```
 #  HÜMEYRA ÇİMEN 19360859053  BURSA TEKNİK ÜNİVERSİTESİ BİLGİSAYAR MÜHENDİSLİĞİ BİOİNFORMATİK DERSİ VİZE-3
```

```
def global_alignment(seq1, seq2):
    match = 1
    mismatch = -1
    gap = -1


    row_len = len(seq1) + 1
    col_len = len(seq2) + 1

    score_matrix = [[0] * col_len for _ in range(row_len)]

    for i in range(1, row_len):
        score_matrix[i][0] = gap * i
    for j in range(1, col_len):
        score_matrix[0][j] = gap * j

    for i in range(1, row_len):
        for j in range(1, col_len):
            if seq1[i-1] == seq2[j-1]:
                score = match
            else:
                score = mismatch
            diag_score = score_matrix[i-1][j-1] + score
            up_score = score_matrix[i-1][j] + gap
            left_score = score_matrix[i][j-1] + gap
            score_matrix[i][j] = max(diag_score, up_score, left_score)

    # hizalama
    align1 = ''
    align2 = ''
    i = row_len - 1
    j = col_len - 1
    while i > 0 and j > 0:
        if score_matrix[i-1][j-1] + (match if seq1[i-1] == seq2[j-1]
else mismatch) == score_matrix[i][j]:
            align1 = seq1[i-1] + align1
            align2 = seq2[j-1] + align2
            i -= 1
            j -= 1
        elif score_matrix[i-1][j] + gap == score_matrix[i][j]:
            align1 = seq1[i-1] + align1
            align2 = '-' + align2
```

```
            i -= 1
        else:
            align1 = '-' + align1
            align2 = seq2[j-1] + align2
            j -= 1


    while i > 0:
        align1 = seq1[i-1] + align1
        align2 = '-' + align2
        i -= 1
    while j > 0:
        align1 = '-' + align1
        align2 = seq2[j-1] + align2
        j -= 1

    return align1, align2


seq1 = 'GATTACA'
seq2 = 'GCATGCU'
alignment = global_alignment(seq1, seq2)

print(alignment[0])
print(alignment[1])
```

G-ATTACA
GCA-TGCU

```
def local_alignment(s1, s2, match=4, mismatch=-2, gap=-1):

    n, m = len(s1), len(s2)
    H = [[0] * (m+1) for _ in range(n+1)]
    F = [[0] * (m+1) for _ in range(n+1)]
    max_score = float('-inf')
    max_i, max_j = None, None


    for i in range(1, n+1):
        for j in range(1, m+1):
            F[i][j] = max(F[i-1][j] + gap, H[i-1][j] + gap)
            H[i][j] = max(H[i-1][j-1] + (match if s1[i-1] == s2[j-1]
else mismatch),
```

```
                            F[i][j], H[i][j-1] + gap)
            if H[i][j] <= 0:
                H[i][j] = 0
                F[i][j] = 0


            if H[i][j] > max_score:
                max_score = H[i][j]
                max_i, max_j = i, j


    align1, align2 = '', ''
    i, j = max_i, max_j
    while H[i][j] != 0:
        if H[i][j] == H[i-1][j-1] + (match if s1[i-1] == s2[j-1] else
mismatch):
            align1 = s1[i-1] + align1
            align2 = s2[j-1] + align2
            i, j = i-1, j-1
        elif H[i][j] == F[i][j]:
            align1 = s1[i-1] + align1
            align2 = '-' + align2
            i -= 1
        else:
            align1 = '-' + align1
            align2 = s2[j-1] + align2
            j -= 1


    return align1, align2, max_score


s1 = 'KVLEFGY'
s2 = 'EQLLKALEFKL'
align1, align2, score = local_alignment(s1, s2, match=4, mismatch=-2,
gap=-1)
print(f's1: {align1}')
print(f's2: {align2}')
print(f'score: {score}')
```

s1: KVLEF
s2: KALEF
score: 14