PROBLEM OF THE MONTH

POM

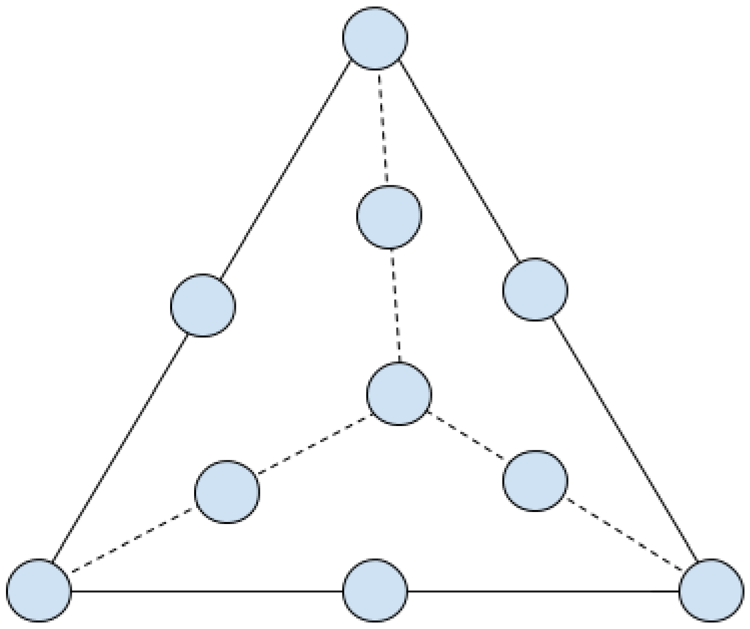
2019 - 2020

September, 2019

MATHEMATICS

5 points:

There are 10 small spheres on a tetrahedron, as shown below. Is it possible to label each sphere with a different digit from 0 to 9 such that the sum of the three digits on each of the six edges is the same?



Hint: Try to compare the total sum of the digits and the total sum of the edges.

Answer: No

Solution:

Suppose that it is possible. Let's say that the sum of the numbers at the vertices is A, the sum of the numbers in the middle of an edge is B, and the sum of the three numbers along each edge is C. It is clear that A + B = 45, because that is the sum of the digits from 0-9. Every sphere on a vertex belongs to three edges, whole the spheres in the middle of an edge belong only to that edge. Therefore, when adding up the sum of all six edges, we get 3A + B = 6C. From here, 2A = 6C – (A + B) = 6C – 45. This is a contradiction because the right hand side of the equation is even and the left hand side is odd, therefore it is not possible to label the spheres this way.

10 points

Alex cut a rectangular sheet of paper along a straight line. Then, he cut one of the two resulting pieces along a straight line. Then, he did the same to one of the resulting three pieces, and so on. Prove that after enough such cuts it will be possible to find, among the resulting pieces of paper, 100 polygons with the same number of vertices (for example, 100 triangles, or 100 quadrilaterals, etc).

Hint

In order to make a polygon with one more side than the original, it is necessary to produce a triangle. For example, to create a pentagon by cutting a quadrilateral in two pieces, the second piece must be a triangle.

Answer: No

Solution:

Upon Alex's cuts, he only obtains convex polygons. Upon cutting a triangle in two, at least one of the pieces is will also be a triangle, so the number of triangles doesn't decrease. A single cut can increase the number of sides in the polygon only by 1, and in order to do so you must create a triangle. If the number of sides is increased 100 times, we will get 100 triangles. Therefore the maximum number of sides that every obtained polygon has is 103, and there can be no more than 100 types of polygons. By the pigeonhole principle, once Alex gets to 9901 pieces, there must be at least 100 types of at least one polygon, since 9901/100 > 99.

PHYSICS

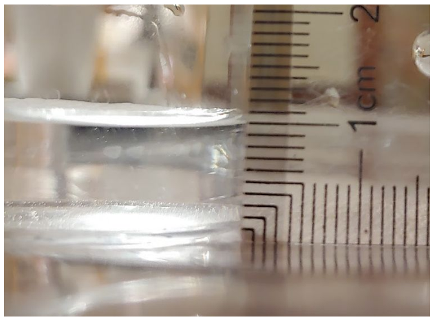
5 points:

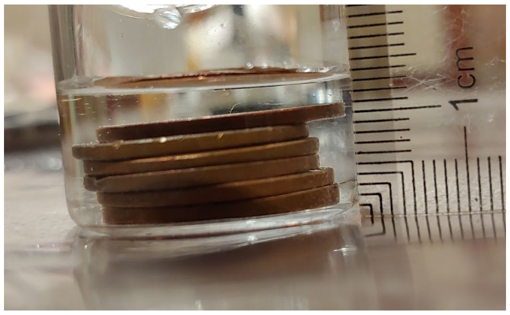
The United States one-cent coins aka pennies, minted before and after 1982 are made of substantially different materials. Design and carry out an experiment in which you measure one physical property of the material from which pennies of each type are made to establish that the materials are indeed different. A physical property of the material does not include the variation in the shape of coins and their appearance due to wear and tear. Examples are density, heat capacity, thermal expansion, electrical conductivity.

Hint: The easiest experiment is to measure density. You may use several coins of the same type to improve accuracy.

Answer: One can measure density. It is supposed to be $7.2\, g/cm^3$ for new pennies and $8.9\, g/cm^3$ for older one. Depending on the method used, the result may be off by up to 1.5 g/cm^3

Solution: In order to measure density with high accuracy, one can weight as many pennies of particular type as possible. The result should be about 2.5 g for new pennies and 3.0 for older ones (with accuracy about 0.1 g). To find the volume, one can use two methods:(1) construct a measuring cylinder and find the amount of displaced water by a bunch of pennies. (2) measure diameter and thickness of the penny and calculate is volume. To measure thickness one needs to put many pennies on top of each other. The result for method (2) will be about $d = 1.9\ cm$ and $h = 0.14\ cm$, which gives the volume of a coin $V = \pi(d/2)^2 h = 0.40 cm^3$. Therefore the density determined by method (2) is $6.3\ g/cm^3$, and $7.5\ g/cm^3$ for new and old pennies respectively. These results are smaller than actual densities ($7.2\ g/cm^3$ and $8.9\ g/cm^3$), due to engraving on the coin: the volume is overestimated. Method (1) can potentially give a better measurement, if carried out carefully. In the example shown in the photo, the water level in a measuring glass of 27(1) mm inner diameter changes by approximately 4 mm (the accuracy of visually determining this change is about 0.5mm, i.e. 12%). The resulting measured densities are $6.6\ g/cm^3$ for the older coins and $7.9\ g/cm^3$ for the newer coins, matching the nominal values within the accuracy of our measurement. Still both methods allow you to





distinguish between old and new coins

10 points:

A ball is dropped with no initial speed from the height h above the plate. The plate is moving upward with constant speed u. Find the time interval between subsequent collisions of the ball with the plate. Neglect air resistance and assume that collisions are absolutely elastic. Acceleration of gravity is g.

Hint: Switch to the reference frame of the plate.

Answer: $T = \frac{2\sqrt{u^2+2gh}}{g}$

Solution: In the reference frame of the plate, the initial speed of the ball is u (moving down), and total energy (Kinetic+Potential) is $\frac{mu^2}{2} + mgh$. By using the conservation of energy, we can find the speed v of the ball when it hits the plate: $\frac{mu^2}{2} + mgh = \frac{mv^2}{2}$, i.e. $v = \sqrt{u^2 + 2gh}$. The time between two subsequent collisions can be found as a time needed to change the velocity from +v (directed up), to -v (directed down), given that the acceleration is -g:

$$T = \frac{2v}{g} = \frac{2\sqrt{u^2+2gh}}{g}.$$

CHEMISTRY

5 points

During Sigma2019, in Fire Chemistry semilab (http://sigmacamp.org/2019/semilabs/fire) , Mark decided to prepare an aqueous solution of silver nitrate ($AgNO_3$). To this end, he asked Luigia, his assistant, to bring some water. He poured water into a test tube and added few crystals of silver nitrate. He expected to obtain a clear solution, but to his surprize, he got the cloudy solution shown in this photo:
https://photos.sigmacamp.org/2019/Day-5-August-15/i-wQLgCNm/A
(the tube on the right, the one Tatiana is holding). He immediately realized his mistake, and asked Luigia: "Please, bring me XYZ water". After Luigia did what Mark asked, he prepared another solution, and it was clear and colorless, as he expected (this solution is shown in the test tube on the left, the one Sophia is holding).

What was the word "XYZ", and why did Mark initially get a cloudy solution?

Hint:

Tap water, especially in Connecticut area, has some admixtures.

Solution:

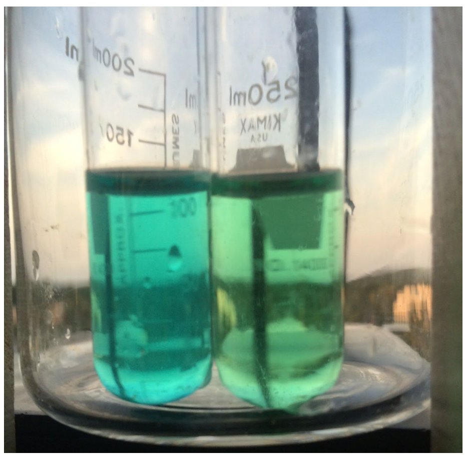
Usually, tap water contains some amount of soluble salts, normally, calcium hydrocarbonatem $CaHCO_3$ and magnesium hydrocarbonale $MgHCO_3$, as well as their chlorides,$CaCl_2$ and $MgCl_2$. When a concentration of these salts is high, such water is called *hard water*. In Connecticut, tap water is hard, which means there is a lot of chlorides there. Silver nitrate reacts with soluble chlorides to form an insoluble silver chloride, for example:

$$2AgNO_3 + CaCl_2 \rightarrow 2AgCl + Ca(NO_3)_2$$

Silver chloride is a white solid, so the liquid Tatiana is holding is turbid due to formation of fine particles of AgCl.

10 points:

Aqueous solutions of both cupric chloride ($CuCl_2$) and nickel chloride ($NiCl_2$) are green, and it is not easy to discriminate them based on their color (see the photo below):



Suppose you have some green solution, and you are not completely sure if it is copper or nickel chloride. How can you figure out what it is if, besides that unknown solution and distilled water, you have just test tubes, and you can spend for chemicals:

- just $7?

- just $11?
- just $23?

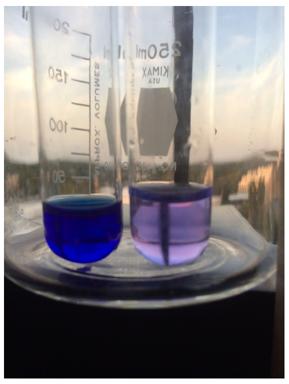
(shipping cost is not included)
Explain your answer bearing in mind that you are only allowed to use chemicals that are sold by regular suppliers to ordinary customers.

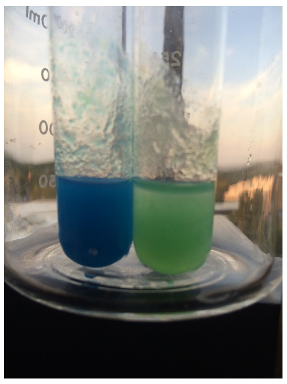
Hint:

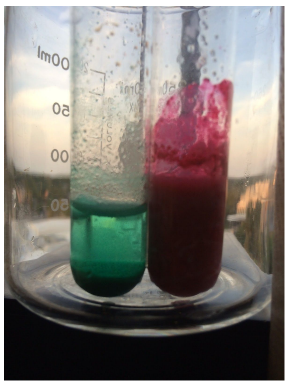
There are two possibilities to check that: to add some chemical that may produce different color with nickel and copper ions, and to capitalise of the fact that copper and nickel have different electrochemical properties (see metal activity series)

Solution:

This problem can be solved in two ways. First, we can use some color reactions. When we add some specific chemicals, copper and nickel can form new compounds with different colors.


A


B


C

Thus, if you add ammonia (it costs less than $7 at Amazon) to copper and nickel chloride, the color changes to deep blue and light blue, accordingly (Fig A.). If you add a sodium hydroxide solution (sodium hydroxide costs ~$10), to copper and nickel chlorides, the precipitates form in both case, but copper chloride produces a deep blue precipitate, whereas nickel chloride forms a light green precipitate (Fig B). If you can afford to spend around $23, you can buy a compound called dimethylglyoxime. It's second name is Chugaev's regent, and it forms a deep red solid in a reaction with nickel salts, whereas most other metals form much less interesting colors (Fig C). In all three cases the test tube with nickel chloride are on the right, and the test tube with copper chloride is on the left.
Chemical reactions in A and C cases are pretty complex, you can try to google them by yourself. The reaction with sodium hydroxide is simple, it is just an exchange reaction where a deep blue copper hydroxide and light green nickel hydroxide form:

$CuCl_2 + 2\,NaOH \rightarrow Cu(OH)_2 + 2\,NaCl$
(blue)

$NiCl_2 + 2\,NaOH \rightarrow Ni(OH)_2 + 2\,NaCl$
(light green)

The second method of discrimination of nickel and copper salts is a replacement reaction of that type:

$CuCl_2 + Fe \rightarrow FeCl_2 + Cu$

This reaction occurs when some more active metal comes to a contact with a salt of some less active metal. Activity of metals (its short version) looks like this:

More active → Mg Al Zn Fe Ni Sn Pb Cu Hg Ag Au Pt ← Less active

Clearly, copper is much less active than iron, so if you put an iron nail into a solution of copper chloride, you will see a fast formation of deep-red copper powder. In contrast, nickel (Ni) is very close to iron (Fe) in the reactivity series, so the reaction

$NiCl_2 + Fe \rightarrow NiCl_2 + Cu$

Will go very slowly. Actually, we will hardly see any reaction.
You can buy a nail for much less that $7, so that is probably the cheapest way to discriminate copper and nickel chloride.

BIOLOGY

5 points

Recent comparative studies of wolves and dogs demonstrated that dogs have a special muscle raising the inner eyebrow, whereas this muscle is almost absent in wolves. Is that difference just a collateral effect of domestication, or this muscle plays some important role? Explain your answer.

Answer:

Hint:

Some dogs look more cute than others. Try to think why.

Solution:

Dogs evolved from those wolves, who changed their normal habits and started to live near humans. That provided them with a constant source of food and provided them protection. Their survival became possible due to two important traits that they acquired during evolution: low aggression to humans and a capability to establish an emotional contact with them. Some scientists think we can speak about a *co-evolution* of dogs and humans, because our ability to communicate with dogs is as inborn as the ability to communicate with each other. To communicate, we need some tools, and one important tool is our facial muscles. Dogs also developed muscles that raise their inner eyebrow. Some dogs have more developed muscles that others, and recent studies demonstrated that the dogs who has well developed muscles raisins an inner eyebrow are more likely to be adopted by humans, which demonstrates that a natural selection is still working.
For more details, see this article
https://www.newscientist.com/article/2206696-dogs-evolved-a-special-muscle-that-lets-them-make-puppy-dog-eyes/ It describes the results of a recent study published in a serious scientific journal, Proceedings of the National Academy of Sciences of the USA.

10 points:

Bacteria demonstrate extreme genetic plasticity. One of the important mechanisms responsible for such plasticity is horizontal gene transfer (HGT): a bacterium can transfer a piece of its genome, in the form of a relatively short DNA, to another bacterium (-a). For example, a typical *E. coli* genome contains approximately 5,000 protein-coding genes, whereas the *E. coli* pan-genome (all the genes in all the strains) is estimated to contain about 16,000 genes. In other words, a bacterial strain that has no gene, which might be needed in a specific situation, can borrow such a gene from another strain (and lose it later, when not needed). This mechanism has numerous evolutionary advantages, allowing, for example, a quick acquisition of antibiotic resistance. Nevertheless, it is virtually absent (or very rare) in eukaryotes. Please explain why eukaryotes do not use HGT for survival as extensively as bacteria do?

Answer:

Hint:

In contrast to us, bacteria are single-cellular organisms; in addition, their genome has a single set of genes, which, as a rule are combined in a single chromosome.

Solution:

Bacteria have much more simple genetic apparatus than eucaryotae. They have one large circular chromosome, and no nucleus. In addition, they grow very rapidly, and each single bacterial cell is absolutely unimportant for the whole colony. Gene transfer in bacteria occure relatively easily, but its probability is still relatively low, so it affects just few bacterial cells in a colony . However, if that gene increases the chances for survival, the cell with a new gene produces more offsprings, and after several generations they become predominant in the colony, so the majority of cells in the colony acquire a new gene. In contrast, eukaryotic cells have a double set of chromosomes, they have a nucleus, and penetration of a foreign DNA into the nucleus is much more difficult process. In addition, for incorporation of a new gene into the eukaryotic genome, it must incorporate into both chromosomes. That makes horizontal gene transfer more difficult, however even if successful incorporation of a new gene into a single (or few) somatic cell does occur, it hardly has any effect on the whole organism, because the rest of the organism is still having an old, non-modified, genome, and it is hard to imagine a mechanism that can replace old cells with new ones. The only scenario when a horizontal transfer may be successful is an incorporation of a new gene into a genome of gametae (either an egg or a spermatozoid), however, that is an extremely rare event.
Instead of horizontal gene transfer, we, eykatyotae, use sexual reproduction, which allows us to maintain genetic diversity using quite different tools.

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:

Your program will determine whether a set of given positive integers are consecutive elements of the Fibonacci sequence (see https://en.wikipedia.org/wiki/Fibonacci_number).
The program should receive a list of positive integers on input. The numbers will not necessarily be presented in a sorted order. The program should print whether they are consecutive Fibonacci numbers or not, and, if yes, print them in sequential order.

Example 1:

```
Input:  5 13 8 21
Output: YES 5 8 13 21
```

Example 2:

```
Input:  5 13 8 22
Output: NO
```

Hint:

Don't forget that Fibonacci sequence starts with 0, 1, 1…

Solution:

Python:

```
import math

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

```
# also verify that they all > 0
assert(all(x > 0 for x in numbers))
assert(len(numbers) > 0)

numbers.sort()

phi = (1 + math.sqrt(5.0)) / 2.0
psi = (1 - math.sqrt(5.0)) / 2.0
def Binet(n):
  fib = int((math.pow(phi, n) - math.pow(psi, n)) / math.sqrt(5.0))
  print("fib(%d) = %d" % (n, fib))
  return fib

def iBinet(fib):
  n = math.floor(math.log(fib*math.sqrt(5.0) + 0.5, phi))
  print("n(%d) = %d" % (fib, n))
  return n

# we'll follow the wikipedia and start from 0
# or use Binet's formula just for the demo
# it's useful only for the very large starting number
n = iBinet(numbers[0]) # just an example of
fib = Binet(n-2)       # how you would use it

prev_prev = 0
prev = 1
fib = prev_prev + prev
print(fib)
while fib < numbers[0]:
  prev_prev = prev
  prev = fib
  fib = prev_prev + prev
  print(fib)

if fib != numbers[0]: # the very first number is not Fibonacci
  print("NO")
  exit(1)

# check the rest numbers
start = 1
if len(numbers)>=2 and numbers[0]==1 and numbers[1]==1:
  start = 2
for i in range(start, len(numbers)):
  x = numbers[i]

  prev_prev = prev
  prev = fib
  fib = prev_prev + prev
  print(fib)

  if fib != x:
    print("NO")
    exit(1)

print("YES ", numbers)
exit(0)
```

Java:
```
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.util.Arrays;
import static java.lang.Math.pow;
import static java.lang.Math.sqrt;

public class Fib5 {
  int[] numbers; // = {1, 2};
  double phi = (1 + sqrt(5.0)) / 2.0;
  double psi = (1 - sqrt(5.0)) / 2.0;
  int prev_prev, prev, fib;

  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+");
    // the following will croak if the elements are not all integers
    numbers = Arrays.stream(nums).map(s ->
Integer.valueOf(s)).mapToInt(Integer::intValue).toArray();
    // also verify that they all > O
    if(!Arrays.stream(numbers).allMatch(i -> i > 0))
      throw new AssertionError("all numbers must be positive");
    if(numbers.length == 0)
      throw new AssertionError("must provide some numbers");
  }

  void sort() {
    Arrays.sort(numbers);
  }

  long Binet(int n) {
    long fib = (long) ((pow(phi, n) - pow(psi, n)) / sqrt(5.0));
    System.out.printf("fib(%d) = %d\n", n, fib);
    return fib;
  }

  int iBinet(long fib) {
    int n = (int) Math.floor(Math.log(fib * sqrt(5.0) + 0.5) / Math.log(phi));
    System.out.printf("n(%d) = %d\n", fib, n);
    return n;
  }

  void interesting() {
    int n = iBinet(numbers[0]);
    long fib = Binet(n - 2);
  }

  boolean findFirst() {
    prev_prev = 0;
    prev = 1;
    fib = prev_prev + prev;
    System.out.println(fib);
    while(fib < numbers[0]) {
      prev_prev = prev;
      prev = fib;
      fib = prev_prev + prev;
      System.out.println(fib);
    }
    return (fib == numbers[0]); // whether the very first number is Fibonacci or not
  }

  boolean checkRest() {
    int start = 1;
    if(numbers.length >= 2 && numbers[0] == 1 && numbers[1] == 1)
      start = 2;
    for(int i = start; i < numbers.length; i++) {
      int x = numbers[i];

      prev_prev = prev;
      prev = fib;
      fib = prev_prev + prev;
      System.out.println(fib);

      if(fib != x)
        return (false);
    }
    return (true);
  }

  public static void main(String[] args) throws IOException {
    Fib5 fib = new Fib5();
    fib.input();
    fib.sort();
    // we'll follow the wikipedia and start from O
    // or use Binet's formula just for the demo
    // it‘s useful only for the very large starting number
    fib.interesting();
    if(!fib.findFirst()) {
      System.out.println("NO");
      System.exit(1);
    }
    if(!fib.checkRest()) {
      System.out.println("NO");
      System.exit(1);
    }
    else {
      System.out.print("YES ");
      System.out.println(Arrays.toString(fib.numbers));
    }
  }
}
```

10 points:

Your program will determine whether a given positive integer can be written as a sum of consecutive elements of the Fibonacci sequence (see https://en.wikipedia.org/wiki/Fibonacci_number).
The program should receive a single positive integer on input. It should print whether the number can be written as a sum of consecutive Fibonacci numbers, and, if yes, print the sequence.

Example 1:

```
Input:  47
Output: YES 5 8 13 21
```

Example 2:

```
Input:  48
Output: NO
```

Hint:

If a positive integer is a sum of some number of sequential elements of the Fibonacci sequence, the largest number in that sequence can not be larger than the positive integer itself (and can only be equal in one trivial case of 0+1=1).

Solution:

Python:

```
"""
i    -> 0  1  2  3  4  5  6   7   8   9  10  11   12   13   14   15   16    17    18    19
20
f(i) -> 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597, 2584, 4181,
6765

  1  2    3      4        5          6             7
1 1 1+1 1+1+2 1+1+2+3 1+1+2+3+5 1+1+2+3+5+8 1+1+2+3+5+8+13
2 -   1   1+2   1+2+3   1+2+3+5   1+2+3+5+8   1+2+3+5+8+13
3 -   -     2     2+3     2+3+5     2+3+5+8     2+3+5+8+13
4 -   -     -       3       3+5       3+5+8       3+5+8+13
5 -   -     -       -         5         5+8         5+8+13
6 -   -     -       -         -           8           8+13
7 -   -     -        -        -           -             13

Note: we will not keep the entire table in memory
      and we will not recalculate the same thing twice (albeit it is a trivial summation but
still).
"""

f = [0, 1] # array of Fibonacci numbers
s = [1] # array of sums of Fibonacci numbers
n = int(input("enter number: ").strip())
if n == 1:
  print(f"YES {f[0:]}")
else:
  found = False
  while f[-1] < n:
    f_next = f[-1] + f[-2]
    f.append(f_next)
    s = [x + f_next for x in s]
    s.append(f[-2]+f_next)
    for i, x in reversed(list(enumerate(s))):
      if x == n:
        print(f"YES {f[i:]}")
        found = True
        break
      elif x > n:
        break
    if found:
      break
  if not found:
    print("NO")
exit(0)
```

Java:

```
/*
i    -> O  1  2  3  4  5  6   7   8   9  10  11   12   13   14   15   16    17    18    19
20
f(i) -> O, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597, 2584, 4181,
6765

  1  2    3      4        5          6             7
1 1 1+1 1+1+2 1+1+2+3 1+1+2+3+5 1+1+2+3+S+8 1+1+2+3+5+8+13
2 -   1   1+2   1+2+3   1+2+3+5   1+2+3+5+8   1+2+3+5+8+13
3 -   -     2     2+3     2+3+5     2+3+5+8     2+3+5+8+13
4 -   -     -       3       3+5       3+5+8       3+5+8+13
5 -   -     -       -         5         5+8         5+8+13
6 -   -     -       -         -           8           8+13
7 -   -     -       -         -           -             13

Note: we will not keep the entire table in memory
      and we will not recalculate the same thing twice (albeit it is a trivial summation but
still).
*/

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.util.ArrayList;
import java.util.stream.Collectors;
import java.util.stream.IntStream;

public class Fib10 {
  ArrayList<Integer> f = new ArrayList<>(); // array of Fibonacci numbers
  ArrayList<Long> s = new ArrayList<>(); // array of sums of Fibonacci numbers
  int n;

  void input() throws IOException {
    f.add(0);
    f.add(1);
    s.add(1L);

    System.out.print("enter number: ");
    BufferedReader reader = new BufferedReader(new InputStreamReader(System.in));
    String line = reader.readLine().trim();
    n = Integer.parseInt(line);
    if(n <= 0)
      throw new AssertionError("the number must be positive");
  }

  int find() {
    if(n == 1)
      return 0;
    while(f.get(f.size() - 1) < n) {
      int f_next = f.get(f.size() - 1) + f.get(f.size() - 2);
      f.add(f_next);
      s = s.stream().map(x -> x + f_next).collect(Collectors.toCollection(ArrayList::new));
      s.add(f.get(f.size() - 2) + (long) f_next);
      for(int i = s.size() - 1; i >= 0; i--) {
        long x = s.get(i);
        if(x == n)
          return i;
        else if(x > n)
          break;
      }
    }
    return -1;
  }

  public static void main(String[] args) throws IOException {
    Fib10 fib = new Fib10();
    fib.input();
    int i = fib.find();
    if(i >= 0) {
      System.out.print("YES ");
      IntStream.range(i, fib.f.size()).forEach(j -> System.out.printf("%d, ", fib.f.get(j)));
      System.out.println();
    }
    else
      System.out.println("NO");
  }
}
```

LINGUISTICS

5 points:

Consider the following singular and plural forms from an Iranian language:

singular	**translation**	**plural**
tapus	'hawk'	tapusān
kres	'crack'	kresehār
pastuwer	'rain'	pastuweruna
tabak	'dish'	tabakuna
tingun	'tenacity'	tingununa
kašak	'guard'	kašakān
xer	'snore'	xerehār
lāmbuzen	'swimmer'	lāmbuzenān
uš	'camel'	ušān
tox	'cough'	toxehār
zum	'son-in-law'	zumān
čāl	'swing'	čāluna
šrak	'whistle'	šrakehār
rap	'second'	rapuna
grab	'clomping'	grabehār
meč	'fly'	mečān

Explain how plural is formed and form the plurals of the following nouns:

dār	'gallows'
dzaz	'hissing'
až	'bear'
māmuriat	'trip'
totkamār	'wizard'

Hint:

Answer:

dār	'gallows'	dāruna
dzaz	'hissing'	dzazehār
až	'bear'	ažān
māmuriat	'trip'	māmuriatuna
totkamār	'wizard'	totkamārān

Solution:

The choice of suffix depends on the meaning of the noun.

- Animate nouns use suffix -ān
- Nouns denoting sounds use suffix -ehār
- Other nouns use suffix -una

10 points:

Consider the following numbers in one of the Uralic languages (spoken in Siberia). Note that the order of translations does not match the order of original phrases.

sompylasar ɛj šitty, muktyssar ɛj ukkyr, sompylasar ɛj sompyla, šittysar, ukkyr ca muktyssar, šitty ca tɛ̄sar, sompylasar ɛj sel'cy, ukkyr ca tōn

20, 38, 52, 55, 57, 59, 61, 99

Translate into this language the following numbers:
41, 48, 77, 98

Hint:

Answer:

41 = tɛ̄sar ɛj ukkyr
48 = šitty ca sompylasar
77 = sel'cysar ɛj sel'cy
98 = šitty ca tōn

Solution:

It is probably the easiest to start with the simplest one-word number, *šittysar*, which we can assume stands for the round number 20. From here, we can assume that *šitty* is 2, and the suffix *-sar* stands for tens. From here, we can get that *sompylasar ɛj sompyla* is 55, *sompylasar ɛj šitty* is 52, where *ɛj* is addition.

Now we can look at *ukkyr ca muktyssar* and *muktyssar ɛj ukkyr*, which are similar - they contain same number of 10s and 1s. There is also a difference in connector: *ca* and *ɛj*. Two numbers can be obtained from same number of 10s and 1s: 59=60-1 and 61=60+1. Hence, we assume that *ca* is subtraction, *muktyssar* = 60, *mukty* = 6, *ukkyr* = 1.

From here we can get the rest of words: *šitty ca tɛ̄sar* = 38, *tɛ̄* = 4, *ukkyr ca tōn* = 99, *tōn* = 100, *sompylasar ɛj sel'cy* = 57. If the number ends in 8 or 9, it is obtained through subtraction from the next round number.