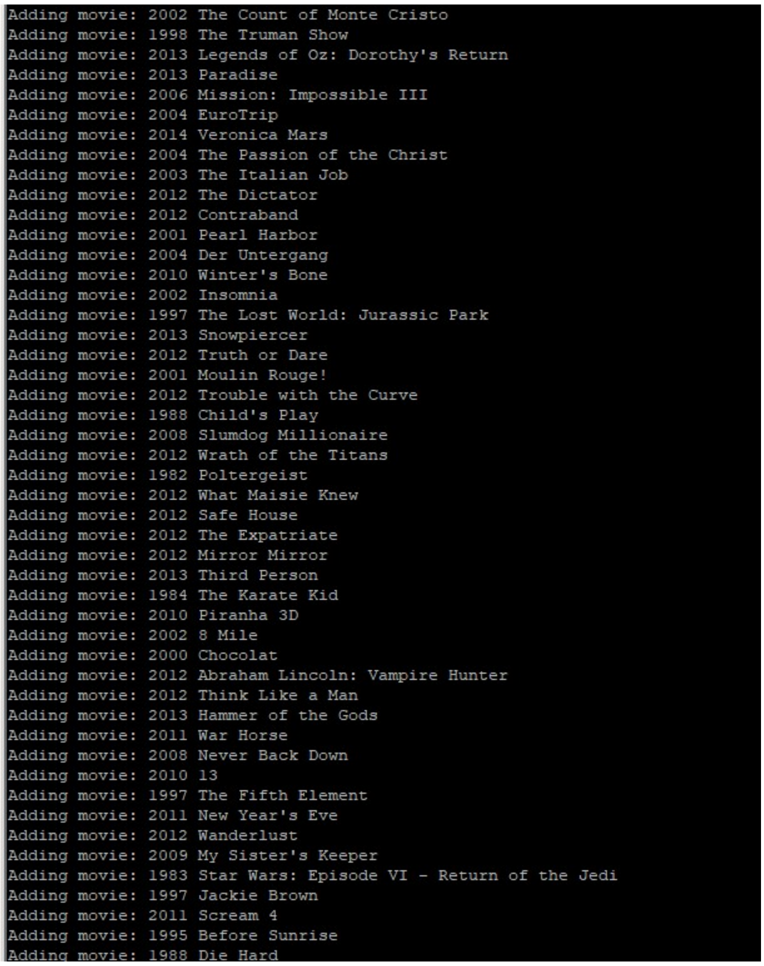
Lab6

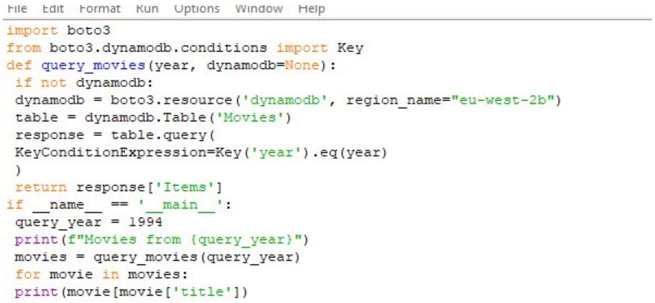
10 February 2024 19:10



```
Adding movie: 2002 The Count of Monte Cristo
Adding movie: 1998 The Truman Show
Adding movie: 2013 Legends of Oz: Dorothy's Return
Adding movie: 2013 Paradise
Adding movie: 2006 Mission: Impossible III
Adding movie: 2004 EuroTrip
Adding movie: 2014 Veronica Mars
Adding movie: 2004 The Passion of the Christ
Adding movie: 2003 The Italian Job
Adding movie: 2012 The Dictator
Adding movie: 2012 Contraband
Adding movie: 2001 Pearl Harbor
Adding movie: 2004 Der Untergang
Adding movie: 2010 Winter's Bone
Adding movie: 2002 Insomnia
Adding movie: 1997 The Lost World: Jurassic Park
Adding movie: 2013 Snowpiercer
Adding movie: 2012 Truth or Dare
Adding movie: 2001 Moulin Rouge!
Adding movie: 2012 Trouble with the Curve
Adding movie: 1988 Child's Play
Adding movie: 2008 Slumdog Millionaire
Adding movie: 2012 Wrath of the Titans
Adding movie: 1982 Poltergeist
Adding movie: 2012 What Maisie Knew
Adding movie: 2012 Safe House
Adding movie: 2012 The Expatriate
Adding movie: 2012 Mirror Mirror
Adding movie: 2013 Third Person
Adding movie: 1984 The Karate Kid
Adding movie: 2010 Piranha 3D
Adding movie: 2002 8 Mile
Adding movie: 2000 Chocolat
Adding movie: 2012 Abraham Lincoln: Vampire Hunter
Adding movie: 2012 Think Like a Man
Adding movie: 2013 Hammer of the Gods
Adding movie: 2011 War Horse
Adding movie: 2008 Never Back Down
Adding movie: 2010 13
Adding movie: 1997 The Fifth Element
Adding movie: 2011 New Year's Eve
Adding movie: 2012 Wanderlust
Adding movie: 2009 My Sister's Keeper
Adding movie: 1983 Star Wars: Episode VI - Return of the Jedi
Adding movie: 1997 Jackie Brown
Adding movie: 2011 Scream 4
Adding movie: 1995 Before Sunrise
Adding movie: 1988 Die Hard
```

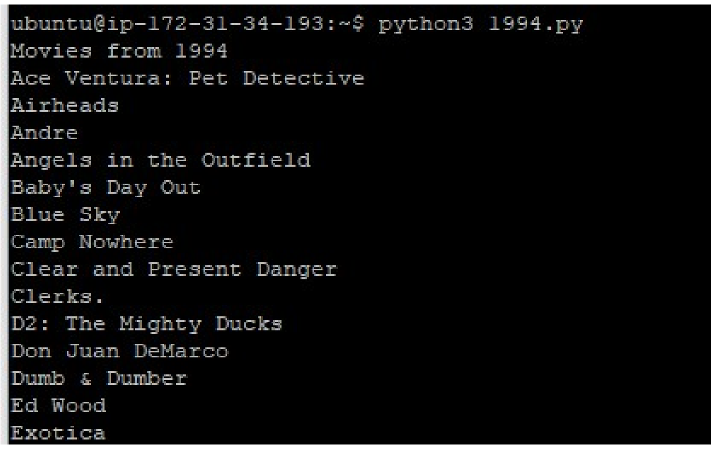
Adding movies from JSON file

Task 1 Print titles of all movies released in 1994



```python
import boto3
from boto3.dynamodb.conditions import Key
def query_movies(year, dynamodb=None):
 if not dynamodb:
 dynamodb = boto3.resource('dynamodb', region_name="eu-west-2b")
 table = dynamodb.Table('Movies')
 response = table.query(
 KeyConditionExpression=Key('year').eq(year)
 )
 return response['Items']
if __name__ == '__main__':
 query_year = 1994
 print(f"Movies from {query_year}")
 movies = query_movies(query_year)
 for movie in movies:
 print(movie[movie['title'])
```

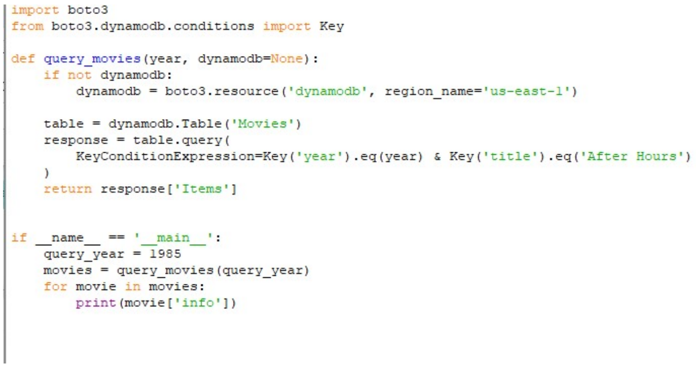
It works



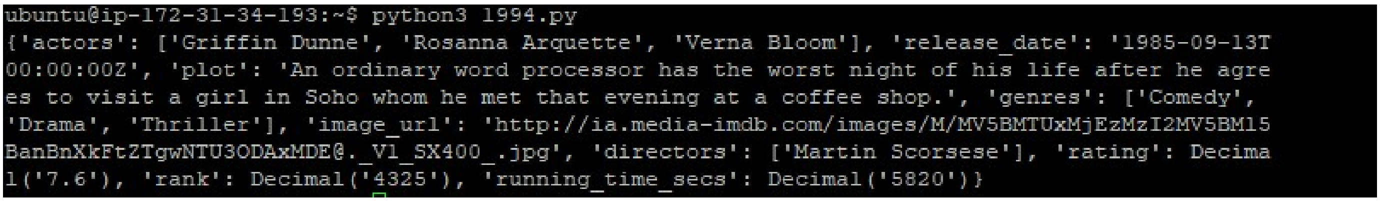
```
ubuntu@ip-172-31-34-193:~$ python3 1994.py
Movies from 1994
Ace Ventura: Pet Detective
Airheads
Andre
Angels in the Outfield
Baby's Day Out
Blue Sky
Camp Nowhere
Clear and Present Danger
Clerks.
D2: The Mighty Ducks
Don Juan DeMarco
Dumb & Dumber
Ed Wood
Exotica
```

Task 2

Get all info on After Hours 1985



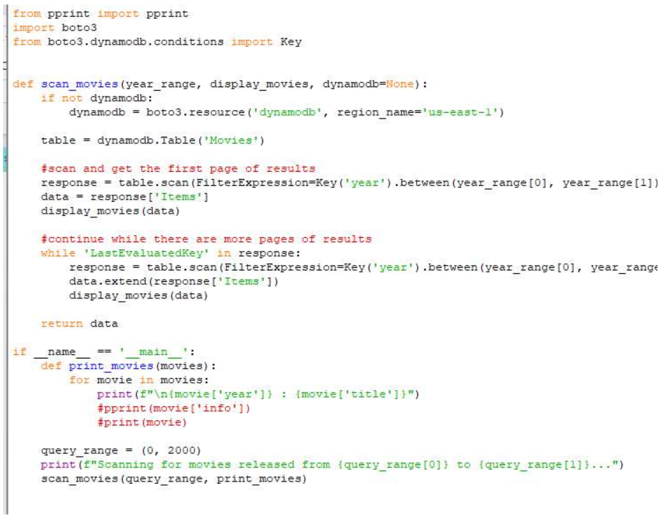
```
import boto3
from boto3.dynamodb.conditions import Key

def query_movies(year, dynamodb=None):
    if not dynamodb:
        dynamodb = boto3.resource('dynamodb', region_name='us-east-1')

    table = dynamodb.Table('Movies')
    response = table.query(
        KeyConditionExpression=Key('year').eq(year) & Key('title').eq('After Hours')
    )
    return response['Items']


if __name__ == '__main__':
    query_year = 1985
    movies = query_movies(query_year)
    for movie in movies:
        print(movie['info'])
```



```
ubuntu@ip-172-31-34-193:~$ python3 1994.py
{'actors': ['Griffin Dunne', 'Rosanna Arquette', 'Verna Bloom'], 'release_date': '1985-09-13T
00:00:00Z', 'plot': 'An ordinary word processor has the worst night of his life after he agre
es to visit a girl in Soho whom he met that evening at a coffee shop.', 'genres': ['Comedy',
'Drama', 'Thriller'], 'image_url': 'http://ia.media-imdb.com/images/M/MV5BMTUxMjEzMzI2MV5BMl5
BanBnXkFtZTgwNTU3ODAxMDE@._V1_SX400_.jpg', 'directors': ['Martin Scorsese'], 'rating': Decima
l('7.6'), 'rank': Decimal('4325'), 'running_time_secs': Decimal('5820')}
```

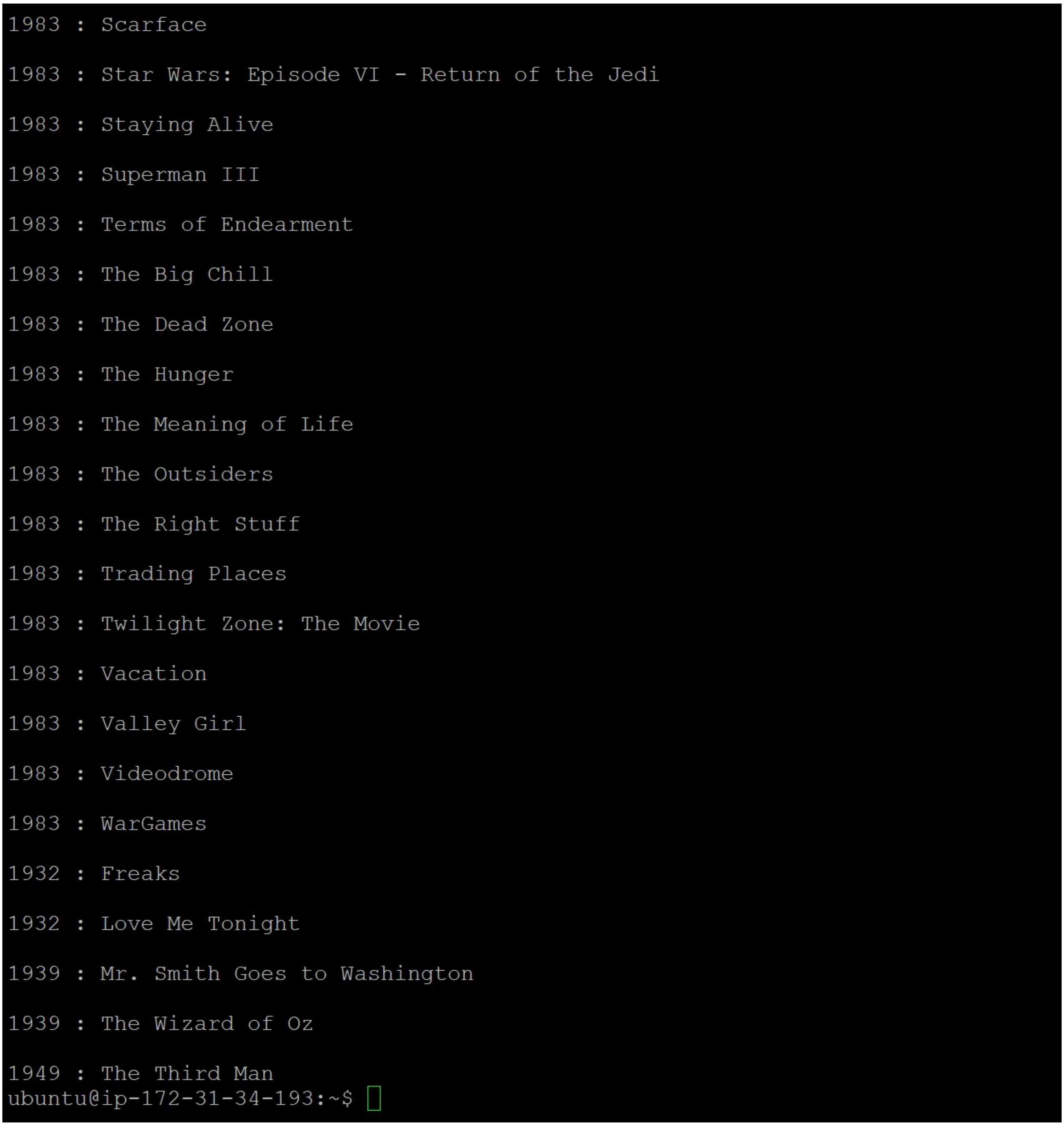
It works

Task 3



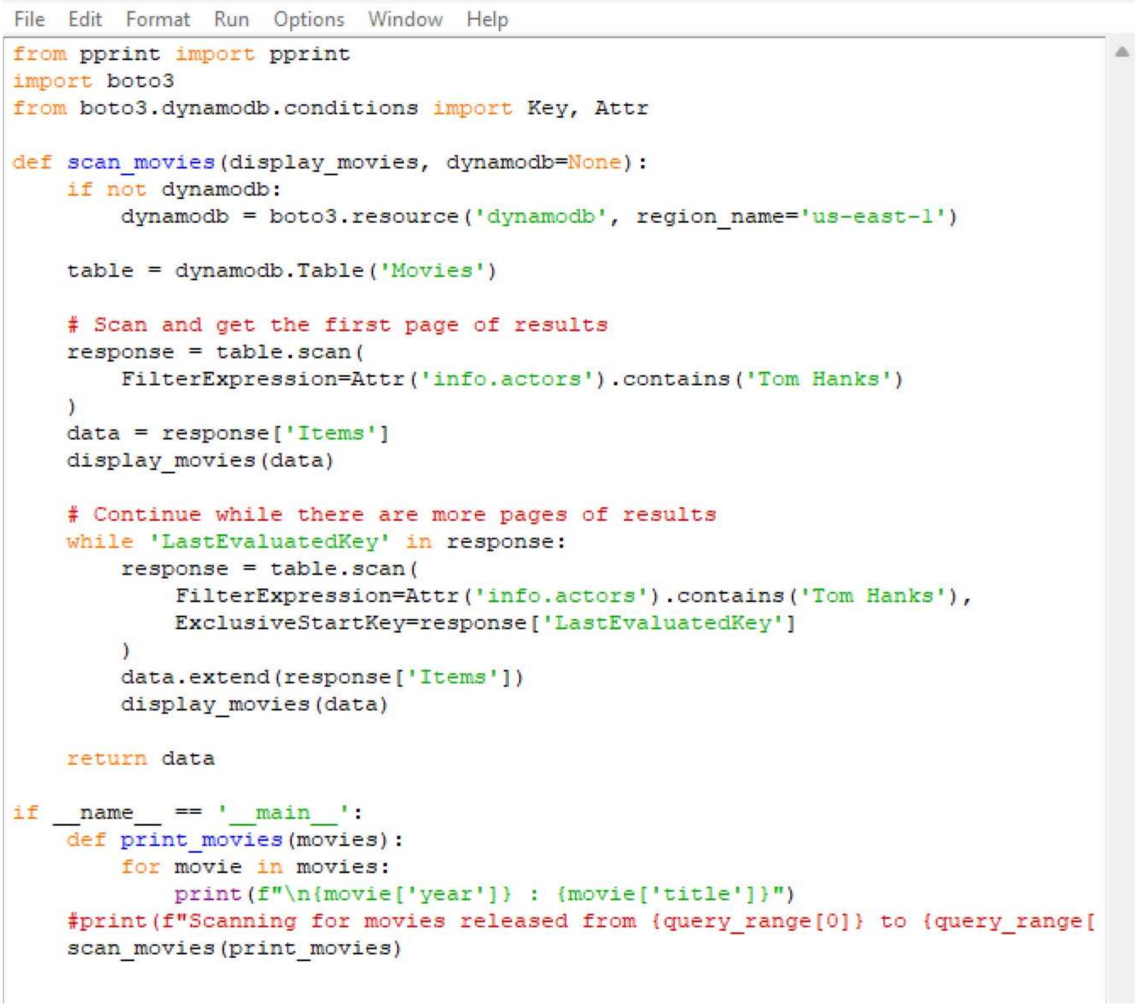
```
from pprint import pprint
import boto3
from boto3.dynamodb.conditions import Key


def scan_movies(year_range, display_movies, dynamodb=None):
    if not dynamodb:
        dynamodb = boto3.resource('dynamodb', region_name='us-east-1')

    table = dynamodb.Table('Movies')

    #scan and get the first page of results
    response = table.scan(FilterExpression=Key('year').between(year_range[0], year_range[1])
    data = response['Items']
    display_movies(data)

    #continue while there are more pages of results
    while 'LastEvaluatedKey' in response:
        response = table.scan(FilterExpression=Key('year').between(year_range[0], year_range
        data.extend(response['Items'])
        display_movies(data)

    return data

if __name__ == '__main__':
    def print_movies(movies):
        for movie in movies:
            print(f"\n{movie['year']} : {movie['title']}")
            #pprint(movie['info'])
            #print(movie)

    query_range = (0, 2000)
    print(f"Scanning for movies released from {query_range[0]} to {query_range[1]}...")
    scan_movies(query_range, print_movies)
```

It works



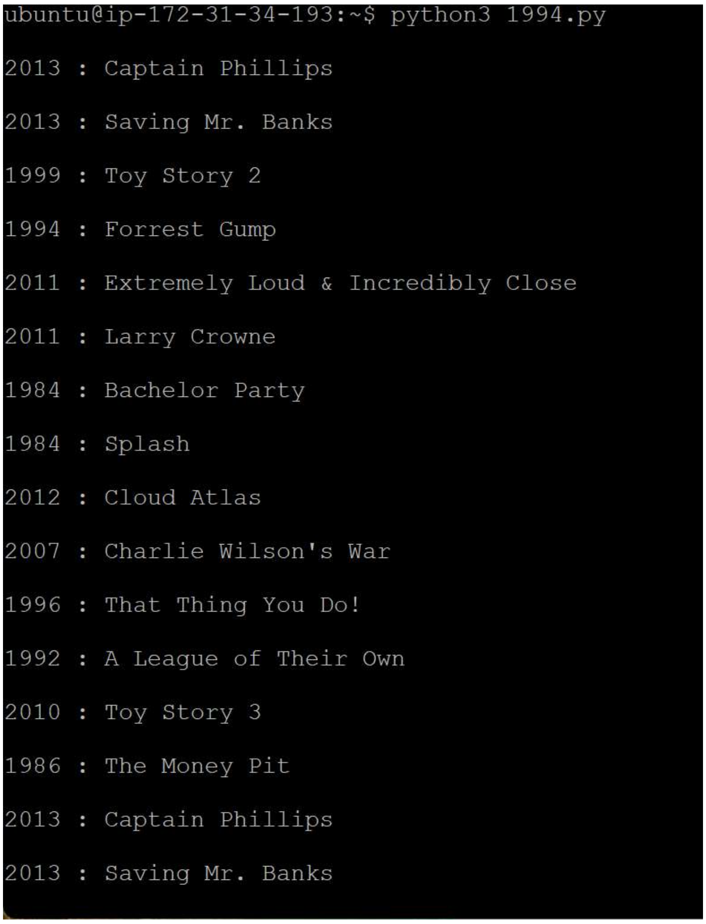
```
1983 : Scarface

1983 : Star Wars: Episode VI - Return of the Jedi

1983 : Staying Alive

1983 : Superman III

1983 : Terms of Endearment

1983 : The Big Chill

1983 : The Dead Zone

1983 : The Hunger

1983 : The Meaning of Life

1983 : The Outsiders

1983 : The Right Stuff

1983 : Trading Places

1983 : Twilight Zone: The Movie

1983 : Vacation

1983 : Valley Girl

1983 : Videodrome

1983 : WarGames

1932 : Freaks

1932 : Love Me Tonight

1939 : Mr. Smith Goes to Washington

1939 : The Wizard of Oz

1949 : The Third Man
ubuntu@ip-172-31-34-193:~$
```

Task 4



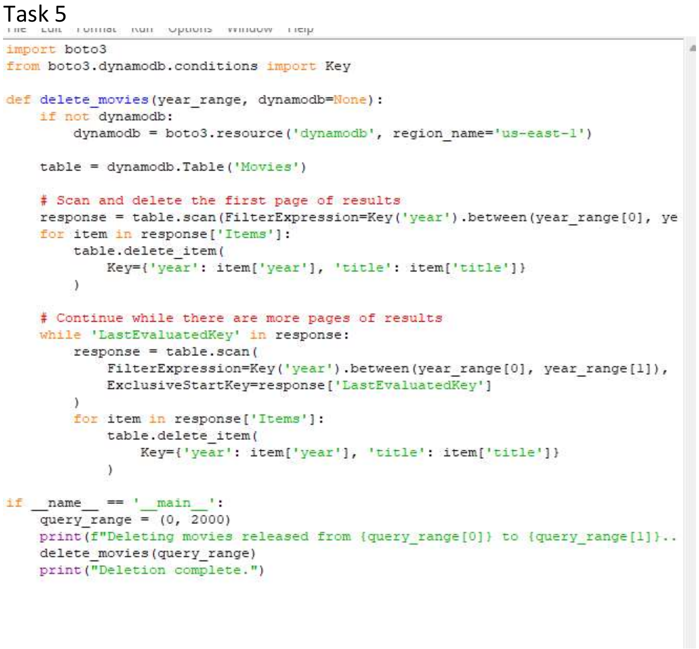
```
from pprint import pprint
import boto3
from boto3.dynamodb.conditions import Key, Attr

def scan_movies(display_movies, dynamodb=None):
    if not dynamodb:
        dynamodb = boto3.resource('dynamodb', region_name='us-east-1')

    table = dynamodb.Table('Movies')

    # Scan and get the first page of results
    response = table.scan(
        FilterExpression=Attr('info.actors').contains('Tom Hanks')
    )
    data = response['Items']
    display_movies(data)

    # Continue while there are more pages of results
    while 'LastEvaluatedKey' in response:
        response = table.scan(
            FilterExpression=Attr('info.actors').contains('Tom Hanks'),
            ExclusiveStartKey=response['LastEvaluatedKey']
        )
        data.extend(response['Items'])
        display_movies(data)

    return data

if __name__ == '__main__':
    def print_movies(movies):
        for movie in movies:
            print(f"\n{movie['year']} : {movie['title']}")
    #print(f"Scanning for movies released from {query_range[0]} to {query_range[
    scan_movies(print_movies)
```

It works:



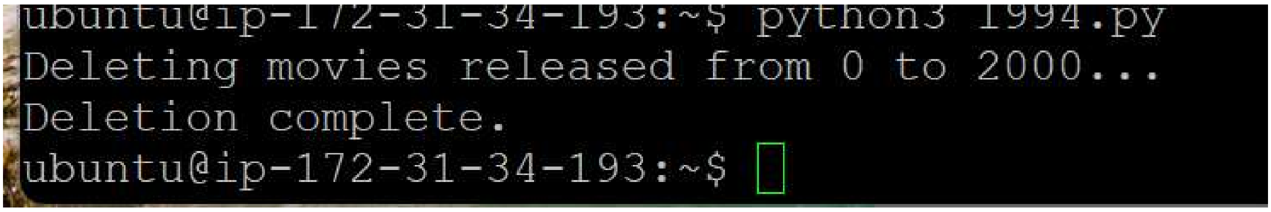
```
ubuntu@ip-172-31-34-193:~$ python3 1994.py

2013 : Captain Phillips

2013 : Saving Mr. Banks

1999 : Toy Story 2

1994 : Forrest Gump

2011 : Extremely Loud & Incredibly Close

2011 : Larry Crowne

1984 : Bachelor Party

1984 : Splash

2012 : Cloud Atlas

2007 : Charlie Wilson's War

1996 : That Thing You Do!

1992 : A League of Their Own

2010 : Toy Story 3

1986 : The Money Pit

2013 : Captain Phillips

2013 : Saving Mr. Banks
```

Task 5



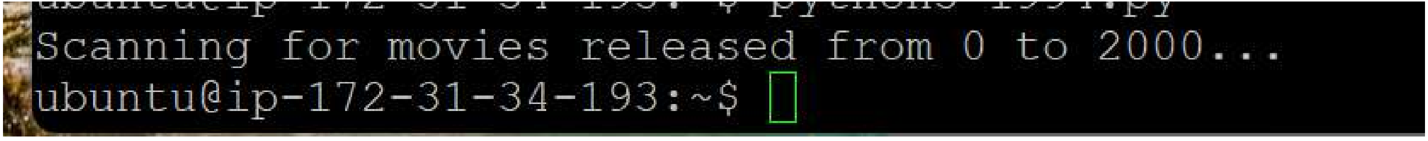
```
import boto3
from boto3.dynamodb.conditions import Key

def delete_movies(year_range, dynamodb=None):
    if not dynamodb:
        dynamodb = boto3.resource('dynamodb', region_name='us-east-1')

    table = dynamodb.Table('Movies')

    # Scan and delete the first page of results
    response = table.scan(FilterExpression=Key('year').between(year_range[0], ye
    for item in response['Items']:
        table.delete_item(
            Key={'year': item['year'], 'title': item['title']}
        )

    # Continue while there are more pages of results
    while 'LastEvaluatedKey' in response:
        response = table.scan(
            FilterExpression=Key('year').between(year_range[0], year_range[1]),
            ExclusiveStartKey=response['LastEvaluatedKey']
        )
        for item in response['Items']:
            table.delete_item(
                Key={'year': item['year'], 'title': item['title']}
            )

if __name__ == '__main__':
    query_range = (0, 2000)
    print(f"Deleting movies released from {query_range[0]} to {query_range[1]}..
    delete_movies(query_range)
    print("Deletion complete.")
```

Took a very long time to run but it works:



```
ubuntu@ip-172-31-34-193:~$ python3 1994.py
Deleting movies released from 0 to 2000...
Deletion complete.
ubuntu@ip-172-31-34-193:~$
```

Confirming movies have been removed by running query on 0-2000 movies



```
ubuntu@ip-172-31-34-193:~$ python3 1994.py
Scanning for movies released from 0 to 2000...
ubuntu@ip-172-31-34-193:~$
```

Success no movies found