CMSC423 Project 1
Handed out: 10/2/2008
Due: 10/28/2008

For this project you will have to create a program that takes as input a sequence and searches for all good alignments of this sequence inside a database, using the Smith-Waterman dynamic programming algorithm described in class.

Inputs:

- one query sequence in FASTA format
- multiple sequence database in FASTA format
- similarity matrix (see format on syllabus page)
- minimum % identity
- gap opening (creation) and extension penalty (i.e. using affine gaps)

Outputs:

- precise local alignment for all hits over %identity

Sample inputs:
See syllabus page: http://www.cbcb.umd.edu/confcour/CMSC423-syllabus.shtml

Output format:
See syllabus page: http://www.cbcb.umd.edu/confcour/CMSC423-syllabus.shtml

```
>gi|90423415|ref|YP_531785.1| Gene info cytochrome c, class I [Rhodopseudomonas
palustris BisB18]

Score = 184, Identities = 49/152 (32%), Gaps = 6/152 (3%)

Query  23   LPKTRTKALLTALTLAAAAAAAPALADVEFRHAL---DDSALDLSPIKGEEITDAVKSFR  79
             P         A       A   AL    FRH     D      S   G   T AV  F
Text   3    MPSFNRSIAISATLAVGLLAPVVALGQEVFRHTVTGEDLKIMETSQPSGRD-TEAVRNFL  61
```

Note:
The output must include, for each database sequence matching the query:

- the header of the database sequence
- aggregate information:
  - Smith-Waterman score, score, number of identities and percentage (w.r.t. length aligned range within query sequence), and # and % of gaps.
- the full alignment information including:
  - the identifiers for the aligned sequences
  - coordinates along these sequences
  - gaps within the sequences

Furthermore, identical amino-acids are highlighted by repeating the identical letter in between the aligned sequences.

Input format:
Your program must accept all parameters from the command line, e.g:

myprogram -m BLOSUM.matrix -d sequence.database -i sequence.input

Submission:
Use the submit program - this project should be submitted as assignment 3:

submit 2008 fall cmsc 423 0101 3 <submission_file>

Grading! We will grade all aspects of the code, including how "pretty" it looks. Specifically pay attention to the following aspects:

1. Please make sure that your code works as advertised in the README file you provided. If your code doesn't work as indicated in the README file you will automatically lose 50% of the grade for this assignment.
2. Please provide copious comments and format your code so that it is easy to read. Part of your grade will be based on the formatting of the code.
3. Fastest programs get additional credit:
    a. 20 points for fastest
    b. 12 points for second fastest
    c. 5 points for third fastest
4. If your code does not implement affine gap penalties the maximum score you will receive is 75 and your program will not be part of the speed competition.

Please contact me and Mohammad as soon as possible if you have any questions regarding this assignment, or if you "get stuck" and might not be able to complete the assignment on time. Once the assignment is due I will no longer accept any excuses.

Important: Please copy all email to both myself and Mohammad if you want a quick reply!

Good Luck!