Practice Problems Index Thu Oct 17 15:56:11 EDT 2019

PRACTICE PROBLEMS FOR BOSPRE 2019
-------- -------- --- ------ ----

These DO NOT COUNT and DO NOT APPEAR ON THE SCOREBOARD.

You may get help on any aspect of these problems during
the contest, including algorithm design and code, from
the host site staff, as long as they have time for you.

practice/encrypt
Child’s cipher.
Boston Preliminary 2000.
Author: Tomas Rokicki

practice/asciiart
A picture is worth a thousand moves.
Boston Preliminary 2015
Author: Bob Walton with help from the
Easier Problems Committee

practice/relativeneighbor
I’m closer than he is.
Boston Preliminary 2011
Author: Bob Walton

The Child’s Cipher
--- ------- ------

As a child, I would often encrypt my journal using a simple substitution cipher. In this cipher, vowels (a,e,i,o,u,y) would be replaced with the next vowel, and consonants would be replaced with the next consonant. This way, words still looked more or less like words. For instance, the word

cat

was encrypted as

dev

The vowel y is replaced with the vowel a, and the consonant z is replaced with the consonant b.

The encryption and decryption is simple enough to be done in your head as you write.

But as I age, I find this more difficult, so I am asking you to write a computer program that automates the process.

Input will be a sequence of text lines. Output should be the encrypted text lines. All case, punctuation, numbers, spaces, blank lines, and anything else in the input file that is not a letter should be preserved.

Sample Input
------ -----

```
*  Lambs born on days near Easter are lion friends.
<END>
He is cold, he is bold, he is not very old.

The last line was blank.
A rose is a nose with no toes.
```

Sample Output
------ ------

```
*  Menct cusp up feat pies Ietvis esi moup gsoipft.
<IPF>
Ji ot dumf, ji ot cumf, ji ot puv wisa umf.

Vji metv mopi xet cmepl.
E suti ot e puti xovj pu vuit.
```

```
File:      encrypt.txt
Author:    Tomas Rokicki <rokicki@cs.stanford.edu>
Date:      Wed Oct 16 03:29:52 EDT 2019
```

ASCII Art
----- ---

ASCII art consists of a drawing made in an ASCII text file, such as

You are to write a program that creates ASCII art from an input command line.

Input

For each of several test cases, a line containing just the test case name, followed by a single command line from which the art is to be created.

Assume your printer has a head that begins at the bottom left of a blank page of ASCII characters and can print a single ASCII character or move one position. Then each command line character is interpreted as follows:

n	move up (North) one square
e	move right (East) one square
s	move down (South) one square
w	move left (West) one square
any other character	print the character

For example, the command line ‘nn*es+’ is interpreted as move up, move up, print ‘*’, move right, move down, print ‘+’.

The page is 40 rows (lines) and 56 columns. There are two special cases. A request to move off the page is turned into a no-operation. A request to overwrite a character previously written in a page location is honored as if the previous character was never written.

Thus a command line that begins with ‘AwsB’ has the same effect as a command line that begins with just ‘B’, because the printer starts at the lower left so ‘w’ and ‘s’ become no-operations and ‘B’ overwrites ‘A’.

The command line may not be longer than 100,000 characters. Input ends with an end of file.

Output

For each test case, first an exact copy of the test case name line, and then the page with the drawing produced by the input command line.

However, blank lines at the TOP of the page MUST be omitted from the output.

Sample Input
------ -----

```
-- SAMPLE 1 --
Aws*n|n|n*e-e-e-e-e-e*s|s|s*w-w-w-w-w-
-- SAMPLE 2 -
nnnn*ne/ne/ne*se\se\se*sw/sw/sw*nw\nw\nweeXeeXsw_
-- SAMPLE 3 -
[see sample.in]
-- SAMPLE 4 -
[see sample.in]
```

Sample Output
------ ------

```
-- SAMPLE 1 --
*-----*
|     |
|     |
*-----*
-- SAMPLE 2 -
   *
  / \
 /   \
* X X *
 \ _ /
  \ /
   *

-- SAMPLE 3 -
     (
  .------.
 /        \
,          ,
|  O    O  |
.     :    .
 \         /
  \  -   /
   *---*
-- SAMPLE 4 -
[see sample.test: just '*' at 4 corners with many
 attempts to move off the page and with overwriting.]
```

```
File:      asciiart.txt
Author:    Bob Walton <walton@seas.harvard.edu>
Date:      Sun Oct 11 01:27:20 EDT 2015

The authors have placed this file in the public domain;
they make no warranty and accept no liability for this
file.
```

Relative Neighbor Graphs
-------- -------- ------

Given a set of points in a plane, the associated relative neighbor graph has an edge between two points P1 and P2 if and only if there is NO point P3 such that

    d(P3,P1) < d(P1,P2) and d(P3,P2) < d(P1,P2)

where d(Px,Py) is the distance between Px and Py.

You have been asked to compute the relative neighbor graph of a set of points.

Input

For each test case, first a line containing just the test case name, and then one line containing just

        N

the number of points. Then N lines each containing

        X Y

which describes the point (X,Y). The points are given identifiers 1, 2, ..., N in the order that their coordinate lines appear in the input. Coordinates are all integers.

        3 <= N <= 100
        -1,000,000 <= X,Y <= +1,000,000

The test case name contains at most 80 characters.
Input ends with an end of file.

Output

For each test case, first a line that is an exact copy of the test case name input line. Then one line for each edge in the relative neighbor graph, this line having the format

        i j

to specify that there is an edge from point i to point j. Here 1 <= i,j <= N.

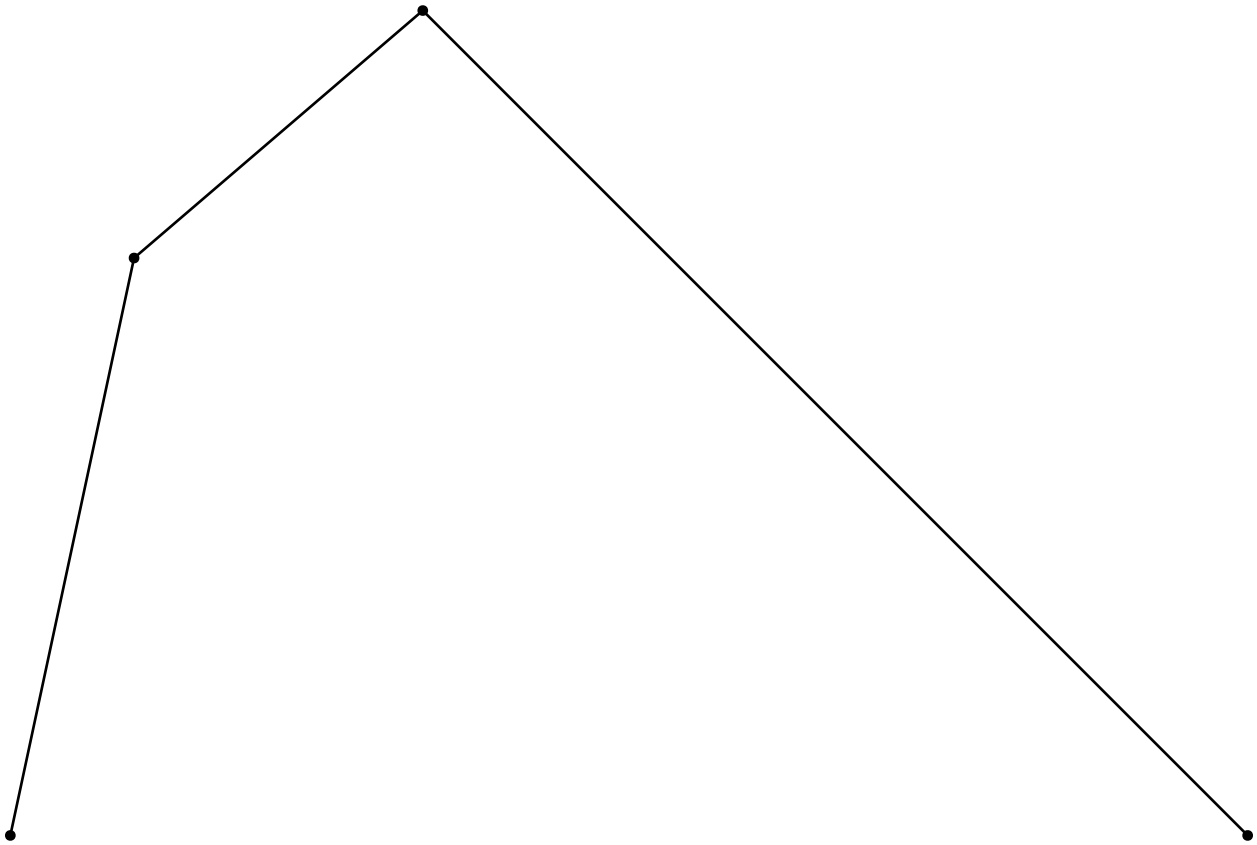
This output MUST BE SORTED so that i < j and the edges are in order of increasing i, and for equal i, in the order of increasing j.

The test case output is ended with a line containing just ‘*’.

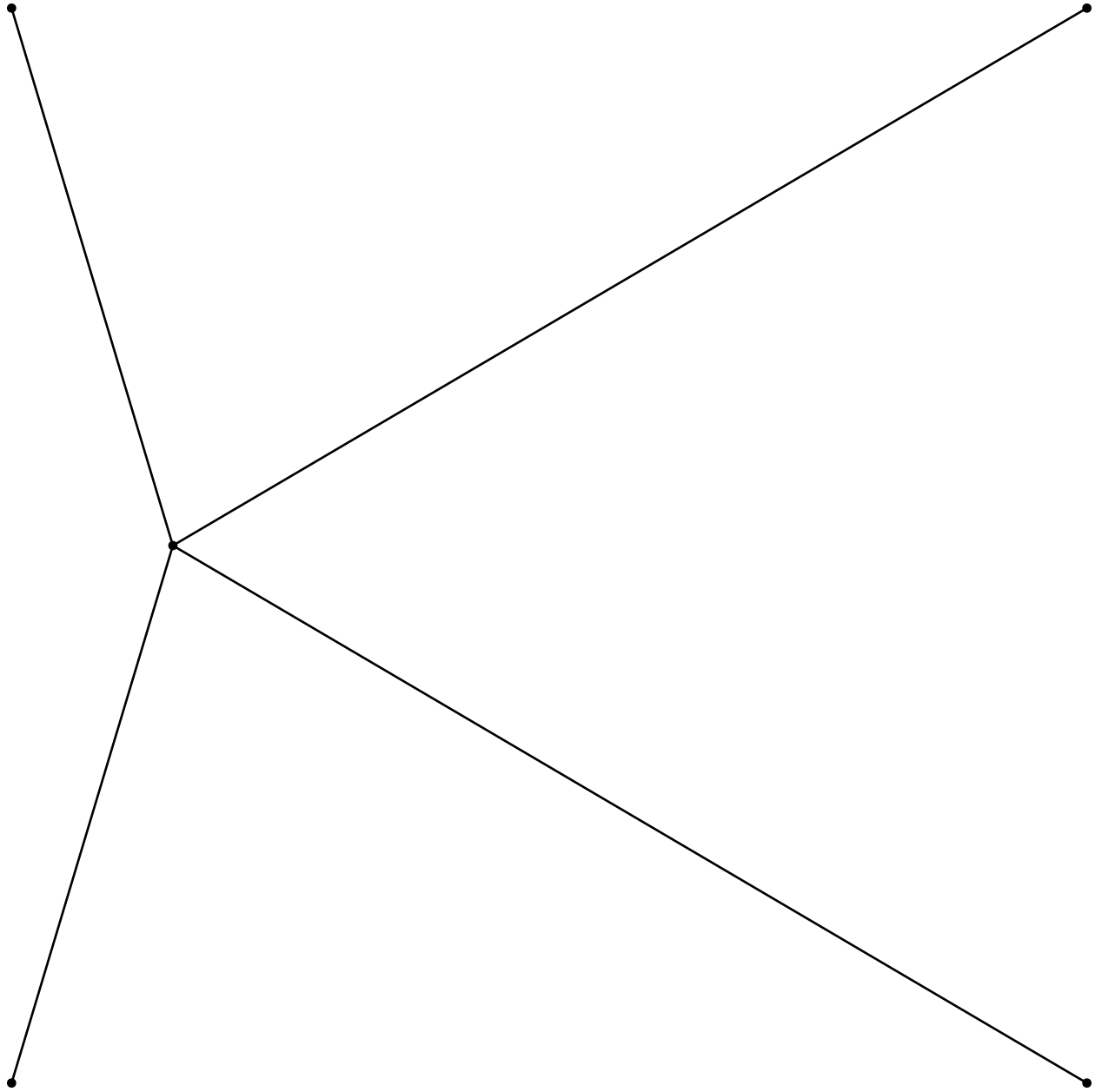
```
Sample Input
------ -----

-- SAMPLE 1 --
4
-1000     0
+2000     0
    0 +2000
 -700  1400
-- SAMPLE 2 --
5
-1000 -1000
+1000 -1000
+1000 +1000
-1000 +1000
 -700     0
-- SAMPLE 3 --
5
-1000 -1000
+1000 -1000
+1000 +1000
-1000 +1000
-1300     0
```

```
Sample Output
------ ------

-- SAMPLE 1 --
1 4
2 3
3 4
*
-- SAMPLE 2 --
1 5
2 5
3 5
4 5
*
-- SAMPLE 3 --
1 2
1 5
2 3
3 4
4 5
*


File:      relativeneighbor.txt
Author:    Bob Walton <walton@seas.harvard.edu>
           Upgraded by Bob Walton in 2017
Date:      Sat Oct  7 10:48:45 EDT 2017
           Original: Sun Oct  2 03:59:50 EDT 2011

The authors have placed this file in the public domain;
they make no warranty and accept no liability for this
file.
```

-- SAMPLE 1 --



-- SAMPLE 2 --



-- SAMPLE 3 --