**BÜTWORK ANSWERS:**

1)

a) minDist(k,l)

b) maxLength(k,l)=maxLength(k-1,l-1)+1 if ak=bl

maxLength(k-1,l), maxLength(k,l-1) if ak!=bl

This function checks letter similarity in words and finds the number of similar letters.

c) *minDist(k,l*) = *minDist(k, l-1)+1*

*d) minDist(k,l*) = *minDist(k-1, l)+1*

e) *minDist(k,l*) = *minDist(k-1, l-1)+1 if ak != bl*

*minDist(k,l*) = *minDist(k-1, l-1) if ak == bl*

*f)* minDist(k,l)= *minDist(k, l-1)+1 (for (1))*

*minDist(k,l*) = *minDist(k-1, l)+1 (for (2))*

*minDist(k,l*) = *minDist(k-1, l-1)+1 if ak != bl*

*minDist(k,l*) = *minDist(k-1, l-1) if ak == bl (for(3))*

2)

a) M(i,j)

b) Product of a chain of matrices

A(i) \* A(i+1) \*A(i+2)\*….\*A(j)

Specified by a given array p(i)=dimensions of matrix A(i)

c) If the last split was at k=1, right-hand side would be M(1,1)\*M(2,n)

d) If the last split was at k=2 right-hand side would be M(1,2)\*M(3,n)

e) M(i,j)=min( M(i,k)+M(k+1, j)+(p(i-1)\*p(k)\*p(j)))