MIDDLE EAST TECHNICAL UNIVERSITY
DEPARTMENT OF COMPUTER ENGINEERING

CENG 111

Fall 2021

Take-Home Exam 2

REGULATIONS

Due date: 23:59, 26 December 2021, Sunday *(Not subject to postpone)*

Submission: Electronically. You should save your program source code as a text file named `the2.py`. Check announcement on ODTUCLASS course page for the submission procedure.

Team: There is **no** teaming up. This is an EXAM.

Cheating: Source(s) and Receiver(s) will receive zero and be subject to disciplinary action.

INTRODUCTION

A *check digit* for a sequence of digits or letters is "a form of redundancy check used for error detection on identification numbers, such as bank account numbers, which are used in an application where they will at least sometimes be input manually"[1]. It consists of one or more digits computed by an algorithm from the other digits (or letters) in the sequence. With a check digit, one can detect simple errors in a sequence of characters (usually digits) such as a single mistyped digit or some permutations of two successive digits.

The NorthStar College implements also a check digit algorithm for student numbers. A student number has exactly four digits. The smallest student number is `"0001"` . The zeros are part of the number and are not omitted. Given such four digits as a student number, a check digit is calculated, and appended with a dash sign to the four digits to form the *student ID*.

Let us denote a student number as:

$$\mathbf{d}_1\mathbf{d}_2\mathbf{d}_3\mathbf{d}_4$$

Then, the check digit for the NorthStar College is computed as:

$$CheckDigit = (2 \times \mathbf{d}_1 + 3 \times \mathbf{d}_2 + 5 \times \mathbf{d}_3 + 7 \times \mathbf{d}_4) \mod 11$$

An uppercase '`X`' is used for a value of 10 of the $CheckDigit$.

Take as example the student number **6792**:

$$\begin{aligned} CheckDigit &= (2 \times \mathbf{6} + 3 \times \mathbf{7} + 5 \times \mathbf{9} + 7 \times \mathbf{2}) \mod 11 \\ &= 92 \mod 11 \\ &= 4 \end{aligned}$$

The check digit of the student number **6792** is found to be **4**. Therefore, similar to your student IDs, this is written as **6792-4**.

[1] Definition taken from Wikipedia.

PROBLEM & SPECIFICATIONS

Your mission, should you choose to accept it, is to determine the digit that is missing from a student-id or, if no digit is missing, to determine whether it is valid or not. A student ID will be read from the standard input as:

####-#

where each # is a decimal digit or a '?' (question mark). There will be at most one question mark. It is also possible that no question mark appears in the input. You will be printing a single line of output which strictly follows the following specifications:

- **There is no question mark:** Print `VALID` if the check digit is correct. Otherwise, print `INVALID`.
- **CheckDigit position has question mark:** Compute the check digit and print the whole student ID (with the correct check digit present).
- **One of the first six digit positions has question mark:** Compute the digit position that has question mark and print the whole student ID (with the correct digit present).

You are not allowed to:

- Define functions.
- Use any kind of repetitive constructs.
- Insert-digit-and-test until you find the correct digit. **This is not allowed.** We will look into your code! You have to find one (or a group of) formula(s) for determining what the '?' mark stands for.

FIVE EXAMPLE RUNS

```
>>> 6792-4
VALID

>>> 6792-5
INVALID

>>> 6792-?
6792-4

>>> 679?-4
6792-4

>>> ?792-4
6792-4
```

HINTS

- `"garabettin".find("b")` will return `4`. Certainly it will work with string holding variables as well.

GRADING

- Comply with the specifications. Do not try to beautify neither the input nor the output. Your program will be graded through an automated process.
- Your program will be tested with multiple data (a distinct run for each data). Any program that performs only 30% and below will enter a glass-box test (eye inspection by the grader TA). The TA will judge an overall THE2 grade in the range of [0,30].
- Half of the test set will be for checking the validity of IDs.
- A program based on randomness will be graded zero.
- The glass-box test grade is not open to discussion nor explanation.
- Your code will be evaluated using Python 3.8.10 in the Linux environment on inek machines. Make sure that you test your code on inek machines before submission.