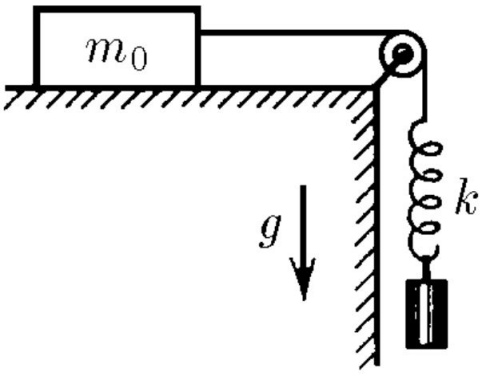
5 points: The scales are based on a spring with spring constant *k=10000N/m* and are calibrated to show the weight in kilograms. Imagine now that you drop a rigid block of mass $M = 1kg$ from height $h = 1$ meters, right on those scales. What will be their maximum reading (in kg)?

Hint: Use conservation of energy to find the maximum spring deformation.

Answer: $\sqrt{2kMh/g} \approx 45kg$

Solution: At the moment when deformation of the spring (x) is maximal, all the original potential energy, Mgh, is completely converted to potential energy of the spring: $Mgh = kx^2/2$. (we used the fact that this deformation, $x = \sqrt{2Mgh/k} \approx 0.045m$ is much less than $h = 1m$). This means that the elastic force is $F = kx = \sqrt{2kMgh}$, which would correspond to the following reading of the scales: $m^* = F/g = \sqrt{2kMh/g} \approx 45kg$

10 points: The system consisting of two blocks, rope, spring, and pulley is shown in the Figure. The friction coefficient between the block of mass m_0 and the horizontal surface of the table is μ. The block attached to the spring is released while the spring is unstretched. What should be the minimum mass M of that block so that the block on the surface is displaced from its original position? Assume that the rope, the spring, and the pulley are weightless, there is no friction in the pulley and the rope is unstretchable.



Hint: See hint to 5 pt. Problem

Answer: $M = \frac{1}{2}\mu m_0$

Solution: The tension force, T, of the string attached to the block m_0 must overcome the friction force, $\mu m_0 g$, where $g \approx 9.8$ m/c^2 is the free fall acceleration. The tension force of the string equals that of the spring, which reaches maximum when block M is at its lowest point, with zero velocity, and the spring has maximum extension, x. This extension can be found from energy conservation, equating the change of potential energy of the block M with the energy of the stretched spring: $Mgx = \frac{1}{2}kx^2$. We thus have, $T = kx = 2Mg = \mu m_0 g$, and $M = \frac{1}{2}\mu m_0$.

CHEMISTRY

5 points:

Brass is an alloy of copper and zinc. Zinc content varies in different types of brass. To measure a composition of some brass sample, 10 grams of brass shavings were placed into a narrow neck flask, 100 mL of 20% HCl was added to it, and a rubber balloon was attached to the neck. When the chemical reaction has ceased, the volume of the balloon was measured. 2.24 L of gas formed in this process. Using this information, calculate the composition of the brass sample.

Hint:

Keep in mind that zinc reacts with HCl to form a hydrogen gas and a water soluble salt (zinc chloride), whereas copper does not.

Solution:

Zinc reacts with HCl according to the equation:

$$Zn + 2HCl \rightarrow ZnCl_2 + H_2$$

That means one mole of hydrogen (or 22.4 L of a hydrogen gas at normal temperature and room pressure) is formed when one mole, or 65 g of zinc reacts. It is easy to see that 2.24 L of hydrogen is produced by 0.1 mole of zinc (or by 0.2 mole of HCl). Clearly, 100 mL of 20% HCl contain more than 0.2 mol of HCl (100 mL of 20% solution contain 20 g of HCl, which is about 20/36.5 = 0.55 mols). Therefore, we have an excess of HCl, and the amount of zinc in the sample was 0.1 mole, or 6.5 grams. Accordingly, there are 65% of zinc in that brass sample.

10 points:

Each year in SigmaCamp, Mark is having the same problem: how to organize the experimental part of the tournament, keeping in mind that (i) the tournament's participants are not supposed to operate with hazardous materials, and (ii) the majority of interesting chemical experiments use some dangerous reactants. Imagine that during one SigmaCamp, Mark managed to collect a set of pretty harmless chemicals that included:

1. Sodium hydrosulfate;
2. Potassium bicarbonate;
3. Magnesium shavings;
4. Calcium chloride;
5. Copper wire;
6. Alkaline batteries;
7. Sodium carbonate;
8. Quicklime (calcium oxide);
9. Aluminium foil;

and some glassware (test tubes, flasks, beakers, etc), scales, other simple chemistry lab equipment, as well as various stuff, such as ropes, balloons, scotch tape etc..

Propose an experimental problem for the tournament that uses only chemicals from the above list.

Hint:

No single correct answer exist. At least three different experiments can be proposed. Also, keep in mind that some compound from this list is a moderately strong acid, whereas another one is a source of a relatively strong base.

Solution:

Among experimental problems that can be proposed are, for example:

- Make a battery from the following materials: magnesium shavings, copper wire, calcium chloride, sodium carbonate, and potassium bicarbonate (actually, the last two chemicals are not needed);
- Make hydrogen using aluminium foil, calcium chloride, potassium hydrosulfate and sodium carbonate (that could be done in two different ways: by reacting aluminium foil with potassium hydrosulfate, which is a moderate acid (HSO_4^- is moderately acidic), or by making NaOH from quick lime and sodium carbonate:

  $$CaO + H_2O + Na_2CO_3 \rightarrow CaCO_3 + 2NaOH$$

  and then by reaction of the NaOH obtained with aluminium:

  $$2Al + 6\ NaOH \rightarrow 2\ Na_3AO_3 + 3\ H_2\ )$$
- Make as much CO_2 as possible (that can be done keeping in mind that potassium hydrosulfate is an acid, so by mixing it with sodium hydrocarbonate one can easily obtain CO_2:

  $$KHSO_4 + NaHCO_3 \rightarrow KNaSO_4 + CO_2 + H_2O)$$

These are just some examples of problems that can be proposed, we are sure you will propose many other interesting problems.

BIOLOGY

5 points:

In living cells, DNA is being copied by an enzyme called DNA polymerase. This enzyme adds, step by step, nucleotides to a growing chain of DNA according to well-known rules: A is added opposite T, G opposite C, etc. Once a nucleotide has been added, the DNA polymerase molecule shifts forward, and the next nucleotide is added. As a result, DNA polymerase is crawling along the template DNA strand with an approximate speed of 1000 nucleotides per second. How long a division of a single procaryotic and/or mammalian cell would take if only one DNA polymerase molecule were present in the cell?

Hint:

Your answer will be somewhat counter-intuitive, but, nevertheless, it might be correct, because DNA copying in a living cell never occurs in the way described in this problem. In a real living cell, a large number of DNA polymerase molecules are involved in DNA replication, and they are doing that concurrently.

Answer:

Actually, this problem is similar to a simple Math/Physics problem like “A car is moving with the speed of X km/h. What time is needed for the car to travel from the city A to the city B if the distance is Y km?” Assuming that the size of an average mammalian chromosome is 1 billion base pairs (1 gigabase), it is obvious that for a polymerase molecule that is moving forward by 1000 bases per seconds, about a million of seconds (or 1000000/60/60/24 = 12 days) is needed to copy one chromosome. That answer looks weird, because it implies that division of a mammalian cell requires at least two weeks. In reality, DNA copying is a parallel process, which is performed by many DNA polymerases. They are working concurrently, and each of them starts DNA copying from some specific positions that are scattered across a chromosome, which are called *origins of replication.* That is why DNA replication occurs much, much faster. Procaryotic (bacterial) genome is much smaller (several millions base pairs), so it takes much less time to copy it. However, bacteria are dividing very fast, so they also have a parallel DNA copying mechanism to provide quick DNA replication.

10 points:

The Nobel Prize in Physics for 2018 was awarded to Arthur Ashkin for his Optical Trap (OT).

Why did this Physics discovery became most important for the development of Molecular Biology?

What is it about the biological molecules that we can see with Optical Trap that we would not be able to see otherwise?

The Optical Trap is able to apply stretching forces to the individual single folded biological molecules (proteins or nucleic acids, i.e. DNA and RNA), and also is able to simultaneously measure the extension between two attachment points of the molecule. Using the conventional laws of mechanics, what can you learn about particular biological molecule? Give at least one example.

Hint 1: The conventional biochemical methods observe the average behavior of the huge number of molecules. However, biological molecules often have alternative conformations that perform specific functions. This is especially true of the molecular machines: molecules that walk, carry cargo, polymerize or unwind double stranded DNA, etc. What can the Optical Tweezers (version of Optical Trap used in molecular biology) do to tell us more about each molecule? Why is this instrument called a Tweezers? Please, describe and explain one good example of the biological molecule that people learned a lot about using Optical Tweezers.

Hint 2: The free energy of the particular folded structure of the biological molecule can be estimated as a product of the measured transition force and extension for a particular

force-induced transition. The examples may include unfolding of RNA or protein structures, force-induced melting of the double stranded DNA, unwrapping the DNA from nucleosome, etc.

Answer:

Optical Trap invented by Ashkin is a way to hold and move micron sized objects by the two crossing laser beams. The laser beams crossover point is a sub-micron size potential well for the bead of the similar size made of the low dielectric constant materials. As all organic molecules and most polymers are of low dielectric constant they can all be trapped. By changing the position of that potential well one can optically move the bead, or even the organelle within the live cell. For this reason, the Optical Trap is also called an Optical Tweezers (OT).

Development of OT became so important for molecular biology, because with it people acquired an ability to study the mechanical properties of the individual biological molecule, such as protein, DNA or RNA, one at a time. The way to do it is to attach a single biomolecule at its two different points to the two micron-sized objects that can be independently manipulated by the OT (or one point can be attached to the surface or micropipette). The instrument can simultaneously measure the distance between the two attachment points and the force that it takes to stretch it to that distance. Multiplying force and extension we can estimate the work or an energy that it takes to make this change in the molecule. This is like studying the static and dynamic properties of the conventional large objects. For example, before any car gets on the road it is subjected to many mechanical tests.

This is a great step for molecular biology, as previously people were only able to measure the average statistical behavior of many, i.e. ~Avagadro number ~$6 \cdot 10^{23}$ per Mole of molecules. As almost all biomolecules have different states corresponding to their different functions, these states are always averaged and missed in such measurements.

Studying the single bio-molecules with OT we can find out about their different stable conformations, how fast the molecule can switch between these states, how these states can be controlled by other molecules binding. We can also study unfolding and re-folding of the complicated 3D structures of proteins and nucleic acids, mimicking the biological processes happening to them during their lives. This is how muscle proteins are often studies. Also, molecular machines that make new DNA and RNA (polymerases), unwind DNA (helicases), untangle the double stranded DNA (topoisomerases), that carry cargo (kinesines), etc., are all studies with OT. Anything that binds, moves, changes shapes, consumes energy can be studied. Obtained data is quantitative, describes life of a single molecule, and can be interpreted in terms of free energies of different molecular conformations.

About half of the biophysical labs all over the world now study single biomolecules using OT.

There are infinite number of examples of the OT being used to find out important biological information. For example, as a stretched single stranded (ss) DNA is longer than double stranded (ds) DNA, the polymerase making a second strand on the ssDNA template leads to the whole DNA molecule shortening. Watching DNA polymerization with OT people were able to measure the rate of a single DNA polymerase (DNApol), its pausing probability, DNApol binding and unbinding rates, effects of other protein members of the DNA replication fork movement, etc. This makes it possible to make detailed “cartoons” of the functioning of the DNA replication fork, such as this one:
https://www.youtube.com/watch?v=4jtmOZaIvS0

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!**

5 points:

For a given finite sequence of N integer numbers a_1, a_2, … a_N, let's define a reverse-weighted sum (RWS) as

RWS = $\sum_{i=1}^{N}(N+1-i)*a_i$

Write a program that receives on input:

- a list of integers a_1, a_2, … a_N

- a target integer T

The program needs to find an integer x that will when inserted at some position in the list will result in RWS of the list being equal to T. Print the list with x inserted at the desired location, or, if such an operation is impossible, indicate so.

Example:

Initial list is 1, 2, 3 and T=15. Output: x=1, updated list: 1, 2, 1, 3.

Hint:

If you have RWS for numbers a_1, a_2, … a_N, think how would it change if you insert b in k-th position.

Solution:

Java:

```
/*
So, our RWS is c[1]*a[1] + c[2]*a[2] + ... +c[N]*a[N], where c[i] = N+1-i
Let's insert a number x somewhere in this sequence, say, at the index k
(in the extreme case k may be equal to 1, when inserted to the left of the original sequence,
or N+1, when inserted to the right of it)
Assume that the original sequence sums up to t. Then all the coefficients to the left of k
will be increased by 1, and nothing changes to the right of the new item c[k]*x.
We want that the new sequence sums up to T. Therefore, we've got
  a[1] + a[2] + ... + a[k-1] + (N+1-k)*x = T - t
Hence x = (T - t - a[1] - a[2] - ... - a[k-1]) / (N + 1 - k)
If this results in an integer then we've found a solution.
Our algorithm will try all k from 1 till N+1.
Be careful as Java indexing starts with 0.
Note: the solution is not unique. Your may find a different one.
*/

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.util.Arrays;

public class RWS5 {
  private int[] a;
  private int T;
  private int N;

  public void input() throws IOException {
    System.out.print("Enter numbers: ");
    BufferedReader reader = new BufferedReader(new InputStreamReader(System.in));
    String line = reader.readLine();
    String[] nums = line.trim().split("\\s*[\\s,]\\s*");
    // the following will croak if the elements are not all integers
    a = Arrays.stream(nums).map(s -> Integer.valueOf(s)).mapToInt(Integer::intValue).toArray();

    System.out.print("Enter the target number: ");
    T = Integer.parseInt(reader.readLine().trim());

    N = a.length;
  }

  public int rws(int[] a) {
    int n = a.length;
    int s = 0;
    for(int i=0; i<n; i++)
      s += (n+1-(i+1))*a[i];
    return s;
  }

  public void solve() throws Exception {
    // calc t
    int t = rws(a);
```

```
    int x = 0;
    // try different k
    boolean found = false;
    int aa = 0; // our sum of a[i]
    int k;
    for(k=1; k<=N; k++) {
      if((T-t-aa) % (N+2-k) == 0) {
        found = true;
        x = (int)((T-t-aa) / (N+2-k));
        break;
      }
      else {
        aa += a[k-1];
      }
    }

    if(found) {
      System.out.printf("x = %d\n", x);
      int[] b = new int[N+1];
      for(int i=0; i<k-1; i++)
        b[i] = a[i];
      b[k-1] = x;
      for(int i=k; i<=N; i++)
        b[i] = a[i-1];
      if(rws(b) != T)
        throw new Exception("rws(b) != T");
      System.out.printf("updated list: %s\n", Arrays.toString(b));
    }
    else
      System.out.println("impossible");
  }

  public static void main(String[] args) throws Exception {
    RWS5 sol = new RWS5();
    sol.input();
    sol.solve();
  }
}
```

Python-3:

```
"""
So, our RWS is c[1]*a[1] + c[2]*a[2] + ... +c[N]*a[N], where c[i] = N+1-i
Let's insert a number x somewhere in this sequence, say, at the index k
(in the extreme case k may be equal to 1, when inserted to the left of the original sequence,
or N+1, when inserted to the right of it)
Assume that the original sequence sums up to t. Then all the coefficients to the left of k
will be increased by 1, and nothing changes to the right of the new item c[k]*x.
We want that the new sequence sums up to T. Therefore, we've got
  a[1] + a[2] + ... + a[k-1] + (N+1-k)*x = T - t
Hence x = (T - t - a[1] - a[2] - ... - a[k-1]) / (N + 1 - k)
If this results in an integer then we've found a solution.
Our algorithm will try all k from 1 till N+1.
Be careful as Python indexing starts with 0.
Note: the solution is not unique. Your may find a different one.
"""
```

```
numbers_str = input("Enter numbers: ").strip().split()
# the following will croak if the elements are not all integers
a = [int(x) for x in numbers_str]
numbers_str = input("Enter the target number: ").strip()
T = (int(numbers_str))

N = len(a)

def rws(a):
  n = len(a)
  s = 0
  for i in range(n):
    s += (n+1-(i+1))*a[i]
  return s

# calc t
t = rws(a)

# try different k
found = False
aa = 0 # our sum of a[i]
for k in range(1,N+1):
  if (T-t-aa) % (N+2-k) == 0:
    found = True
    x = int((T-t-aa) / (N+2-k))
    break
  else:
    aa += a[k-1]

if found:
  print("x = %d" % x)
  b = a.copy()
  b.insert(k-1, x)
  assert(rws(b) == T)
  print("updated list: ", b)
else:
  print("impossible")
```

10 points:

Alice and Bob are playing a collaborative video game. They are given a target number **T** (integer). Each of them can press the UP or DOWN arrow, and upon each press their current Collaborative Number goes up (or down) by **a** for Alice's presses and **b** for Bob's. The game starts with the Collaborative Number being 0. Write a program to help Alice and Bob reach their target number T. Your program should take integers a, b and T on input, and it should print the number of up or down presses for each of Alice and Bob to win the game. If the target can't ever be reached, it should print so.

Example:
a=2 b=3 T=10
Output: Alice presses UP 2 times, Bob presses UP 2 times.

Hint:

Try to express the task in mathematical terms. Then look up algorithms for solving it.

Solution:

Java:

```
/*
If we use x for Alice's number of presses and y for Bob's, then we can say that
  a*x + b*y = T,
where a,b,x,y,T are all integers. This is a linear Diophantine equation
(https://en.wikipedia.org/wiki/Diophantine_equation)
It has a solution if T is a multiple of GCD of a and b.
The solution and explanation can be found here:
https://www.math.utah.edu/~carlson/hsp2004/PythonShortCourse.pdf (page 11)
Note: the solution is not necessarily unique, and we are not tasked with finding the smallest
number of presses.
*/

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;

import static java.lang.Math.abs;

class Tuple {
  public double _1, _2;
  public Tuple(double x, double y) {
    _1 = x;
    _2 = y;
  }
}

public class Diophantine10 {
  private int a, b, T;

  public void input() throws IOException {
    BufferedReader reader = new BufferedReader(new InputStreamReader(System.in));
    System.out.print("Enter Alice's Collaborative Number: ");
    a = Integer.parseInt(reader.readLine().trim());
    System.out.print("Enter Bob's Collaborative Number: ");
    b = Integer.parseInt(reader.readLine().trim());
    System.out.print("Enter the target number: ");
    T = Integer.parseInt(reader.readLine().trim());
  }

  public Tuple isolve(int a, int b, int c) {
    int q = a / b;
    int r = a % b;
    if(r == 0)
      return new Tuple(0, (double)c/b);
    else {
      Tuple sol = isolve(b, r, c);
      double u = sol._1;
```

```
      double v = sol._2;
      return new Tuple(v, u - q * v);
    }
  }

  public void solve() {
    Tuple s = isolve(a, b, T);
    double x = s._1;
    double y = s._2;
    if(x%1 != 0 || y%1 != 0)
      System.out.println("impossible");
    else
      System.out.printf("Alice presses %s %d times, Bob presses %s %d times\n",
                (x >= 0) ? "UP" : "DOWN", abs((int)x), (y >= 0) ? "UP" : "DOWN", abs((int)y));
  }

  public static void main(String[] args) throws Exception {
    Diophantine10 sol = new Diophantine10();
    sol.input();
    sol.solve();
  }
}
```

Python-3:

```
"""
If we use x for Alice's number of presses and y for Bob's, then we can say that
  a*x + b*y = T,
where a,b,x,y,T are all integers. This is a linear Diophantine equation
(https://en.wikipedia.org/wiki/Diophantine_equation)
It has a solution if T is a multiple of GCD of a and b.
The solution and explanation can be found here:
https://www.math.utah.edu/~carlson/hsp2004/PythonShortCourse.pdf (page 11)
Note: the solution is not necessarily unique, and we are not tasked with finding the smallest
number of presses.
"""

# a = 2
# b = 3
# T = 10

numbers_str = input("Enter Alice's Collaborative Number: ").strip()
a = (int(numbers_str))
numbers_str = input("Enter Bob's Collaborative Number: ").strip()
b = (int(numbers_str))
numbers_str = input("Enter the target number: ").strip()
T = (int(numbers_str))

def isolve(a, b, c):
  q, r = divmod(a, b)
  if r == 0:
    return [0, c/b]
  else:
    sol = isolve(b, r, c)
    u = sol[0]
    v = sol[1]
    return [v, u - q*v]
```

```
sol = isolve(a, b, T)
if sol[0]%1 != 0 or sol[1]%1 != 0:
  print("impossible")
else:
  print("Alice presses %s %d times, Bob presses %s %d times" %
         ("UP" if sol[0] >= 0 else "DOWN", abs(int(sol[0])), "UP" if sol[1] >= 0 else "DOWN",
         abs(int(sol[1]))))
```

LINGUISTICS

5 points:

Once upon a time a large and friendly family from one Northern-European country gathered for a holiday dinner. Almost everyone from the family could make it, save for three people. In total, 16 people gathered at the table:

1) The head of the household Algirdas;
2) his wife Irma;
3) Irma's brother, Jonas;
4) Irma's sister, Iolanta;
5) Algirdas's sister, Lada;
6) Lada's husband, Giedrius;
7) Lada's and Giedrius's son, Juozas;
8) Lada's and Giedrius's daughter, Anna;
9) elder daughter of Algirdas and Irma, Grazina;
10) middle daughter of Algirdas and Irma, Rasa;
11) husband of the yourngest daughter of Algirdas and Irma, Rimas;
12) Rimas's brother, Edgaras
13) Rimas's sister, Elena;
14) Elena's husband, Aidas;
15) Rasa's daughter, Maria;
16) Rimas's daughter, Elzbieta.

The last names of the participants of the dinner are given below in a random order:
Jurenas, Šeštokas, Balsiene, Matulite, Balsyte, Matulis, Šeštokas, Ambraziene, Adomaytite, Jurenaite, Matulis, Adomaitis, Jureniene, Matuliene, Ambrazas, Šeštokaite

Question 1: Determine the last names of the following people and explain your reasoning:

a. *Algirdas*
b. *Jonas*
c. *Elena*
d. *Elzbieta*

Question 2: Determine the last names of the three people who did not participate in a holiday dinner and explain your reasoning

a. *Rasa's husband*
b. *Rimas's wife*
c. *Elena's and Aidas's daughter.*

Solution:

We can draw the family tree, and adopt the following assumptions: wives take last names of their husbands, and the last names of children have the same root as the last names of their parents. This way, all present family members are broken into 6 groups, corresponding to different roots of last names, of sizes 4, 3, 3, 2, 2, 2. The only root with 4 last names is *Matul-*, so we can deduce which group of family members have these last names.

Now, all male first names end in *-s*, so we can assume that last names of males also end in *-s*. There seems to be two different endings for female last names, we can assume that they correspond to married vs. unmarried women. Combining all of these together, we get the following answers.

Question 1: Determine the last names of the following people and explain your reasoning:

e. *Algirdas: Jurenas*
f. *Jonas: Adomaitis*
g. *Elena: Ambraziene*
h. *Elzbieta: Šeštokaite*

Question 2: Determine the last names of the three people who did not participate in a holiday dinner and explain your reasoning

d. *Rasa's husband: Balsas*
e. *Rimas's wife: Šeštokiene*
f. *Elena's and Aidas's daughter: Ambrazaite*

10 points:

Roman numerals are written in the following way:

1. 1, 2, 3, 4, 5, 6, 7, 8, 9 are I, II, III, IV, V, VI, VII, VIII, IX
2. 10, 20, 30, 40, 50, 60, 70, 80, 90 are X, XX, XXX, XL, L, LX, LXX, LXXX, XC
3. 100, 200, 300, 400, 500, 600, 700, 800, 900 are C, CC, CCC, CD, D, DC, DCC, DCCC, CM
4. 1000 is M

A number containing several decimal places is represented, as in the Arabic system, by writing its power-of-ten parts – thousands, hundreds, tens and units – in sequence, from left to right, in descending order of value. For example:

39 = 30 + 9 = XXX + IX = XXXIX.
246 = 200 + 40 + 6 = CC + XL + VI = CCXLVI.

789 = 700 + 80 + 9 = DCC + LXXX + IX = DCCLXXXIX.
2,421 = 2000 + 400 + 20 + 1 = MM + CD + XX + I = MMCDXXI.

Any missing place (represented by a zero in the Arabic equivalent) is omitted, as in Latin (and English) speech:

160 = 100 + 60 = C + LX = CLX
207 = 200 + 7 = CC + VII = CCVII
1,009 = 1,000 + 9 = M + IX = MIX

Question: provide a formal algorithm that allows one to add two Roman numerals (from 1 to 2,000) without converting them to decimal system. You can use any operations (substitution, order change, rewriting, etc), and anyone should be able to execute them, without knowing anything about Roman numerals.

Solution:

Obviously there can be several solutions to this problem. One potential one is the following:
1). Let's get rid of all subtractions first: Change IX -> VIIII; IV -> III; XC -> LXXXX; XL -> XXXX; CM -> DCCCC; CD -> CCCC in both of the numbers. The order of substitutions doesn't matter.
2). Now, take "digits" of both of the numbers, and write them by groups, in the following order: M...D...C...L...X...V...I…
3). Now, perform the following substitutions in the following order, **from the left of the string**: IIIII -> V; VV -> X; XXXXX -> L; LL -> C; CCCCC -> D; DD -> M.
It is important to do such substitutions from the left, so that IIIIII -> VI and not IIIIII -> IV
4). Now, so the substitutions opposite of the ones in Step 1)., in an order specified there. Order now is important, so that we get VIIII -> IX and not VIIII -> VIV