FPC 2016 problem presentation; spoiler alert!

A - A Match of Table Tennis

B - Breaking the Cipher

C - Crawling

D - Debug

E - Expensive Floor

F - Fences

G - Guessing Game

H - Helping Out

FPC
2016

A - Sample (1/2)

A - A Match of Table Tennis

B - Breaking the Cipher

C - Crawling

D - Debug

E - Expensive Floor

F - Fences

G - Guessing Game

H - Helping Out

FPC 2016

Problem description

Lorum ipsum dolor amet.

Solution - Variables:

- Lorum
- Ipsum

A - Sample (2/2)

Solution - Processing

- Lorum
- Ipsum

B - Breaking the Cipher (1/2)

Problem description

Given an encrypted integer $C \equiv M^e \pmod{n}$, the primes p and q and the integer e, return the decrypted integer $M \equiv C^d \pmod{n}$.

Modular Arithmetic

- $x = k \cdot n + b \quad \Rightarrow \quad b \equiv x \pmod{n}$
- $(a \cdot b) \mod n \equiv ((a \mod n) \cdot (b \mod n)) \mod n$

Solution (1/2)

- Compute $\phi(n) = (p-1) \cdot (q-1)$

B - Breaking the Cipher (2/2)

Solution (2/2)

- Find d by trying every $d \in [1, \phi(n)]$ and checking if $(d \cdot e)\%\phi(n) = 1$.
- Decrypt by computing C^d, but apply $(\mod n)$ after each multiplication:
- $M = 1$; for$(i = 1..d)$ $M = (M * d)\%n$

C - Crawling (1/2)

A - A Match of Table Tennis
B - Breaking the Cipher
C - Crawling
D - Debug
E - Expensive Floor
F - Fences
G - Guessing Game
H - Helping Out

FPC
2016

Problem description

Given the time of the fastest student measured, does Saint Alex have a student that can beat this time?

Solution - Variables:

- t - The time of the fastest student measured.
- l - The length of the pool.
- n - Number of students. For every student: f, b - The speed of the front resp. back crawl of the student.

For every student, calculate $c = \frac{l}{f} + \frac{l}{b}$. If you find a student whose $c < t$, print "HOPE" and return. If all students are checked, print "DOOMED".

C - Crawling (2/2)

Pitfalls

One of the more easy problems in the set, some points:

- Use floats (or doubles) and not integers.
- Do NOT calculate the average speed i.e. $\frac{f+b}{2}$ and then divide $2l$ by this float. This is not correct.

D - Debug (1/2)

Problem description

Find out what the given code does and make it faster.

Solution:

- The code is checking whether a given number is a prime number. If so, it outputs *yes*, else, it outputs *no*.
- Note that 1 is *not* a prime number.
- How can we make it faster?

D - Debug (2/2)

Making the code faster

- Various optimizations possible:
- Start the loop from 3 and create if-statements for the cases $n = 1$ (no prime) and $n = 2$ (prime).
- Let the for-loop skip even numbers and check whether the number is even before entering the for-loop.
- Return immediately from the method if a divisor of n is found in the for-loop.
- Loop to (including) $\sqrt{n}$ at most since the minimum of a and b where $ab = n$ is at most $\sqrt{n}$.

A - A Match of Table Tennis

B - Breaking the Cipher

C - Crawling

D - Debug

E - Expensive Floor

F - Fences

G - Guessing Game

H - Helping Out

FPC 2016

E - Expensive Floor

Problem description

Determine the total number of squares in a n x n floor.

Solution:

- The total number of squares $f(n)$ in a n x n floor is given by the following summation:
- $f(n) = \sum_{i=1}^{n} i^2$
- Can be implemented by using one for-loop. Create a variable to keep track of the current answer.
- You should be using the *long* data type in Java in order to avoid overflow errors for a big value of n.

A - A Match of Table Tennis

B - Breaking the Cipher

C - Crawling

D - Debug

E - Expensive Floor

F - Fences

G - Guessing Game

H - Helping Out

FPC 2016

F - Fences

Problem description

- Given the area A
- Compute the radius $r = \sqrt{\frac{A}{\pi}}$
- Compute the perimeter $P = 2\pi r$
- Round up the solution $\frac{\lceil P \cdot 10 \rceil}{10}$

G - Guessing Game

A - A Match of Table Tennis

B - Breaking the Cipher

C - Crawling

D - Debug

E - Expensive Floor

F - Fences

G - Guessing Game

H - Helping Out

FPC 2016

Problem

Guess the correct number.

- You can't guess every number, that would be too slow.
- Use binary search!

Solution

- Keep track of a lower l and upper u limit and repeat:
  - guess $x = (l + u)/2$
  - If x is too low, set $l = x + 1$
  - If x is too high, set $u = x - 1$

H - Helping Out

Problem description

Given a list of participants and their scores, give the total score of each participant, list the total score of each participant in alphabetic order.

Solution

- Use a Map<String,Integer> to store the score of each participant.
- If the map already contains the name, add the new score to the current score to the map.
- Otherwise add a new entry to the map.
- Use a TreeMap to automatically print in alphabetic order.