SCHOOLOGY PROGRAMMING CHALLENGES

Below are six programming puzzles. Pick and solve two of them. All solutions must be written in PHP, Java, Ruby, Python, or any other popular server side programming language. Looking online for libraries or code snippets is encouraged; however, do not search for exact solutions to the problem and paste them as your own solution.

The solutions must be completed as a web form, using textfields and textareas as inputs for the challenges. If there is a “sample input” included in the challenge, use those as default values for the inputs that can be changed by the user.

ENSLIGH-NUMBER TRANSLATOR

In this problem, you will be given one or more integers in English. Your task is to translate these numbers into their integer representation. The numbers can range from negative 999,999,999 to positive 999,999,999. The following is an exhaustive list of English words that your program must account for:

negative, zero, one, two, three, four, five, six, seven, eight, nine, ten, eleven, twelve, thirteen, fourteen, fifteen, sixteen, seventeen, eighteen, nineteen, twenty, thirty, forty, fifty, sixty, seventy, eighty, ninety, hundred, thousand, million

INPUT AND OUTPUT

Notes on input:

- Negative numbers will be preceded by the word negative.
- The word ``hundred'' is not used when ``thousand'' could be. For example, 1500 is written ``one thousand five hundred'', **not** ``fifteen hundred''.

SAMPLE INPUT

```
six, negative seven hundred twenty nine, one million one hundred one
```

SAMPLE OUTPUT

```
6, -729, 1000101
```

FIBONACCI FREEZE

The Fibonacci numbers (0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, ...) are defined by the recurrence:

$$F_0 = 0$$
$$F_1 = 1$$
$$F_i = F_{i-1} + F_{i-2} \; for \; all \; i \geq 2$$

Write a program to calculate the Fibonacci Numbers.

INPUT AND OUTPUT

The input to your program would be a sequence of numbers smaller or equal than 5000, each on a separate line, specifying which Fibonacci number to calculate.

Your program should output the Fibonacci number for each input value, one per line.

SAMPLE INPUT

```
5, 7, 11
```

SAMPLE OUTPUT

```
5, 13, 89
```

KAPREKAR NUMBERS

Wolfram's MathWorld describes Kaprekar numbers like this:

> *Consider an n-digit number k. Square it and add the right n digits to the left n or n-1 digits. If the resultant sum is k, then k is called a Kaprekar number. For example, 9 is a Kaprekar number since $9^2 = 81$ and 8 + 1 = 9 and 297 is a Kaprekar number since $297^2 = 88209$ and 88 + 209 = 297.*

Your task is to write a function that identifies Kaprekar numbers and to determine the Kaprekar numbers less than a thousand.

ALIEN NUMBERS

The decimal numeral system is composed of ten digits, which we represent as “0123456789″ (the digits in a system are written from lowest to highest). Imagine you have discovered an alien numeral system composed of some number of digits, which may or may not be the same as those used in decimal. For example, if the alien numeral system were represented as “oF8″, then the numbers one through ten would be:

```
F, 8, Fo, FF, F8, 8o, 8F, 88, Foo, FoF
```

We would like to be able to work with numbers in arbitrary alien systems. More generally, we want to be able to convert an arbitrary number that’s written in one alien system into a second alien system.

INPUT

```
alien_number source_language target_language
```

Each language will be represented by a list of its digits, ordered from lowest to highest value. No digit will be repeated in any representation, all digits in the alien number will be present in the source language, and the first digit of the alien number will not be the lowest valued digit of the source language (in other words, the alien numbers have no leading zeroes). Each digit will either be a number 0-9, an uppercase or lowercase letter, or one of the following symbols !”#$%&’()*+,-./:;?@[\]^_`{|}~

OUTPUT

Your function should return the translated number.

HAPPY NUMBERS

A happy number is defined by the following process. Starting with any positive integer, replace the number by the sum of the squares of its digits, and repeat the process until the number equals 1 (where it will stay), or it loops endlessly in a cycle which does not include 1. Those numbers for which this process ends in 1 are happy numbers, while those that do not end in 1 are unhappy numbers (or sad numbers).

For example, 7 is a happy number, as $7^2=49$, $4^2+9^2=16+81=97$, $9^2+7^2=81+49=130$, $1^2+3^2+0^2=1+9+0=10$, and $1^2+0^2=1+0=1$. But 17 is not a happy number, as $1^2+7^2=1+49=50$, $5^2+0^2=25+0=25$, $2^2+5^2=4+25=29$, $2^2+9^2=4+81=85$, $8^2+5^2=64+25=89$, $8^2+9^2=64+81=145$, $1^2+4^2+5^2=1+16+25=42$, $4^2+2^2=16+4=20$, $2^2+0^2=4+0=4$, $4^2=16$, $1^2+6^2=1+36=37$, $3^2+7^2=9+49=58$, and $5^2+8^2=25+64=89$, which forms a loop.

Write a function that takes an array of numbers and filters out non-happy numbers (i.e. it returns an array of happy numbers).

GREPLIN CHALLENGE 1

In the block of text below, find the longest substring that is the same in reverse (palindrome).

As an example, if the input was "Ilikeracecarsthatgofast" the answer would be "racecar".

```
Fourscoreandsevenyearsagoourfaathersbroughtforthonthiscontainentanewnationconceiv
edinzLibertyanddedicatedtothepropositionthatallmenarecreatedequalNowweareengagedi
nagreahtcivilwartestingwhetherthatnaptionoranynartionsoconceivedandsodedicatedcanlo
ngendureWeareqmetonagreatbattlefiemldoftzhatwarWehavecometodedicpateaportionoft
hatfieldasafinalrestingplaceforthosewhoheregavetheirlivesthatthatnationmightliveItisalto
getherfangandproperthatweshoulddothisButinalargersensewecannotdedicatewecannotco
nsecratewecannothallowthisgroundThebravelmenlivinganddeadwhostruggledherehaveco
nsecrateditfaraboveourpoorponwertoaddordetractTgheworldadswfilllittlenotlenorlongre
memberwhatwesayherebutitcanneverforgetwhattheydidhereItisforusthelivingrathertobed
edicatedheretotheulnfinishedworkwhichtheywhofoughtherehavethusfarsonoblyadvancedI
tisratherforustobeherededicatedtothegreattdafskremainingbeforeusthatfromthesehonore
ddeadwetakeincreaseddevotiontothatcauseforwhichtheygavethelastpfullmeasureofdevoti
onthatweherehighlyresolvethatthesedeadshallnothavediedinvainthatthisnationunsderGod
shallhaveanewbirthoffreedomandthatgovernmentofthepeoplebythepeopleforthepeoplesh
allnotperishfromtheearth
```