Create a program called “peptides” that will create a k-d tree from a peptide database (comma separated value formatted file) based on:

1. Mass
2. Normalized Elution Time **(NET)**

Your program should compute the mass of a peptide based on the amino acid characters. These values can be found in the `"aminoAcidList.csv"` file, but they are also pasted here for viewing:

Characters	Monoisotopic Value
A	71.03711
R	156.10111
N	114.04293
D	115.02694
C	103.00919
E	129.04259
Q	128.05858
G	57.02146
H	137.05891
I	113.08406
L	113.08406
K	128.09496
M	131.04049
F	147.06841
P	97.05276
S	87.03203
T	101.04768
W	186.07931
Y	163.06333
V	99.06841

You must create a hash table, including the hash function, to do so. **You may not use a map** or data structure from the standard template library.

The program should also read a file containing lists of 2-D points of mass and NET, called the *observed list*. Each item in this file will also have a number called the ID (for index).
For each item in the observed list, perform a nearest neighbor search returning the closest peptide sequence and elution time. Your program should print **(in CSV format)** to standard output the list of all found peptides. Your distance function should be a Euclidean distance based on mass and NET.

Example data files are given on the Angel site.
Your program will be run as following:

```
peptides peptideDatabase.csv observedList.csv
```

Failed inputs should say "usage: peptides databaseFile observedList-File"

Example Output (only showing one hit)

```
Observed ID, Peptide, NET, Mass, Observed NET, Observed Mass
0, AGGVGGK, 0.1494728, 523.42, .1495, 523.426
```