**5)** (30 /100 points) You will develop an algorithm for solving protein sequence alignment which aims to find optimum matching between two amino acid sequences. Inputs are: two given strings x, y where length(x)=m and length(y)=n, and a cost matrix C which prescribes a cost for matching and mismatching characters. A *gap penalty* is the component deducted from alignment score due to the presence of a gap, i.e., matching of a letter in one sequence with a dash (space) in the other sequence. A gap penalty may be a function of the length of the gap; for example, a linear gap penalty is a constant *g* such that each inserted or deleted symbol is charged *g*; the total cost of a gap of length *L* is equal to *gL*.

**Input:**

1.Two protein strings x and y, each one is in one line of the file (at most 1000 amino acid)

2. Cost matrix (BLOSUM62 scoring matrix)

3. Linear gap penalty equal to -5 (i.e., a cost of -5 is assessed for each gap symbol)

**Output:**

1. The maximum alignment score between x and y.

2. Best alignment between x and y.

**Example:**

Input:

PLEASANTLY

MEANLY

Output:

Score: 8

PLEASANTLY

-MEA--N-LY

PLEASANTLY

-ME--AN-LY

You are expected to code this question in Java. You must have a single “.java” file named as “**name\_surname\_studentID.java**” as seen in the figure below. Do not archive! Upload directly your .java file. You must copy the values of “Blosum62.docx” into a matrix in the code. (Don’t read “Blosum62.docx” in the code!). Read the “input.txt” file, each line corresponds to a protein sequence. (Don’t upload “input.txt” file.)

|  |
| --- |
| Gurkan\_Grace\_2020510999.java |
| **public class Gurkan\_Grace\_2020510999 {**    // you can add functions here…  **public static void main(String[] args){**    //your code starts from here…  //inputs:  //Copy “Blosum62.docx” into a matrix here.  // Read “input.txt” file.  // define gap penalty  //….  //output:  // Print the maximum alignment score  // Print the best alignment of given two sequences  **}**  **}** |