adj2neg

A utility program for converting adjacency matrices NEGOPY-style files, in which there is one link per line of data.

Sometimes it is necessary to convert data that has been organized as an adjacency matrix so it can be used by NEGOPY. The adj2neg program was written to make this task easy for you.

Here are some examples of adjacency matrix formats you might encounter.

A. Some files contain pure adjacency matrices -- no ID numbers, no format specification. This is the simplest form of adjacency matrix format, since it contains only the matrix itself. The first line of data in the file contains the first row of the matrix.

```
0 1 1 0 1 0 0 0 1 0 0 0 0 1 1
0 0 1 0 1 0 1 1 1 0 0 1 0 0 0
1 0 0 0 1 0 0 0 0 0 1 0 0 0 1
1 1 0 0 0 1 1 1 0 0 0 0 0 1 0
0 1 0 1 0 1 0 1 1 0 1 0 0 1 0
           :
           :
```

B. Some files prefix each row of the adjacency matrix with an ID number. Once again, the first line of data in the file contains the first row of the matrix, but the first number in each row must be treated differently from all the other numbers.

```
1 0 1 1 0 1 0 0 0 1 0 0 0 0 1 1
2 0 0 1 0 1 0 1 1 1 0 0 1 0 0 0
3 1 0 0 0 1 0 0 0 0 0 1 0 0 0 1
4 1 1 0 0 0 1 1 1 0 0 0 0 0 1 0
5 0 1 0 1 0 1 0 1 1 0 1 0 0 1 0
             :
             :
```

C. Some files begin with a format specification that describes the data. This file is similar to the first one described above, but it has a line containing a FORTRAN-style format specification for the data preceding the actual matrix itself.

This example has a format but no ID numbers:

```
(15i2)
 0 1 1 0 1 0 0 0 1 0 0 0 0 1 1
 0 0 1 0 1 0 1 1 1 0 0 1 0 0 0
 1 0 0 0 1 0 0 0 0 0 1 0 0 0 1
 1 1 0 0 0 1 1 1 0 0 0 0 0 1 0
 0 1 0 1 0 1 0 1 1 0 1 0 0 1 0
            :
            :
```

This one has a format and ID numbers, but the ID numbers occupy the same space as any column in the matrix:

```
(16f2.0)
 1 0 1 1 0 1 0 0 0 1 0 0 0 0 1 1
 2 0 0 1 0 1 0 1 1 1 0 0 1 0 0 0
 3 1 0 0 0 1 0 0 0 0 0 1 0 0 0 1
 4 1 1 0 0 0 1 1 1 0 0 0 0 0 1 0
            :
            :
```

This example has a format and ID numbers, but the ID numbers occupy three spaces, while the columns of the matrix occupy only two spaces:

```
(i3,15f2.0)
101 0 1 1 0 1 0 0 0 1 0 0 0 0 1 1
102 0 0 1 0 1 0 1 1 1 0 0 1 0 0 0
103 1 0 0 0 1 0 0 0 0 0 1 0 0 0 1
104 1 1 0 0 0 1 1 1 0 0 0 0 0 1 0
            :
            :
```

D. Some files contain additional information before the format specification or before the data.

This example has the matrix size before the format specification:

```
 20
(i3,15f2.0)
101 0 1 1 0 1 0 0 0 1 0 0 0 0 1 1 1 0 1 0 0
102 0 0 1 0 1 0 1 1 1 0 0 1 0 1 0 1 0 0 0 0
103 1 0 0 0 1 0 0 0 1 0 1 0 0 0 0 1 0 0 0 1
104 1 1 0 0 0 1 0 1 0 0 1 1 1 0 0 0 0 0 1 0
              :
              :
```

The program can handle all of the above situations.

When it begins, it asks for the name of the data file you want to convert and the name you want to use for the new file.

It then draws a ruler on the screen, followed immediately by the first four lines of data in the file. This allows you to see what is in the file and helps in answering the following questions.

```
         1         2         3         4
1234567890123456789012345678901234567890
----------------------------------------
18  1                                   <- First four lines
(f2.0,18f2.0)                           <- of your data file
 1 0 1 1 0 0 0 0 1 1 0 0 1 0 1 0 0 0 0  <-
 1 1 0 0 0 0 0 1 1 1 0 0 1 0 1 1 1 0 1  <-

How many nodes in the matrix? (5-400)
```

1. The program then asks how many nodes are there in the network.

2. The second question is whether there is a format specification in the beginning of the file. Answer "y" if there is one and you want to use it. If there is a format specification that you don’t want to use, you should answer "n" to this question.

3. Next, the program asks if you want it to skip some lines at the beginning of the file. If you answer "y", it will ask how many lines you want it to skip. If you answered "y" to the question about format specifications, it will ask you how many lines you want it to skip before the format.

Be careful not to tell it to skip lines that include the format unless you answered "n" to question 2. If you inadverdently skip the format that you wanted the program to use, you will receive an error message and the program will cease execution.

If you want it to skip the format specification in the file, you must tell it "n" when it asks whether there is a format.

4. Now the program asks whether there is an ID number at the beginning of each line of data. If you answer "y", it will ask you how many digits wide they are. See the screen display on the next page.

5. The program will then ask if you want it to skip the ID numbers in the file. If you tell it "y", it will assume that the first row in the adjacency matrix belongs to node #1, that the second row belongs to node #2, and so on, regardless of the ID numbers in the file that contains the adjacency matrix. If you tell it "n", it will use whatever values it finds at the beginning of the lines in your data file.

```
         1         2         3         4
1234567890123456789012345678901234567890
18  1                                    <- First four lines
(f2.0,18f2.0)                            <- of your data file
 1 0 1 1 0 0 0 0 1 1 0 0 1 0 1 0 0 0     <-
 1 1 0 0 0 0 0 1 1 1 0 0 1 0 1 1 1 0 1   <-

How many nodes in the matrix? (5-400) 18

Should lines in the beginning of your file be skipped? (y/n) y
How many lines before the format do you want to skip? (1-20) 1
Does each row begin with an ID number? (y/n)
```

6. If you told the program you do not have a format specification in the file (question #2), it will ask you how many digits each column of the adjacency matrix occupies.

7. It will then ask if you want to skip spaces before the data. If you do, tell it how many spaces you want it to skip. If you don't, answer "0".

At this point the program will have all the information it needs. It will convert the data for you. When it finishes it will show the last line of data it processed. It's a good idea to compare this with the last row of the adjacency matrix in your data file.

If there was a problem with the way you instructed the program (you skipped too many lines, you said there were 22 rows when there were actually only 19, etc....) the last line processed will help you figure out what caused the problem.