## Question 2:

### Question 2.a

|  |
| --- |
| fun distance(arg1, arg2)  String str1 = cleanWhiteSpace(arg1);  String str2 = cleanWhiteSpace(arg2);  int[][] distance = new int[str1.length() + 1][str2.length() + 1];  for i from 0 to str1.lenght  distance[i][0] = i \* LAMBDA;    for j from 0 to str2.lenght  distance[0][j] = j \* LAMBDA;  for i from 0 to str1.lenght  for j from 0 to str2.lenght  int a = distance[i - 1][j] + LAMBDA;  int b = distance[i][j - 1] + LAMBDA;  //Cost calculate the cost between two char(Question b)  int c = distance[i - 1][j - 1] + cost(str1.charAt(i - 1), str2.charAt(j - 1));  distance[i][j] = minimum(a, b, c);    return distance(str1.lenght, str2.lenght)      fun cleanWhiteSpace(String arg)  String str = replaceMultipleBlankByOne(arg);  if(str.startwith(" "))  remove first charatcter;  if(str.endwith(" "))  remove last charatcter;  return str; |

### Question 2.b

The letters have been grouped in several group(may be in multiple one) with each group a similarity value.(ex: ‘o0’ group has value 2 but ‘abcdegopq0689’ has value 4) Then we can return the best value of the groups where both char are in. If they have no group in common we set a default value higher.