SUYASH PRATAP SINGH(181B226)

```
In [12]:  suy_1 = "breakthrough drug for schizophrenia"
          teju_2 = "new schizophrenia drug"
          vineetp_3 = "new approach for treatment of schizophrenia"
          kp_4 = "new hopes for schizophrenia patients"
```

```
In [13]:  mafia = [suy_1, teju_2, vineetp_3, kp_4]
          mafia
```

```
Out[13]:  ['breakthrough drug for schizophrenia',
           'new schizophrenia drug',
           'new approach for treatment of schizophrenia',
           'new hopes for schizophrenia patients']
```

```
In [14]:  tony = {term for doc in mafia for term in doc.split()}
          tony
```

```
Out[14]:  {'approach',
           'breakthrough',
           'drug',
           'for',
           'hopes',
           'new',
           'of',
           'patients',
           'schizophrenia',
           'treatment'}
```

```
In [15]:  inverted_index = {}

          for k, doc in enumerate(mafia):
              for term in doc.split():
                  if term in inverted_index:
                      inverted_index[term].add(k)
                  else: inverted_index[term] = {k}

          inverted_index
```

```
Out[15]:  {'breakthrough': {0},
           'drug': {0, 1},
           'for': {0, 2, 3},
           'schizophrenia': {0, 1, 2, 3},
           'new': {1, 2, 3},
           'approach': {2},
           'treatment': {2},
           'of': {2},
           'hopes': {3},
           'patients': {3}}
```

In [16]:
```python
postinglist = inverted_index['for']
postinglist
```

Out[16]: {0, 2, 3}

In [17]:
```python
def and_postings(posting1, posting2):
    t1 = 0
    t2 = 0
    result = list()
    while t1 < len(posting1) and t2 < len(posting2):
        if posting1[t1] == posting2[t2]:
            result.append(posting1[t1])
            t1 += 1
            t2 += 1
        elif posting1[t1] > posting2[t2]:
            t2 += 1
        else:
            t1 += 1
    return result
```

In [18]:
```python
pl_1 = list(inverted_index['for'])   #merge
pl_2 = list(inverted_index['new'])
and_postings(pl_1, pl_2)
```

Out[18]: [2, 3]