Elementary Programming Examination

※**Open book**, you're welcome to browse any resources.

※You're welcome to use any programming language.

※We have 6 questions, each **10** points, total **60** points. Just do your best.

※After completion of each question, save it as a .py file, and name it as [name]_test[question number].py

No. 1 Narcissistic number | ★☆☆☆☆ (Algorithm)

Given the definition of a Narcissistic number (k is the number of digits of n):

$$n = d_k^{\ k} + d_{k-1}^{\ \ k} + \dots + d_2^{\ k} + d_1^{\ k}.$$

For instance, $153 = 1^3 + 5^3 + 3^3$. Therefore, 153 is a Narcissistic number. Please write a program that can automatically find out Narcissistic numbers between the interval [100 ~ 999]

No. 2 A Simple Differentiation rule | ★★☆☆☆ (Algorithm)

Assumed that you guys haven't dived into the world of Calculus(微積分). Therefore, given the definition of a simple differentiate rule(微分法則):

$$y = f(x) = x^n \quad \text{differentiate} \rightarrow \quad \frac{dy}{dx} = f'(x) = nx^{n-1}$$

$$\text{Example:} \ y = f(x) = x^2 \quad \text{differentiate} \rightarrow \quad \frac{dy}{dx} = f'(x) = 2x$$

We have two program inputs: $X \in Z$, $P \in Z$, which represent X^P ,write a function diff(X, P) that output the result of differentiation.

No. 3 A Bounding Box | ★★☆☆☆ (Algorithm)

Given a bounding area consist of 4 points, write a program that determine wether the (0, 0) is in this bounding area. The order will be clockwise, and the input will be :

```
1 4   #(X1, Y1)
2 -1
-4 -4
-3 4
```
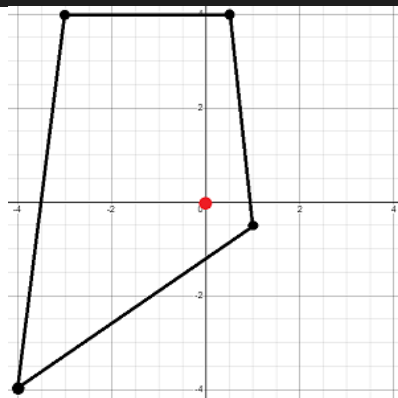

If the point is in the area, output 1, else 0.

(hint: use the area of triangle)

(another hint: topology is a good stuff, odd is inside, even is outside)

No. 4 Gravity on | ★★☆☆☆ (Algorithm & Data Structure)

Given a matrix containing several floating blocks, please design a function that will switch on the gravity of that matrix.

For instance:

```
# Gravity off
[['■','■','■','□','■','■'],
 ['□','□','■','□','■','□'],
 ['□','■','□','■','□','□'],
 ['■','□','□','■','■','■']]

# Gravity on
[['□','□','□','□','□','□'],
 ['□','□','□','□','■','□'],
 ['■','■','■','■','■','■'],
 ['■','■','■','■','■','■']]
```

As you can see, after the gravity switch on, all the solid floating blocks fall, just like the sand blocks in Minecraft!

No. 5 Vertical Text | ★★☆☆☆ (Algorithm & Data Structure)

Create a function that converts a string into a matrix of characters that can be read vertically. Add spaces when characters are missing.

For instance:

```
"Hloy bananas"

[["H", "b"],
 ["o", "a"],
 ["l", "n"],
 ["y", "a"],
 [" ", "n"],
 [" ", "a"],
 [" ", "s"]]
```

No. 6 Orthogonal set | ★☆☆☆☆ (Algorithm & Data Structure)

An orthogonal matrix(正射矩陣) is a set that consist of several vectors, and every vector is orthogonal to each other. We say that two vectors are orthogonal to each other iff the inner product is 0.

And the definition of an orthogonal set is defined as:

$$S = [V_1 \ V_2 \ V_3 \ ... \ V_n] \ , \sum_{i=1}^{n} \sum_{j!=i}^{n} V_i \cdot V_j = 0$$

Input example:

```
[[1,0,0,0],
 [0,1,0,0],
 [0,0,1,0],
 [0,0,0,1]]
```

Write a program that determine whether an input matrix is an orthogonal set.