**Problem 1:**

**Stepwise word**

**Challenge Description:**

Print the longest word in a stepwise manner.

**Input sample:**

The first argument is a path to a file. Each line contains a test case with a list of words that have different or the same length.

For example:

cat dog hello

stop football play

music is my life

**Output sample:**

Find the longest word in each line and print it in one line in a stepwise manner. Separate each new step with a space. If there are several words of the same length and they are the longest, then print the first word from the list.

h \*e \*\*l \*\*\*l \*\*\*\*o

f \*o \*\*o \*\*\*t \*\*\*\*b \*\*\*\*\*a \*\*\*\*\*\*l \*\*\*\*\*\*\*l

m \*u \*\*s \*\*\*i \*\*\*\*c

**Constraints:**

1. The word length is from 1 to 10 characters.
2. The number of words in a line is from 5 to 15.
3. If there are several words of the same length and they are the longest, then print the first word from the list.
4. The number of test cases is 40.

Problem 2

**Challenge Description:**

You are building a new social platform and want to store user's work experience. You have decided to calculate the total experience of each user in years based on the time periods that they provided. Using this approach, you need to be sure that you are taking into account the overlapping time periods in order to retrieve the actual work experience in years.

For example:

Jan 2010-Dec 2010

Jan 2010-Dec 2010

Two jobs with 12 months of experience each, but actual work experience is 1 year because of the overlapping time periods. The task is to calculate the actual work experience based on the list of time intervals.

**Input sample:**

Your program should accept a path to a filename as its first argument. Each line of the file contains a list of time periods separated by a semicolon and a single space. Each time period is represented as the begin date and the end date. Each date consists of a month as an abbreviated name and a year with century as a decimal number separated by a single space. The begin date and the end date are separated by a hyphen.

For example:

Feb 2004-Dec 2009; Sep 2004-Jul 2008

Aug 2013-Mar 2014; Apr 2013-Aug 2013; Jun 2014-Aug 2015; Apr 2003-Nov 2004; Apr 2014-Jan 2015

Mar 2003-Jul 2003; Nov 2003-Jan 2004; Apr 1999-Nov 1999

Apr 1992-Dec 1993; Feb 1996-Sep 1997; Jan 2002-Jun 2002; Sep 2003-Apr 2004; Feb 2010-Nov 2011

Feb 2004-May 2004; Jun 2004-Jul 2004

**Output sample:**

Print out the actual work experience in years for each test case.

For example:

5

4

1

6

0

**Constraints:**

1. The number of lines in a file is in a range from 20 to 40.
2. The dates are in a range from Jan 1990 to Dec 2020.
3. The end date is greater than the begin date.
4. The begin date is the first day of a given month, and the end date is the last day of a given month.

Problem 3

When John was a little kid he didn't have much to do. There was no internet, no Facebook, and no programs to hack on. So he did the only thing he could... he evaluated the beauty of strings in a quest to discover the most beautiful string in the world.  
  
Given a string s, little Johnny defined the beauty of the string as the sum of the beauty of the letters in it. The beauty of each letter is an integer between 1 and 26, inclusive, and no two letters have the same beauty. Johnny doesn't care about whether letters are uppercase or lowercase, so that doesn't affect the beauty of a letter. (Uppercase 'F' is exactly as beautiful as lowercase 'f', for example.)  
  
You're a student writing a report on the youth of this famous hacker. You found the string that Johnny considered most beautiful. What is the maximum possible beauty of this string?

### Input sample:

Your program should accept as its first argument a path to a filename. Each line in this file has a sentence. E.g.

ABbCcc

Good luck in the Facebook Hacker Cup this year!

Ignore punctuation, please :)

Sometimes test cases are hard to make up.

So I just go consult Professor Dalves

### Output sample:

Print out the maximum beauty for the string. E.g.

152

754

491

729

646

Problem 5

The goal of this challenge is to create a simple calculator.  
The following operations should be supported with their order (operator precedence):

1 **()** Brackets

2 **-** Unary minus

3 **^** Exponent

4 **\***, **/** Multiply, Divide (left-to-right precedence)

5 **+**, **-** Add, Subtract (left-to-right precedence)

### Input sample:

Your program should accept as its first argument a path to a filename. The input file contains several lines. Each line is one test case. Each line contains mathematical expression. E.g.

250\*14.3

3^6 / 117

(2.16 - 48.34)^-1

(59 - 15 + 3\*6)/21

### Output sample:

For each set of input produce a single line of output which is the result of calculation.

3575

6.23077

−0.02165

2.95238

Note: Don't use any kind of **eval** function.

**Constraints:**  
Each number in input expression is greater than -20,000 and less than 20,000.  
Each output number is greater than -20,000 and less than 20,000.  
If output number is a float number it should be rounded to the 5th digit after the dot.  
E.g 14.132646 gets 14.13265, 14.132644 gets 14.13264, 14.132645 gets 14.13265.  
  
If output number has less than 5 digits after the dot you don't need to add zeros.  
E.g. you need to print 16.34 (and not 16.34000) in case the answer is 16.34.  
And you need to print 16 (and not 16.00000) in case the answer is 16.

Problem 6:

**Data Recovery**

**Challenge Description:**

Your friends decided to make fun of you. They have installed a script on your computer which shuffled all words within a text. It is a joke, so they have left hints for each sentence. The hints will allow you to rebuild the data easily, but you need to find out how to use them.

Your task is to write a program which reconstructs each sentence out of a set of words and prints out the original sentences.

**Input sample:**

Your program should accept a path to a filename as its first argument. Each line is a test case which consists of a set of words and a sequence of numbers separated by a semicolon. The words within a set and the numbers within a sequence are separated by a single space.

For example:

2000 and was not However, implemented 1998 it until;9 8 3 4 1 5 7 2

programming first The language;3 2 1

programs Manchester The written ran Mark 1952 1 in Autocode from;6 2 1 7 5 3 11 4 8 9

**Output sample:**

Print out a reconstructed sentence for each test case, one per line.

For example:

However, it was not implemented until 1998 and 2000

The first programming language

The Manchester Mark 1 ran programs written in Autocode from 1952

**Constraints:**

1. The number of test cases is in a range from 20 to 40.
2. The words consist of ASCII uppercase and lowercase letters, digits, and punctuation marks.