Cross-Validation Nearest-Neighbors Project

Write a program to implement cross validation evaluation of the k-nearest-neighbors algorithm.

Input

The input consists of two files. The first file contains cross-validation information, and the second file contains the data.

The first file

Here is an example of the first file:

```
2 9 3
0 1 2 3 4 5 6 7 8
8 1 2 3 4 5 6 7 0
4 0 8 2 6 3 7 1 5
```

The numbers in each row are separated by a single space. The first number is the k of k-fold, to be used in the k-fold cross validation scheme. The second number is m, the number of examples. The third number is t, the number of random permutations (shuffles). The information in the above the file is: 2-fold cross-validation on 9 examples, with 3 random permutations. The rest of the lines are the t "random" permutations.

The second file

Here is an example of the second file:

```
4 5
. + . . −
. + . − −
. + . . −
. + . . −
```

The numbers and characters in each row are separated by a single space. The first line has two numbers: rows cols. This is followed by a grid of size rows by cols. Each entry in the grid is one of $\{+, -, .\}$, where "+" indicates a positive example, "−" indicates a negative example, and "." indicates the location is not an example. Thus, the above file specifies the following 9 examples:

example number	x_1	x_2	y
0	1	0	+
1	4	0	−
2	1	1	+
3	3	1	−
4	4	1	−
5	1	2	+
6	4	2	−
7	1	3	+
8	4	3	−

Observe that the examples are numbered in "raster" ordering: top-down left-right. The value of x_1 is the horizontal coordinate, and the value of x_2 is the vertical coordinate, measured downwards.

Output

Evaluate k-nearest neighbors for $k = 1, 2, 3, 4, 5$. In each case produce the following:

1. The estimate e of for the error.

2. The estimate σ of for the error standard deviation.

3. The labeling of the entire grid according to k-nearest neighbors.

For the example above, for $k = 1$ the output format should be (the numbers are most likely wrong):

```
  k=1 e=0.1 sigma=0.01
+ + + − −
+ + − − −
+ + + − −
+ + + − −
```

As previously explained, your output should be for $k = 1, 2, 3, 4, 5$.

Additional requirements

1. Your method of partitioning the permutation of size m into k-fold folds must satisfy the following:

- **a.** Each example must be in a fold.
- **b.** If m is divisible by k-fold, each fold should have the exact same number of examples.
- **c.** "Approximately" the same number of examples should be in each fold.

  For our example, with k-fold=2, $m = 9$, an acceptable partitioning is 4,5. Unacceptable is 4,4.

2. When the number of positives and negatives are the same among the nearest neighbors, select "−" as the label.

What you need to submit

Submit on eLearning the source code and executable of your program. Make sure that your submission includes both source code and binaries. (In the case of Java, the binaries should be understood as the class files.) Include an additional file explaining how to run your program.
You **must** be available to demonstrate your program to the TA. Time slots will be announced later.

Deadline:

TBA.