BIOL 4540 Sequence Analysis
Homework 2 -- Due Monday Oct 1, 2018

Starting with your DP alignment program from HW1, extend program in four ways:
(1) use BLOSUM,
(2) use affine gap penalty,
(3) do local alignment, and
(4) output alignment in blocks.

Replace the pseudocode with real code. WARNING: There may be bugs in the pseudocode! Your best defense against bugs is to understand what you are doing.

Test your code on the sequences provided under HW2 of www.bioinfo.rpi.edu/bystrc/courses/biol4540/homework.html

```plaintext
## Read input file names and parameters from stdin or command line.  
## (Don't hardwire filenames, except for debugging purposes)
#------------------------------------------------------------#
## Read sequence 1 and sequence 2, FASTA format.
  aa1 = read(seq1file)
  aa2 = read(seq2file)
#------------------------------------------------------------#
##  Read in the BLOSUM62 matrix  
##  Order is alphabetical by 1-letter code.  
##  BLOSUM(1:20,1:20) = read(blosumfile)
#------------------------------------------------------------#
## Write the function matchscore() to return the BLOSUM score
integer function matchscore(i,j)
  return BLOSUM(posit(aa1[i]), posit(aa2[j]))
## ...where aa1[i] is sequence 1, position i...
#------------------------------------------------------------#
##  Write the function posit() which returns the  
##  position of the amino acid character in constant  
##  astring="ACDEFGHIKLMNPQRSTVWY"  
##  For example posit("A") returns 1.
integer function posit(aa)
  return index(astring, aa)
  // Return 0 if aa is not an amino acid character
#------------------------------------------------------------#
## Write the function getmaxarrow(), which performs
## one step of the forward summation using affine gap penalty.

Call this function for every square in the alignment matrix starting from the upper left and ending in the lower right.

The function returns the score for that box, and also returns the "arrow", defined by fromi and fromj. Box (fromi, fromj) is the box with the maximum score going into box (i,j).

Within this function loop over all boxes in the previous column from 1 to j-1, and all boxes in the previous row from 1 to i-1. Be sure to calculate the correct gap penalties.

For example, the gap penalty for fromi=i-1 and fromj=j-2 is gop + gep

```plaintext
real function getmaxarrow(matrix, i,j, gop, gep, fromi, fromj)
For all positions in the row (i-1, j=1:j-1) and in the column (i=1:i-1, j-1) sum the matrix value and the appropriate affine gap penalty. Keep the highest score and return it as the return value. Also return fromi and fromj as the positions to trace back to. Save a tuple (fromi, fromj) in a matrix of tuples called traceback(i,j).
If zero is the highest score found, then return the score 0 and set (fromi, fromj) to (0,0). Make sure your code works properly for all values of i and j, including (1,1). Refer to Lecture 4, Slide 22 and Slide 12.
```

#------------------------------------------------------------#
## Write getmaximum()
## Find maximum value of the finished matrix() after you have run getmaxarrow() over the whole matrix. Return the position of that maximum.
(integer, integer) function getmaximum(matrix)
return (i,j) where matrix(i,j) is the maximum value.

#------------------------------------------------------------#
## Write gaps()
character function gaps(n)
return a string of length n, all "."
# Write the MAIN function. Part 1.
# Forward summation.
# alignment matrix given a BLOSUM matrix and affine gap penalties.
For i=1,sizeof(aa1)
    For j=1,sizeof(aa2)
        matrix(i,j) = matchscore(i,j) +
                     getmaxarrow(matrix,i,j,gop,gep,fromi,fromj)
        traceback(i,j) = (fromi, fromj)
# Write the MAIN function. Part 2.
# Traceback from the maximum score position to the
# beginning position (0,0). nextmatch is a list of tuples,
# each derived from traceback().
nextmatch = (fromi, fromj) = getmaximum(matrix)
i = fromi
j = fromj
while (i > 0 or j > 0)
    append (fromi, fromj) = traceback(i,j) to nextmatch
    (i,j) = (fromi, fromj)
# MAIN function. Part 3.
# Output sequences in blocks of 50.

## First, calculate the number of alignment columns.
ncolumns = sizeof(aa1) + sizeof(aa2) - sizeof(nextmatch)
## ...where sizeof(nextmatch) is the number of match columns.
## Check my math here. The total number of columns is the
## total number of characters, minus the ones that are aligned. OK?
## The number of tuples in nextmatch is the number aligned.

## Then, calculate the total number of blocks
## Make each block 50 columns.
nblocks = ceiling(ncolumns/50)

## Create two output strings for the alignment, one for
## each sequence.
k = sizeof(nextmatch)
outputstr1 = "" // empty string
outputstr2 = "" // empty string

## Walk backwards through nextmatch and append characters
## to the output strings accordingly.
## The number of characters appended can be as low
## as one and as high as the whole sequence.
## += means "append"
(fromi, fromj) = (0,0)
while (k > 0)
    (toi, toj) = nextmatch(k)
    outputstr1 += aa1[fromi+1:toi-1]
    outputstr1 += gaps(toj - fromj - 1)
    outputstr1 += aa1[toi]
    outputstr2 += aa2[fromj+1:toj-1]
    outputstr2 += gaps(toi - fromi - 1)
    outputstr2 += aa2[toj]
    k--

## Append the remaining unaligned sequence and gaps at the end.
## The remaining unaligned sequence can be the empty string.
outputstr1 += aa1[toi+1:nres1] += gaps(nres2-toj-1)
outputstr2 += gaps(nres1-toi-1) += aa2[toj+1:nres2]

## Output in blocks of 50 characters with a space line
## by printing substrings of the output strings.
Interate ibl from 1 to nblocks
    print outputstr1[ (ibl-1)*50 + 1: ibl*50)
    print outputstr2[ (ibl-1)*50 + 1: ibl*50)
    print empty line

================================================================

Files you need are in
www.bioinfo.rpi.edu/bystrc/courses/biol4540/homework.html
under HW2

Use smallseq1, smallseq2 to debug. You should get this alignment no matter
what gap penalty you pick.

seq1 EFGHIKLM
seq2 EFGHIKLM

If you didn't get this alignment, debug the program by writing out
(for example) matrix, traceback, etc. Here's my debugging output for
gop=10 and gep=5. I wrote out the BLOSUM matrix, the alignment score
and the alignment matrix after the forward sum. These are the correct
numbers for smallseq1, smallseq2 given gop=10, gep=5. i is row, j is
column.

20 lines read from blosum.dat
4  0  -2  -1  -2  0  -2  -1  -1  -1  -2  -1  -1  -1  1  0  0  -3  -2
0  9  -3  -4  -2  -3  -3  -1  -3  -1  -1  -3  -3  -3  -3  -1 -1 -1 -2 -2
-2  -3  6  2  -3  -1  -1  -4  -3  1  -1  0  -2  0  -1 -3 4 -3
-1  -4  2  5  -3  -2  0  -3  1  -3  -2  2  0  0 -1 -2 -3 -2
-2  -2  -3  -6  -3  -1  0  -3  0  0  -3  -4  -3  -3  -2 -2 -1 1 3
0  -3  -1  -2  -3  6  -2 -4 -2 -4 -3  0 -2 -2 -2 0 -2 -3 -2 -3
-2  -3  -1  0  -1  -2  8  -3  -1 -3  -1  -2  1  -2  0  0 -1 -2 -3 -2
-1  -1  -3  -3  0  -4  -3  4  -3  2  1  1 -3  -3 -3 -3 -2 -1 3 -3 -1
-1  -1 -1  1  -3  -2  -1  -3  -5 -2 -1 0 -1 1 -1 0 0 -1 -2 -3 -2
-1  -1  -3  -1  0  -3  -2  1  -1  0  -4 -2 -1 1 0 0 1 0 0 -3 -4 -2
-1  -1  -3  -2  0  -3  -2  1 -1 2 5 -2 -2 0 -1 -1 -1 -1 -1
-2  -3  -1  0  -3  0  1 -3  0 -3 -2 6 -2 0 0 1 0 -3 -4 -2
-1  -3  -1  -1  -4  -2 -2 -3 -1 -3 -2 -2 7 -1 -2 -1 -1 -2 -4 -3
Here's what you should get for the two sequences used in HW1. gop=8, gep=2

Alignment score = 97.000

FAbP human MVEPFLTKLVSSENFEDYMGEFGVHIFISVGDKM
FAbP paral FSGKYQLQSQENFEAFMKAIGLPEELIQKGKDIKGVSEIYVQNGKF

FAbP human TIRTESSFQDTKISFKLGEEFDETTADNRKVKTVITLENGSMIHVQKWL
FAbP paral KFTITAGSKVIQNEFTVGEECELETMTGEKVKTVVQLEGDNKLVVTTFKNI

FAbP human GKETTIKRVKIERMVECKMNIVSTRYEK
FAbP paral KSVTELNGDIITNTM------------------T

Assignment:

On the homework web page, click "program" and upload your program. Click "results" and upload a plain-text file (no PDF or Word files) with the following in it:
1) Alignment output for the sequences linked under "1fab" and "2fab" using gop=10 and gep=1.
2) Alignment output for the sequences linked under "hydrolase1" and "hydrolase2" using gop=5 and gep=0.5