

1.Initialize the following term-incidence matrix. Process the following query: “Brutus AND Caesar AND NOT Calpurnia”

```
In [ ]: plays=['Antony and Cleopatra', 'Julius Caesar', 'The Tempest', 'Hamlet', 'Othello', 'Macbeth']
characters=['Antony', 'Brutus', 'Caesar', 'Calpurnia', 'Cleopatra', 'mercy', 'worser']

In [ ]: involvement={'Antony': '110001', 'Brutus': '110100', 'Caesar': '110111',
'Calpurnia': '010000', 'Cleopatra': '100000', 'mercy': '101111', 'worser': '101110'}

In [ ]: def andOperation(str1,str2):
    str3=''
    for i,j in zip(str1,str2):
        str3+='1' if i==j=='1' else '0'
    return str3

def orOperation(str1,str2):
    str3=''
    for i,j in zip(str1,str2):
        str3+='0' if i==j=='0' else '1'
    return str3

def notOperation(str1):
    str2=''
    for i in str1:
        str2+='1' if i=='0' else '0'
    return str2

In [ ]: Brutus_AND_Caesar = andOperation(involvement['Brutus'],involvement['Caesar'])
Brutus_AND_Caesar

Out[ ]: '110100'

In [ ]: Brutus_AND_Caesar_AND_NOT_Calpurnia=andOperation(Brutus_AND_Caesar,notOperation(involvement['Calpurnia']))
Brutus_AND_Caesar_AND_NOT_Calpurnia

Out[ ]: '100100'

In [ ]: for i,play in zip(Brutus_AND_Caesar_AND_NOT_Calpurnia,plays):
    if i=='1':
        print(play)

Antony and Cleopatra
Hamlet
```

2. Given four documents: Generate the term-document incidence matrix.

```
Doc1: Breakthrough drug for Schizophrenia

Doc2: New Schizophrenia drug

Doc3: New approach for treatment of Schizophrenia

Doc4: New hopes for Schizophrenia patients

In [ ]: Doc1=['Breakthrough', 'drug', 'for', 'Schizophrenia']
Doc2=['New', 'Schizophrenia', 'drug']
Doc3=['New', 'approach', 'for', 'treatment', 'of', 'Schizophrenia']
Doc4=['New', 'hopes', 'for', 'Schizophrenia', 'patients']
terms=['Schizophrenia', 'hopes', 'New', 'patients', 'approach', 'treatment', 'Breakthrough', 'for', 'drug', 'of']
Docs=[Doc1,Doc2,Doc3,Doc4]
```

```
In [ ]: matrix=[]
for Doc in Docs:
    row=[]
    for term in terms:
        if term in Doc:
            row.append(1)
        else:
            row.append(0)
    matrix.append(row)

In [ ]: matrix

Out[ ]: [[1, 0, 0, 0, 0, 0, 1, 1, 1, 0],
[1, 0, 1, 0, 0, 0, 0, 0, 1, 0],
[1, 0, 1, 0, 1, 1, 0, 1, 0, 1],
[1, 1, 1, 1, 0, 0, 0, 1, 0, 0]]
```

3.Construct an Inverted Index for the above specified input document collections.

```
In [ ]: Doc1=['Breakthrough', 'drug', 'for', 'Schizophrenia']
Doc2=['New', 'Schizophrenia', 'drug']
Doc3=['New', 'approach', 'for', 'treatment', 'of', 'Schizophrenia']
Doc4=['New', 'hopes', 'for', 'Schizophrenia', 'patients']
terms=['Schizophrenia', 'hopes', 'New', 'patients', 'approach', 'treatment', 'Breakthrough', 'for', 'drug', 'of']
Docs=[Doc1,Doc2,Doc3,Doc4]

def toLower(lists):
    for i in range(len(lists)):
        lists[i]=lists[i].lower()

# converting all terms in lower case
toLower(terms)
for doc in Docs:
    toLower(doc)

In [ ]: # creating inverted index
for term in terms:
    print(term,end=':')
    for i in range(len(Docs)):
        if term in Docs[i]:
            print('-> '+str(i+1),end='')
    print()

schizophrenia:-> 1-> 2-> 3-> 4
hopes:-> 4
new:-> 2-> 3-> 4
patients:-> 4
approach:-> 3
treatment:-> 3
breakthrough:-> 1
for:-> 1-> 3-> 4
drug:-> 1-> 2
of:-> 3
```

4.Construct a Sorting based Inverted Index for the above specified input document collections.

```
In [ ]: Doc1=['Breakthrough', 'drug', 'for', 'Schizophrenia']
Doc2=['New', 'Schizophrenia', 'drug']
Doc3=['New', 'approach', 'for', 'treatment', 'of', 'Schizophrenia']
Doc4=['New', 'hopes', 'for', 'Schizophrenia', 'patients']
terms=['Schizophrenia', 'hopes', 'New', 'patients', 'approach', 'treatment', 'Breakthrough', 'for', 'drug', 'of']
Docs=[Doc1,Doc2,Doc3,Doc4]

In [ ]: # function to convert terms of list in lower case
def toLower(lists):
    for i in range(len(lists)):
        lists[i]=lists[i].lower()

# converting all terms in lower case
toLower(terms)
for doc in Docs:
    toLower(doc)

# sorting the terms
terms.sort()

In [ ]: # creating inverted index
for term in terms:
    print(term,end=':')
    for i in range(len(Docs)):
        if term in Docs[i]:
            print('-> '+str(i+1),end='')
    print()

approach:-> 3
breakthrough:-> 1
drug:-> 1-> 2
for:-> 1-> 3-> 4
hopes:-> 4
new:-> 2-> 3-> 4
of:-> 3
patients:-> 4
schizophrenia:-> 1-> 2-> 3-> 4
treatment:-> 3
```

5.Process the query 'BRUTUS AND CALPURNIA' using the intersect algorithm.

```
In [ ]: def intersection(list1,list2):
    answer=[]
    len1,len2=0,0
    while len1!=len(list1) and len2!=len(list2):
        if list1[len1]==list2[len2]:
            answer.append(list1[len1])
            len1+=1
            len2+=1
        elif list1[len1]<list2[len2]:
            len1+=1
        else:
            len2+=1
    return answer

In [ ]: BRUTUS=[1,2,4,11,31,45,173,174]
CALPURNIA=[2,31,54,101]

In [ ]: print(intersection(BRUTUS,CALPURNIA))

[2, 31]

In [ ]:
```