Homework-3

Sanjana Reddy Thigulla
1002142811

1. **Substitution Matrices**

	A	G	T	C
A	1	-4	-1	-1
G	-4	1	-1	-1
T	-1	-1	1	-4
G	-1	-1	-4	1

Given that the transition mutations (A ↔ G and T ↔ C) are less common than tranversions (A ↔ T, A ↔ C, G ↔ T, and G ↔C). In the initial matrix all the mismatches are given a score of '-1'. But since a few transition mutations are less likely than the others the mismatch score should be less comparatively. So -4 has been used as the mismatch score for the less likely mutations.

2. **Global Alignment**

Needleman-Wunsch algorithm has been used for global alignment. The code takes in 2 input sequences, substitution matrix and gap penalty. It returns alignments.
An example of sequences "gata","ctac" with gap penalty -2 and match and mismatch of 1 and -1 is given. When the function is run with this input the output obtained is depicted below.

```
[0, -2, -4, -6, -8]
[-2, -1, -3, -5, -7]
[-4, -3, -2, -2, -4]
[-6, -5, -2, -3, -3]
[-8, -7, -4, -1, -3]
('GATA-', 'C-TAC')
```

Figure 1: output of the given example input sequences

3. **Local Alignment**

Smith-Waterman algorithm has been used for local alignment. The code takes in 2 input sequences, substitution matrix and gap penalty. It returns alignment tuples.
An example of sequences "gata","ctac" with gap penalty -2 and match and mismatch of 1 and -1 is given. When the function is run with this input the output obtained is depicted below.

```
[0, 0, 0, 0, 0]
[0, 0, 0, 0, 0]
[0, 0, 0, 1, 0]
[0, 0, 1, 0, 0]
[0, 0, 0, 2, 0]
('ta', 'ta')
```

Figure 2: output of the given example input sequences

4. **Custom Alignment**

   - Using my first and last name a substitution matrix for alphabets has been created where matches are given a score of 2, semi-matches (characters in first and last name) are given a score of 1 and mismatches are given -1 . The output matrix has been pretty printed and is provided in the file "10012142811S.txt"
   - after running the custom substitution matrix with "local alignment" function , a gap penalty of -2, my concatenated name ("sanjanareddy") as the first string, and the pangram "thequickbrownfoxjumpsoverthelazydog" as the second string the output tuples are:
     [('sa', 'er'), ('sa', 'yd'), ('an', 'er'), ('an', 'yd'), ('nj', 'er'), ('nj', 'yd'), ('ja', 'er'), ('ja', 'yd'), ('an', 'er'), ('an', 'yd'), ('na', 'er'), ('na', 'yd'), ('ar', 'er'), ('ar', 'yd'), ('re', 'er'), ('re', 'yd'), ('ed', 'er'), ('ed', 'yd'), ('dd', 'er'), ('dd', 'yd'), ('dy', 'er'), ('dy', 'yd')]
     matrix D for the input strings has been provided in the file "1002142811D.txt"

5. **Difficulty Adjustment** It took me 11 -15 hours for this assignment. The first three parts took less time compared to the fourth part. I could obtain the subsitution matrix but then pretty printing was a bit confusing. I had to refer about pretty printing and also about how to give double arrows,tables in latex .