Searching algorithms - Linear search

In [ ]:

```
# Input and output analysis
cards = [13,11,10,7,4,3,1,0]
query = 7
#o/p = 3
```

In [ ]:

```
tests = []
```

In [ ]:

```
tests.append({'input':{'cards':[13,11,10,7,4,3,1,0],'query':7},'output':3})
```

In [ ]:

```
tests.append({'input':{'cards':[13,11,10,7,4,3,1,0],'query':13},'output':0})
```

In [ ]:

```
tests.append({'input':{'cards':[13,11,10,7,4,3,1,0],'query':0},'output':8})
```

In [ ]:

```
tests.append({'input':{'cards':[6],'query':6},'output':0})
```

In [ ]:

```
tests.append({'input':{'cards':[13,-1,-2,-3,-10],'query':-10},'output':4})
```

In [ ]:

```
tests.append({'input':{'cards':[13,11,10,7],'query':5},'output':-1})
```

In [ ]:

```
tests.append({'input':{'cards':[],'query':5},'output':-1})
```

In [ ]:

```
tests.append({'input':{'cards':[13,13,13,13,11,10,7,7,7,7,4,4,3,1,0],'query'
```

In [ ]:

```
tests
```

Out[ ]:

```
[{'input': {'cards': [13, 11, 10, 7, 4, 3, 1, 0], 'query': 7}, 'output': 3},
 {'input': {'cards': [13, 11, 10, 7, 4, 3, 1, 0], 'query': 13}, 'output':
0},
 {'input': {'cards': [13, 11, 10, 7, 4, 3, 1, 0], 'query': 0}, 'output': 8},
 {'input': {'cards': [6], 'query': 6}, 'output': 0},
 {'input': {'cards': [13, -11, -10, -7, -4, -3, -1, 0], 'query': -4},
  'output': 4},
 {'input': {'cards': [13, -1, -2, -3, -10], 'query': -10}, 'output': 4},
 {'input': {'cards': [13, 11, 10, 7], 'query': 5}, 'output': -1},
 {'input': {'cards': [], 'query': 5}, 'output': -1},
 {'input': {'cards': [13, 13, 13, 13, 11, 10, 7, 7, 7, 7, 4, 4, 3, 1, 0],
   'query': 10},
  'output': 5}]
```

In [ ]:

```
#Linear search = Brute force

cards = [50,30,10,3,2,0,-1,-2,-3]
query = -2


def locate_card(cards, query):
    pos = 0
    length = len(cards)

    for pos in range(length+1): # +1 -> traverse the whole list and to find
        if pos <= (length-1): # looks for number within the list length
            if query == cards[pos]:
                return pos
            else:
                continue
        else:                    # anything above the list length is not a m
```

```
            return -1

locate_card(cards, query)
```

Out[ ]:

```
7
```

In [ ]:

```
'''
cards = [50,30,10,3,2,0,-1,-2,-3]
query = -2

def linear_searching(cards,query):
    pos = 0
    while True:
        if cards[pos] == query:
            print(f"The Card {cards[pos]} is at {pos}")
        pos += 1

        if pos == len(cards):
            print("Card unavailable")

linear_searching(cards,query)
'''
```

In [ ]:

```
for test in tests:
    if(locate_card(**test['input']) == test['output']):
        print("Pass!")
```

```
Pass!
Pass!
Pass!
Pass!
Pass!
Pass!
Pass!
Pass!
```