The CENTRE for EDUCATION
in MATHEMATICS and COMPUTING

2007 Canadian Computing Competition: Junior Division

Sponsor:

University of
Waterloo

Canadian Computing Competition
Student Instructions for the Junior Problems

1. You may only compete in one competition. If you wish to write the Senior paper, see the other problem set.

2. Be sure to indicate on your **Student Information Form** that you are competing in the **Junior** competition.

3. You have three (3) hours to complete this competition.

4. You should assume that

   - all input is from the keyboard
   - all output is to the screen

   For some problems, you may be asked for prompting: please provide this for the user. If no prompting is required, you do not need to provide any. Be sure your output matches the output in terms of order, spacing, etc. IT SHOULD MATCH EXACTLY!

5. Do your own work. Cheating will be dealt with harshly.

6. Do not use any features that the judge (your teacher) will not be able to use while evaluating your programs.

7. Books and written materials are allowed. Any machine-readable materials (like other programs which you have written) are *not* allowed. However, you are allowed to use "standard" libraries for your programming languages; for example, the STL for C++, java.util.*, java.io.*, etc. for Java, and so on.

8. Applications other than editors, compilers, debuggers or other standard programming tools are **not** allowed. Any use of other applications will lead to disqualification.

9. Please use file names that are unique to each problem: for example, please use `j1.pas` or `j1.c` or `j1.java` (or some other appropriate extension) for Problem J1. This will make the evaluator's task a little easier.

10. Your program will be run against test cases other than the sample ones. Be sure you test your program on other test cases.

11. Note that the top 2 Junior competitors in each region of the country will get a plaque and $100, and the schools of these competitors will also get a plaque. The regions are:

    - West (BC to Manitoba)
    - Ontario North and East
    - Metro

- Ontario Central and West
- Quebec and Atlantic

12. Check the CCC website at the end of March to see how you did on this contest, how the problems were meant to be solved, and to see who the prize winners are. The CCC website is:

`www.cemc.uwaterloo.ca/ccc`

Problem J1: Who is in the middle?

Problem Description

In the story *Goldilocks and the Three Bears*, each bear had a bowl of porridge to eat while sitting at his/her favourite chair. What the story didn't tell us is that Goldilocks moved the bowls around on the table, so the bowls were not at the right seats anymore. The bowls can be sorted by weight with the lightest bowl being the Baby Bear's bowl, the medium bowl being the Mama Bear's bowl and the heaviest bowl being the Papa Bear's bowl. Write a program that reads in three weights and prints out the weight of Mama Bear's bowl. You may assume all weights are positive integers less than 100.

Sample Input

```
10
5
8
```

Output for Sample Input

```
8
```

Problem J2: I Speak TXTMSG

Problem Description

Text messaging using a cell phone is popular among teenagers. The messages can appear peculiar because short forms and symbols are used to abbreviate messages and hence reduce typing.

For example, "LOL" means "laughing out loud" and ":-)" is called an emoticon which looks like a happy face (on its side) and it indicates chuckling. This is all quite a mystery to some adults.

Write a program that will continually input a short form and output the translation for an adult using the following translation table:

Short Form	Translation
CU	see you
:-)	I'm happy
:-(	I'm unhappy
;-)	wink
:-P	stick out my tongue
(~.~)	sleepy
TA	totally awesome
CCC	Canadian Computing Competition
CUZ	because
TY	thank-you
YW	you're welcome
TTYL	talk to you later

Input Specifications

The user will be prompted to enter text to be translated one line at a time. When the short form "TTYL" is entered, the program ends. Users may enter text that is found in the translation table, or they may enter other words. All entered text will be symbols or upper case letters. There will be no spaces and no quotation marks.

Output Specifications

The program will output text immediately after each line of input. If the input is one of the phrases in the translation table, the output will be the translation; if the input does not appear in the table, the output will be the original word. The translation of the last short form entered "TTYL" should be output.

Sample Session (user input is in *italics*)

```
Enter phrase> CCC
Canadian Computing Competition
```

```
Enter phrase> :-)
I'm happy
Enter phrase> SQL
SQL
Enter phrase> TTYL
talk to you later
```

Problem J3: Deal or No Deal Calculator

Problem Description
"Deal or No Deal" (TM) is a game show on NBC. (You can play it at
`http://www.nbc.com/Deal_or_No_Deal/game/flash.shtml`)

In the CCC version of the game, there are 10 possible dollar amounts: \$100, \$500, \$1 000, \$5 000, \$10 000, \$25 000, \$50 000, \$100 000, \$500 000, \$1 000 000 sealed in imaginary briefcases. These dollar amounts are numbered 1 – 10 (i.e. 1 $\rightarrow$ \$100, 2 $\rightarrow$ \$500, 3 $\rightarrow$ \$1 000, ..., 10 $\rightarrow$ \$1 000 000). Before the game starts the contestant will have chosen one of the briefcases as his/hers to possibly keep. During the game, some of the ten possible dollar amounts have been eliminated from the game because the contestant has selected some of the other briefcases and revealed the amounts inside.

At some point, the contestant will stop opening briefcases, and a "Banker" will offer the contestant cash in exchange for what might be contained in his/her chosen briefcase. Then the contestant is asked: "Deal or No Deal?".

Write a program that helps a player decide if he/she should choose "deal" or "no deal", by calculating the average of the remaining amounts (i.e., all unopened briefcases, including his/her "own" briefcase), and comparing that value to the "Banker's" offer. If the offer is higher than the average, then the player should "deal" otherwise, the player should say "no deal".

Input Specifications
The user must input a number n $(1 \leq n < 10)$ which indicates how many cases have been opened so far, followed by a list of n integers between 1 and 10 representing the values in the game that have been eliminated, followed by the "Banker's" offer. For example: `3 2 5 10 300` indicates that briefcases containing \$500, \$10 000, and \$1 000 000 have been eliminated and the Banker's offer is \$300. You may assume that no duplicate case numbers are entered for the eliminated values, and you may assume that the "Banker's" offer is an integer greater than 10.

Output Specifications
The program will print out one of two statements: "deal" or "no deal".

Sample Input 1
```
2
3
8
198000
```

Output for Sample Input 1
```
no deal
```

Sample Input 2

```
8
10
9
8
7
6
5
4
3
400
```

Output for Sample Input 2

```
deal
```

Problem J4: Anagram Checker

Problem Description
An anagram is a word or a phrase formed by rearranging the letters of another phrase such as "ITEM" and "TIME". Anagrams may be several words long such as "CS AT WATERLOO" and "COOL AS WET ART". Note that two phrases may be anagrams of each other even if each phrase has a different number of words (as in the previous example). Write a program to determine if two phrases are anagrams of each other.

Input Specifications
The program should prompt the user for two phrases, each entered on a separate line. You may assume that the input only contains upper case letters and spaces.

Output Specifications
The program will print out one of two statements: "Is an anagram." or "Is not an anagram."

Sample Prompting and User Input (user input in *italics*)
`Enter first phrase>` *CS AT WATERLOO*
`Enter second phrase>` *COOL AS WET ART*

Output for Sample Input
```
Is an anagram.
```

Problem J5: Keep on Truckin'

Problem Description

A truck driver is planning to drive along the Trans-Canada highway from Vancouver to St. John's, a distance of 7000 km, stopping each night at a motel. The driver has been provided with a list of locations of eligible motels, with the respective distance of each motel, measured in km, from the starting point in Vancouver. Some of the motel locations are:

0, 990, 1010, 1970, 2030, 2940, 3060, 3930, 4060, 4970, 5030, 5990, 6010, 7000

but more motel locations may be added just before the trip begins.

Determine if it is possible to complete the journey if:

1. the trucking company insists that the driver travels a minimum distance of A km per day,
2. the law sets a maximum distance of B km per day, and
3. each night, the driver must stay at an eligible motel (from the above list or the additional locations described below).

The driver is interested in different options when making the trip, and you are to write the program to calculate how many different options there are.

For example, if no new motel locations are added, $A = 1$ and $B = 500$, then it is impossible to make the trip, i.e., the number of options is 0. If $A = 970$ and $B = 1030$ then there is one way to make the trip, but if $A = 970$ and $B = 1040$ then there are four ways to make the trip. There are two ways to make the trip if $A = 970$, $B = 1030$, and we add one stop at 4960.

Input Specification

The first two lines of the input are the minimum distance A and the maximum distance B ($1 \leq A \leq B \leq 7000$), both of which are integers. The third line of the input is an integer N ($0 \leq N \leq 20$), followed by N lines, each giving the location m of an additional eligible motel ($0 < m < 7000$). You should note that no two motels are located at the same distance from Vancouver.

Output Specification

Output the number of different ways the driver can choose the motels to make the trip, under the given constraints.

Sample Inputs and Outputs

Sample Input	Sample Output
1 500 0	0
970 1030 0	1
970 1040 0	4
970 1030 1 4960	2