



The CENTRE for EDUCATION  
in MATHEMATICS and COMPUTING

*2010  
Canadian  
Computing  
Competition:  
Junior  
Division*

Sponsor:



## 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 MUST 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. Inefficient solutions may lose marks for some problems. Be sure your code is as efficient (in terms of time) as possible.
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 Toronto area
- 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 and to see who the prize winners are. The CCC website is:

[www.cemc.uwaterloo.ca/ccc](http://www.cemc.uwaterloo.ca/ccc)

## Problem J1: What is $n$ , Daddy?

### Problem Description

Natalie is learning to count on her fingers. When her Daddy tells her a number  $n$  ( $1 \leq n \leq 10$ ), she asks “What is  $n$ , Daddy?”, by which she means “How many fingers should I hold up on each hand so that the total is  $n$ ?”

To make matters simple, her Daddy gives her the correct finger representation according to the following rules:

- the number may be represented on one or two hands;
- if the number is represented on two hands, the larger number is given first.

For example, if Natalie asks “What is 4, Daddy?”, her Dad may reply:

- 4 is 4.
- 4 is 3 and 1.
- 4 is 2 and 2.

Your job is to make sure that Natalie’s Daddy gives the correct number of answers.

### Input Specification

The input will be a single integer in the range 1..10.

### Output Specification

The output is the number of ways of producing that number on two hands, subject to the rules outlined above.

### Sample Input

4

### Output for Sample Input

3

## Problem J2: Up and Down

### Problem Description

Nikky and Byron are playing a silly game in gym class.

Nikky is told by his teacher to walk forward  $a$  steps ( $1 \leq a \leq 100$ ) and then walk backward  $b$  steps ( $1 \leq b \leq 100$ ), after which he repeats  $a$  forward,  $b$  backward, etc. Likewise, Byron is told to walk forward  $c$  steps ( $1 \leq c \leq 100$ ) and then walk backward  $d$  steps ( $1 \leq d \leq 100$ ), after which he repeats  $c$  forward,  $d$  backward, etc. You may assume that  $a \geq b$  and  $c \geq d$ .

Byron and Nikky have the same length of step, and they are required to take their steps simultaneously (that is, Nikky and Byron will both step forward on their first steps at the same time, and this will continue for each step).

Nikky and Byron start walking from one end of a soccer field. After  $s$  steps ( $1 \leq s \leq 10000$ ), the gym teacher will blow the whistle. Your task is to figure out who has moved the farthest away from the starting position when the whistle is blown.

### Input Specification

The input will be the 5 integers  $a$ ,  $b$ ,  $c$ ,  $d$ , and  $s$ , each on a separate line.

### Output Specification

The output of your program will be one of three possibilities: `Nikky` if Nikky is farther ahead after  $s$  steps are taken; `Byron` if Byron is farther ahead after  $s$  steps are taken; `Tied` if Byron and Nikky are at the same distance from their starting position after  $s$  steps are taken.

### Sample Input

```
4
2
5
3
12
```

### Output for Sample Input

```
Byron
```

### Explanation of Output for Sample Input

Notice that after 12 steps, Nikky has moved  $4 - 2 + 4 - 2$  steps, for a total of 4 steps from the starting position, whereas Byron has moved  $5 - 3 + 4$  steps, for a total of 6 steps from the starting position. Thus, Byron is ahead.

## Problem J3: Punchy

### Problem Description

In the early days of computing, instructions had to be “punched” onto rectangular cards, one instruction per card. This card deck was then fed into a card reader so the program could be read and executed. Students put elastic bands around their card deck, and, often, carried their cards in a box for fear that they would become rearranged, and thus, their program would be incorrect.

Poor Bill though...he left his cards right near a window and the wind blew his neat deck of cards all over the place, and thus his program is out of order! Bill decides to pick up the cards in some random order and then execute the program.

Write a program to read and execute the commands in Bill’s “new” program.

### Input Specification

The programming language that Bill is using has only two variables ( $A$  and  $B$ ) and seven different types of instructions.

Initially, the variables  $A$  and  $B$  contain the value 0.

There is one instruction per line. An instruction is an integer in the range 1...7, possibly followed by a variable name, which in turn is possibly followed by either a number or a variable.

In all instructions below, the variable  $X$  or  $Y$  may refer to either  $A$  or  $B$ . The specific instructions are:

- 1  $X\ n$  means set the variable  $X$  to the integer value  $n$ ;
- 2  $X$  means output the value of variable  $X$  to the screen;
- 3  $X\ Y$  means calculate  $X + Y$  and store the value in variable  $X$ ;
- 4  $X\ Y$  means calculate  $X * Y$  and store the value in variable  $X$ ;
- 5  $X\ Y$  means calculate  $X - Y$  and store the value in variable  $X$ ;
- 6  $X\ Y$  means calculate the quotient of  $X/Y$  and store the value in variable  $X$  as an integer, discarding the remainder.
- 7 means stop execution of this program.

You may assume that all division instructions do not cause a division by zero, and that all other operations (including instruction 1) do not cause the computed/stored value to be larger than 10,000 or smaller than -10,000.

(To clarify division of negative numbers,  $-3/2$  and  $3/-2$  both have quotient  $-1$  and  $-3/-2$  has quotient 1.)

### Output Specification

Your program should output the value of the indicated variables, one integer per line, until the “stop” instruction has been read in, at which time your program should stop execution.

### Sample Input (with output shown in *italics*)

```
1 A 3
1 B 4
2 B
4
2 A
3
3 A B
2 A
7
5 A A
2 A
0
2 B
4
7
```

## Problem J4: Global Warming

### Problem Description

Your task is to help scientists predict the trend of the global warming. One of the hypotheses they are considering is that over long periods of time, the average temperature follows certain cycles, but each time the cycle starts from a higher temperature level. The temperatures are measured over five-year averages, and expressed in tenths of a degree.

For example, if the following five-year averages are observed:

3, 4, 6, 4, 5, 7, 5

then we can calculate that the temperature changes first 1 tenth up, then 2 up, then 2 down, 1 up, 2 up, and 2 down. There is a cycle of changes of length three which covers all of the temperature differences:  $(+1, +2, -2)$ . In other words, if we look at the differences starting at the first position, there is a cycle of length three of the form  $(+1, +2, -2)$  followed by another cycle of length three of exactly the same form.

By way of another example, suppose the following average temperatures are observed:

3, 4, 6, 7.

In this case, there is a difference of one up, two up, then one up. We would consider the shortest cycle to be length two in this case: the cycle  $(+1, +2)$ . Notice that this cycle occurs once, followed by one truncated occurrence of exactly the same cycle.

Your task is to find the shortest such cycle from a given sequence of temperatures.

### Input Specification

The input consists of a number of test cases. Each test case starts with the number  $n$  ( $1 \leq n \leq 20$ ), representing the number of temperatures in a sequence, followed by the sequence of  $n$  temperatures. You may assume that each temperature input is an integer in the range  $-1000 \dots 1000$  inclusive. The numbers are separated by a single space. The last test case is indicated by a zero and should produce no output.

### Output Specification

For each test case, produce the length of the shortest temperature cycle. The cycle always exists, since the whole sequence could be treated as one long cycle.

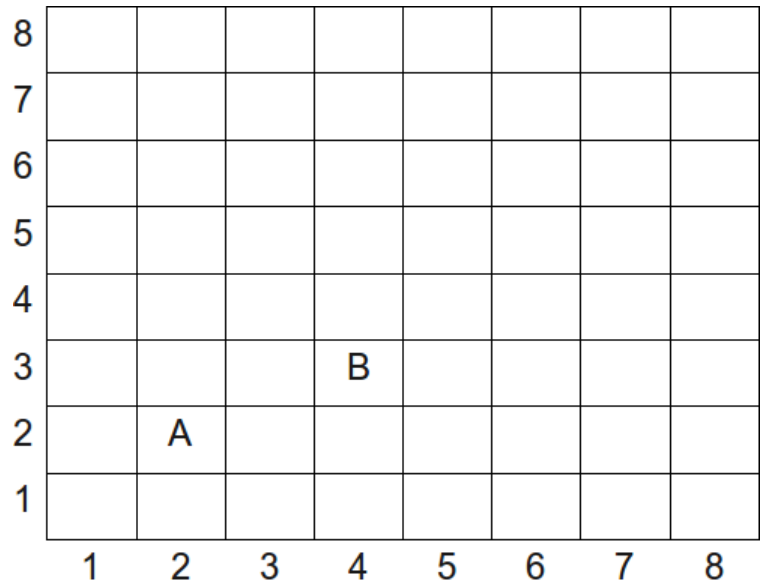
<i>Sample Input</i>	<i>Output for this test case</i>
7 3 4 6 4 5 7 5	3
3 1 3 5	1
3 1 4 5	2
4 3 4 6 7	2
0	



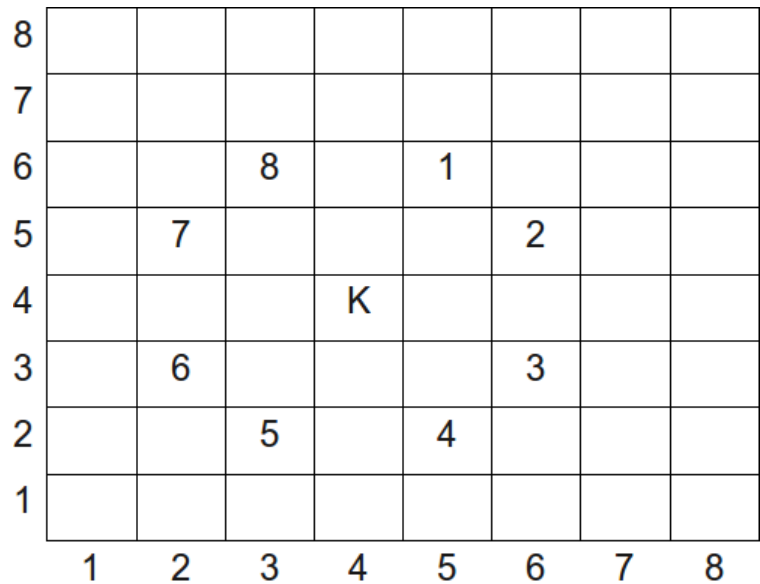
# Problem J5: Knight Hop

## Problem Description

Below is an  $8 \times 8$  chessboard on which we will designate square locations using the ordered pairs as indicated. For example, notice that piece  $A$  is at position  $(2, 2)$  and piece  $B$  is at position  $(4, 3)$ .



A knight is a special game piece that can leap over other pieces, moving in the “L” pattern. Specifically, in the diagram below,  $K$  represents the knight’s starting position and the numbers 1 through 8 represent possible places the knight may move to.



Your program will read the starting location of the knight and output the smallest number of jumps or moves needed to arrive at a location specified in the second input.

### **Input Specification**

Your program will read four integers, where each integer is in the range 1...8. The first two integers represent the starting position of the knight. The second two integers represent the final position of the knight.

### **Output Specification**

Your program should output the minimum (non-negative integer) number of moves required to move the knight from the starting position to the final position. Note that the knight is not allowed to move off the board during the sequence of moves.

### **Sample Input 1**

2 1  
3 3

### **Output for Sample Input 1**

1

### **Sample Input 2**

4 2  
7 5

### **Output for Sample Input 2**

2