S.A. Computer Olympiad Second Round 2005

Standard Bank

Q1. Sum of Squares

Prepared by Shen Tian

Description

The square of a number is the product of the number with itself. For example, the square of 5, written 5^2 is 5 x 5 = 25.

Task

Your task is to find the sum of the squares of all integers (whole numbers) between two given integers, m and n, inclusive of m and n.

Constraints

$0 \leq m < n \leq 45$

Sample run

For example, if m = 1 and n = 3, the sum we are looking for is $1^2 + 2^2 + 3^2 = 1 + 4 + 9 = 14$.

Input

```
Enter m: 1
Enter n: 3
```

Output

```
The sum of squares is: 14
```

Test your program with

a. m = 3, n = 7
b. m = 10, n = 45

Q2. Encryption

Prepared by Shen Tian

Description

Encryption is the process of encoding information, in our case a message, to make it unreadable without special knowledge. This knowledge consists of the method of encryption and something called a key.

We shall use a very simple method of encrypting a message. The message we will encrypt will only contain the 26 capital letters and spaces. There is no punctuation in the messages.

The key is a sequence of distinct letters.

The method of encryption is as follows:

- Look at each letter of the message.
- If the letter is in the key, replace it with the letter following it in the key. If it is the last letter in the key, replace it with the first letter in the key.
- If the letter is not in the key, don't change it.
- Leave spaces unchanged.

Task

Your task will be to encrypt a message. You will be supplied with the unencrypted message and the key. The unencrypted message will contain capital letters and spaces, and will be at most 255 characters long. The key will be at most 13 characters long. Letters will not be repeated in the key.

Sample run

Given the message COMPUTER OLYMPIAD, and the key MOPED, we can do the following:

- C is not in the key. It is left unchanged.
- O is in the key. Replace it with the letter after O, P
- …
- …
- D is in the key. It is the last letter in the key. Replace it with the first letter of the key, M.

The full encrypted message will be:

CPOEUTDR PLYOEIAM

Input

```
Message: COMPUTER OLYMPIAD
Key: MOPED
```

Output

```
CPOEUTDR PLYOEIAM
```

Test your program with

a. Message: MOSTLY HARMLESS
   Key: HARMONY
b. Message: ALL YOUR BASES BELONG TO US
   Key: AWDXBFTYHNKSU

Q3. Digits

Prepared by Harry Wiggins and Bruce Merry

Task

Given two positive integers M and N, find the smallest integer greater than M whose digits add up to N. For example, the smallest integer greater than 100 whose digits add up to 11 is 119.

Sample run

Your program should request the integer M and digit sum N.

Input

```
Enter integer (M): 100
Enter digit sum (N): 11
```

Output

```
Next integer: 119
```

Test your program with

a. M = 56, N = 10
b. M = 999, N = 27

SHUTTLEWORTH FOUNDATION

CSIR OPEN SOURCE CENTRE (OSC)

SACO Second Round 2005

Standard Bank

Q4. Date Calculations

Prepared by Carl Hultquist

Description

Alison wants to work out how old people are, but has the strange habit of wanting to know how many *days* old a person is. Alison's big problem is that she's not good at remembering all the rules about leap years (in a leap year, February will have 29 days). All years that are divisible by 4 are leap years, *except* for years which are divisible by 100. If a year is divisible by 100, then it is only a leap year if it is also divisible by 400; otherwise it is not a leap year.

Task

You must write a program that, given a person's date of birth and a target date, calculates how many days old the person is on that target date. Dates should be entered in the form `dd mm yyyy` (2 digits for the day, 2 digits for the month, and 4 digits for the year, with a single space between the day and month, and between the month and year). If the birth date and target date are the same, then we say that the person is 0 days old. All dates will be A.D. You may not use any date or time related functions available in your programming language. All date calculations must be done by your program using the usual arithmetic features.

Sample run

Input

```
Enter date of birth: 16 09 2003
Enter target date: 04 05 2005
```

Output

```
Difference in days: 596
```

Test your program with

a. Date of birth: 21 02 1780
   Target date: 30 11 1903

b. Date of birth: 29 02 1980
   Target date: 31 12 1999

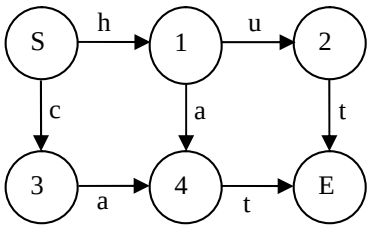
Q5. Finite State Machine

Prepared by Bruce Merry

Description

A finite state machine (FSM) is a conceptual machine used to do string matching. It consists of a number of states (circles in the diagram), linked by arrows that are labelled with letters. From each state, there can be many arrows, but there is at most one outgoing arrow for each letter. A FSM matches a word if (and only if) there is a path from the start state to the end state that spells out the word.

For example, the machine below matches the words 'hat', 'hut', and 'cat' (and no others). S represents the start state and E the end state.



There can be more than one FSM for a given set of strings. We could, for example, have merged states 2 and 4 above, to give a machine with only 5 states.

Task

Write a program that will, given a list of words (lowercase English letters), determine the minimum number of states in any finite state machine that matches those words and no others.

Sample run

Input

```
Enter a list of words, one per line,
with a blank line to end:
hat
hut
cat
```

Output

```
Minimum size of FSM: 5 states
```

Test your program with

a.
```
the
two
thousand
and
five
computer
olympiad
```

b.
```
one
sole
sock
seek
muck
mule
mane
```