Project 1: Pairwise global alignment with linear gap cost

This project is about global alignment of sequences using linear gap cost. The project should be done in groups of 2-3 students. Hand in via BrightSpace before **Monday, Feb 14, 16:00**. Please list the names and student IDs of all group members in your hand-in. To hand in via BrightSpace, you must enroll in a group and the group must upload their answers as an ascii or pdf file (the option to hand in will not be available until groups are formed at the first exercise session).

Problem

Consider the following substitution matrix for DNA sequences:

	A	C	G	T
A	10	2	5	2
C	2	10	2	5
G	5	2	10	2
T	2	5	2	10

Question 1: What is the optimal (here maximal) cost of an alignment of AATAAT and AAGG using the above substitution matrix and gap cost -5?

```
seq1 = 'AATAAT'
seq2 = 'AAGG'

match_scores = {'A': {'A': 10, 'T': 2, 'G': 5, 'C': 2},
                'T': {'A': 2, 'T': 10, 'G': 2, 'C': 5},
                'G': {'A': 5, 'T': 2, 'G': 10, 'C': 2},
                'C': {'A': 2, 'T': 5, 'G': 2, 'C': 10}}

def C(s1,s2, gapscore = -5):
    t_mat = np.zeros(((len(str(s1))+1),len(str(s2))+1))
    for i in range(len(t_mat)):
        t_mat[i][0] = gapscore * i
    for j in range(len(t_mat[0])):
        t_mat[0][j] = gapscore * j
    for i in range(1,len(s1)+1):
        for j in range(1,len(s2)+1):
            v1 = t_mat[i][j-1] + gapscore
            v2 = t_mat[i-1][j] + gapscore
            v3 = t_mat[i-1][j-1] + match_scores[s1[i-1]][s2[j-1]]
            t_mat[i][j] = max(v1,v2,v3)
    return t_mat

print(C(seq1,seq1))
```

The optimal cost of the alignment is 20.

Question 2: What is the optimal (here maximal) cost of an alignment of seq1.fasta and seq2.fasta using the same substitution matrix and gap cost? (You probably want to implement the algorithm for computing the cost of an optimal alignment.)

```
def read_fasta_file(filename):
    lines = []
    for l in open(filename).readlines():
        if l[0] != ">" and l[0] != ';':
            lines.append(l.strip())
    return "".join(lines).replace(' ','')

fasta1 = read_fasta_file('seq1.fasta')
fasta2 = read_fasta_file('seq2.fasta')

print(C(fasta1,fasta2))
```

Here the optimal cost of the alignment is 1346.

Question 3 (optional): What does an optimal alignment look like for the above two pairs of sequences using the given substitution matrix and gap cost -5? (you probably want to implement the algorithm for finding an optimal alignment by backtracking through the dynamic programming table.)

```
def optimal_alignment(s1,s2, gapscore = -5):
    optimal_alignment_1 = ''
    optimal_alignment_2 = ''
    current_entry = C(s1,s2, gapscore = -5)[-1][-1]
    i = 1
    j = 1
    while i < len(s1) + 1 and j < len(s2) +1:
        if current_entry ==  C(s1,s2, gapscore = -5)[-i-1][-j] + gapscore:
            optimal_alignment_1 = s1[-i] + optimal_alignment_1
            optimal_alignment_2 = '-' + optimal_alignment_2
            current_entry = C(s1,s2, gapscore = -5)[-i-1][-j]
            i+=1
        elif current_entry == C(s1,s2, gapscore = -5)[-i][-j-1] + gapscore:
            optimal_alignment_1 = '-' + optimal_alignment_1
            optimal_alignment_2 = s2[-j] + optimal_alignment_2
            current_entry = C(s1,s2, gapscore = -5)[-i][-j-1]
            j+=1
        elif current_entry == C(s1,s2, gapscore = -5)[-i-1][-j-1] + match_scores[s1[-i]][s2[-j]]:
            optimal_alignment_1 = s1[-i] + optimal_alignment_1
            optimal_alignment_2 = s2[-j] + optimal_alignment_2
            current_entry = C(s1,s2, gapscore = -5)[-i-1][-j-1]
            i+=1
            j+=1
            print(i)
    return optimal_alignment_1 + '\n' + optimal_alignment_2

print(optimal_alignment(seq1, seq2))
print(optimal_alignment(fasta1, fasta2))
```

The optimal alignment for the first pair of sequences is:

AATAAT
AA-GG-

The optimal alignment for the second pair of sequences is this:

```
Seq1:   GGCCTAAAGGCGCCGGTCTTTCGTACCCCAAAATCTCG-GCATTTTAAGATAAGTG-AGTGTTGCGTTAC
Seq2:   GGGCTAAAGGTTAGGGTCTTTCACACTAAAGAGTGGTGCGTATCGT-GGCTAA-TGTACCGCTTC-TGGT

Seq1:   ACTAGCGATCTACCGCGTCTTATACT-TAAGCG-TATGCCC-AGATCTGA-CTAATCGTGCCCCCGGATT
Seq2:   A-TCGTGGCTTA-CG-GCCAGAC-CTACAAGTACTAGACCTGAGAACTAATCTTGTCGAGCCTTC-CATT

Seq1:   AGACGGGCTTGATGGGAAAG AACAGCTCGTC---TGTT-TAC--GTATAAACAGAATCGCCTGGGTTCGC
Seq2:   -GA-GGG--TAATGGGAGAG AACATCGAGTCAGAAGTTATTCTTGTTTACGTAGAATCGCCTGGGTCCGC
```

Question 4 (optional): How many optimal alignments are for the above two pairs of sequences using the given substitution matrix and gap cost -5? Explain how you can compute the number of optimal alignments.

To test your programs you are welcome to use the two test examples in project1_examples.txt