finger print

In [1]:

```
def solve_fingerprint(seq_list, no_of_col):
  seq_dict=dict()
  for column in range(no_of_col):
    counta,countc,countt,countg=0,0,0,0
    for colseq in seq_list:
      if colseq[column]=='A':
        counta+=1
      elif colseq[column]=='T':
        countt+=1
      elif colseq[column]=='C':
        countc+=1
      elif colseq[column]=='G':
        countg+=1
    seq_dict[column]=[counta,countc,countt,countg]
  display_results(seq_dict)

def display_results(seq_dict):
  print("\tA \tC \tT \tG")
  for key in seq_dict:
    print("\n",*seq_dict[key],sep="\t")

no_of_seq=int(input("Enter the number of sequence: "))
print("Enter all the sequences")
seq_list=[]
for _ in range(no_of_seq):
  seq_list.append(list(map(str, input("").split())))
solve_fingerprint(seq_list,len(seq_list[0]))
```

```
Enter the number of sequence: 4
Enter all the sequences
A C T G A T G
A T C A G A A
A T A A G C A
A G T T A G C
 A  C  T  G

 4 0 0 0

 0 1 2 1

 1 1 2 0

 2 0 1 1

 2 0 0 2

 1 1 1 1

 2 1 0 1
```

pairwise

In [2]:

```
se1=input("Enter the first sequence::")
se2=input("Enter the second sequence::")
seq1=list(se1)
seq2=list(se2)
score=[]
def Pairwise_alignment(a,b):
 gap(a,b)
 print(a)
 print(b)
 value=0
 length=len(a)
 for i in range(0,length):
    if(a[i]==b[i]):
        score.append('1')
        value=value+1
    else:
        score.append('0')
 print(score)
 print(value)
def gap(a,b):
    if(len(a)==len(b)):
        print()
    else:
     k=int(input("enter the position to insert::"))
     if (len(a)<len(b)):
         a.insert(k,'-')
     else:
         b.insert(k,'-')
    return(a,b)

Pairwise_alignment(seq1,seq2)
```

```
Enter the first sequence::abcdefgh
Enter the second sequence::abdcdef
enter the position to insert::2
['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h']
['a', 'b', '-', 'd', 'c', 'd', 'e', 'f']
['1', '1', '0', '1', '0', '0', '0', '0']
3
```

Identity

In [3]:

```
se1=input("Enter the first sequence::")
se2=input("Enter the second sequence::")
seq1=list(se1)
seq2=list(se2)
def find_identity(a,b):
    gap(a,b)
    print(a)
    print(b)
    score=0
    length=len(a)
    total_elements=len(a)*len(b)
    for i in range(0,length):
        for j in range(0,length):
            if(a[i]==b[j]):
                score=score+1
    identity=(score/total_elements)*100
    print("Matching Score::",score)
    print("Identity of the sequences::",identity)

def gap(a,b):
    if(len(a)==len(b)):
        print()
    else:
        k=int(input("enter the position to insert gap ::"))
        if (len(a)<len(b)):
            a.insert(k,'-')
        else:
            b.insert(k,'-')
    return(a,b)


if __name__ == "__main__":
    find_identity(seq1, seq2)
```

```
Enter the first sequence::abcdefgh
Enter the second sequence::abdcdefg

['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h']
['a', 'b', 'd', 'c', 'd', 'e', 'f', 'g']
Matching Score:: 8
Identity of the sequences:: 12.5
```

regualar expression

In [4]:

```
def gen_reg_exp(seq_list, no_of_col):
    final_list=[]
    for colnum in range(no_of_col):
        collist=[]
        for colseq in seq_list:
            collist.append(colseq[colnum])
        if len(set(collist))==len(collist):

            final_list.append('x')
        else:
            if len(set(collist))==1:
                final_list.append(collist[0])
            else:
                final_list.append(''.join(set(collist)))
    display_output(final_list)

def display_output(final_list):
    print(*final_list, sep='-')



if  __name__ == '__main__':

    no_of_seq = int(input("Enter the number of sequence: "))
    print("Enter all the sequences")
    seq_list = []
    for _ in range(no_of_seq):
        seq_list.append(list(map(str, input("").split())))
    gen_reg_exp(seq_list, len(seq_list[0]))
```

```
Enter the number of sequence: 4
Enter all the sequences
A D L G A V F A L C D R Y F Q
S D V G P R S C F C E R F Y Q
A D L G R T Q L R C D R Y Y Q
A D I G Q P H S L C E R Y F Q
AS-D-LIV-G-x-x-x-x-LRF-C-ED-R-FY-YF-Q
```

similarity

In [5]:

```
sequence_one=input("Enter the first sequence: ")
sequence_two=input("Enter the second sequence: ")
how_many=int(input("How many elements for similarity condition?"))
similarities=[]
for i in range(0,how_many):
    a=input("Enter an element: ")
    c=int(input("How many elements is it similar to? "))
    similarities.append([])
    similarities[i].append(a)

    for j in range(0,c):
        b=input("What is it similar to? ")
        similarities[i].append(b)
def compare(o,t,s):
  print(o)
  print(t)
  print(s)

  score=0
  for i in range(len(o)):
    for j in range(len(s)):
      if o[i] in s[j] and t[i] in s[j] and o[i] != t[i]:
           score+=1

  similarity= (score*100)/len(o)
  return similarity
print(compare(list(sequence_one),list(sequence_two),similarities),"%")
```

```
Enter the first sequence: abcdefgh
Enter the second sequence: abdcdefg
How many elements for similarity condition?2
Enter an element: a
How many elements is it similar to? 1
What is it similar to? a
Enter an element: b
How many elements is it similar to? 2
What is it similar to? b
What is it similar to? d
['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h']
['a', 'b', 'd', 'c', 'd', 'e', 'f', 'g']
[['a', 'a'], ['b', 'b', 'd']]
0.0 %
```

percentage matching

In [ ]: