```
###Lab: 03                                                    Date: 18.02.222
###Program No.:03

###Program Name: Write a Python Program to find Term Document Incidence Matrix
#                and Process Boolean Query.

#ANINDYA NAG
# ROLL : UG/02/BTCSE/2018/005

#CODE:1


# Collection of documents(corpus)

D1="I play tennis everyday"
D2="Everyday I go to play cricket"
D3="I like cricket and tennis"
D4="Dhoni won the world cup"
D5="I love a cup of tea everyday"


#python list

docs = [D1,D2,D3,D4,D5]
docs

     ['I play tennis everyday',
      'Everyday I go to play cricket',
      'I like cricket and tennis',
      'Dhoni won the world cup',
      'I love a cup of tea everyday']


# Gather the set of all unique terms

unique_terms = {term for doc in docs for term in doc.split()}
unique_terms

 ⮡  {'Dhoni',
      'Everyday',
      'I',
      'a',
      'and',
      'cricket',
      'cup',
      'everyday',
      'go',
      'like',
      'love',
      'of',
      'play',
      'tea',
```

```
       'tennis',
       'the',
       'to',
       'won',
       'world'}


# Construct a term-document matrix
# here as a Python dictionary for ease of interpretability

doc_term_matrix = {}

for term in unique_terms:
    doc_term_matrix[term] = []

    for doc in docs:
        if term in doc:
            doc_term_matrix[term].append(1)
        else: doc_term_matrix[term].append(0)

doc_term_matrix

     {'Dhoni': [0, 0, 0, 1, 0],
      'Everyday': [0, 1, 0, 0, 0],
      'I': [1, 1, 1, 0, 1],
      'a': [1, 1, 1, 0, 1],
      'and': [0, 0, 1, 0, 0],
      'cricket': [0, 1, 1, 0, 0],
      'cup': [0, 0, 0, 1, 1],
      'everyday': [1, 0, 0, 0, 1],
      'go': [0, 1, 0, 0, 0],
      'like': [0, 0, 1, 0, 0],
      'love': [0, 0, 0, 0, 1],
      'of': [0, 0, 0, 0, 1],
      'play': [1, 1, 0, 0, 0],
      'tea': [0, 0, 0, 0, 1],
      'tennis': [1, 0, 1, 0, 0],
      'the': [0, 0, 0, 1, 0],
      'to': [0, 1, 0, 0, 0],
      'won': [0, 0, 0, 1, 0],
      'world': [0, 0, 0, 1, 0]}


# The query to find all documents containing " " AND " "
# Is just a bitwise AND:

import numpy as np

docs_array = np.array(docs, dtype='object')

v1 = np.array(doc_term_matrix[input()])
v2 = np.array(doc_term_matrix[input()])

print(v1)
```

```
print(v2)
print('-----------')
v3 = v1 & v2
print(v3)
```

```
tennis
like
[1 0 1 0 0]
[0 0 1 0 0]
-----------
[0 0 1 0 0]
```

```
# We can now get the matching documents from our corpus with the result
[doc for doc in v3 * docs_array if doc]
```

```
['I like cricket and tennis']
```

```
# Bitwise OR to construct 'this' or 'that' queries.

v1 = np.array(doc_term_matrix[input()])
v2 = np.array(doc_term_matrix[input()])

print(v1)
print(v2)
print('-----------')
v3 = v1 | v2
print(v3)
```

```
cricket
play
[0 1 1 0 0]
[1 1 0 0 0]
-----------
[1 1 1 0 0]
```

```
# We can now get the matching documents from our corpus with the result
[doc for doc in v3 * docs_array if doc]
```

```
['I play tennis everyday',
 'Everyday I go to play cricket',
 'I like cricket and tennis']
```

```
# But Not

v1 = np.array(doc_term_matrix[input()])
v2 = np.array(doc_term_matrix[input()])

print(v1)
print(v2)
print('-----------')
```

```
v3 = ~ v2

v4 = v1 & v3
print(v4)
```

```
     cricket
     like
     [0 1 1 0 0]
     [0 0 1 0 0]
     -----------
     [0 1 0 0 0]
```

```
# We can now get the matching documents from our corpus with the result
[doc for doc in v4 * docs_array if doc]
```

```
     ['Everyday I go to play cricket']
```

✓ 0s completed at 3:27 PM