Disk Queue:
54, 95, 180, 34, 119, 11, 123, 62, 64 ,76

Head at 50
Tail at 199

FCFS:

Total Tracks = (54 - 50) + (95-54) + (180-95) + (180-34) + (119 - 34) + (119 - 11) + (123 - 11)
+ (123 - 62) + (64 - 62) + (76 - 64)
= 4 + 41 + 85 + 61 + 85 + 108 + 112 + 61 + 2 + 12
= 656

SSTF:

Total_tracks = 299

1. 54 is closer to 50,
total_tracks = 54 - 50 = 4 , head = 54, queue = [95, 180, 34, 119, 11, 123, 62, 64 ,76]

2. 62 is closer to 54
total_tracks = 4 + (62-54) = 12, head = 62, queue = [95, 180, 34, 119, 11, 123, 64 ,76]

3. 64 is closer to 62
total_tracks = 12 + (64-62) = 14 , head = 64 , queue = [95, 180, 34, 119, 11, 123, 76]

4.76 is closer to 64
total_tracks = 14 + (76 - 64) = 26, head = 76, queue = [95, 180, 34, 119, 11, 123]

5. 95 is closer to 76
total_tracks = 26 + (95 - 76) = 45, head = 95, queue = [180, 34, 119, 11, 123]

6. 119 is closer to 95
total_tracks = 45 + (119 - 95) = 69, head = 119, queue = [180, 34, 11, 123]

7. 123 is closer to 119
total_tracks = 69 + (123-119) = 73, head = 123, queue = [180,34,11]

8. 180 is closer to 123
total_tracks = 73 + (180 - 123) = 130, head = 180, queue = [34,11]

9. 34 is closer to 180
total_tracks = 130 + (180 - 34) = 276, head = 34, queue = [11]

10. 11 is closer to 34
total_tracks = 276 + (34-11) = 299, head = 11, queue = []

SCAN Scheduling:

Serviced 34,11 and hits 0
Reverse direction services 54,62,64,76,95,119,123,180

total_track = 50 + 180 = 230

C-SCAN Scheduling:
Serviced 54,62,64,76,95,119,123,180
Go Back to 0 and serviced 11,34
total_track = 180 - 50 + 34 = 164

Look Scheduling:
Serviced 34,11 and reverses
Reverse direction services 54,62,64,76,95,119,123,180
total_track = 50-11 + (180 -11) = 39 + 169 = 208

C-Look Scheduling:
Serviced 54,62,64,76,95,119,123,180
Go Back to 0 and serviced 11,34
total_track = 180 - 50 + 34 = 164

Programs:

FCFS:

```
def find_total_tracks(queue,head,tail=0):
    total_track = 0
    for track in queue:
        total_track += abs(track-head)
        head = track
    print total_track

def t1():
    queue = [98, 183, 37, 122, 14, 124, 65, 67]
    head = 53
    find_total_tracks(queue,head)

def t2():
    queue = [54, 95, 180, 34, 119, 11, 123, 62, 64 ,76 ]
    head = 50
    tail = 199
    find_total_tracks(queue,head,tail)

t2()
```

SSTF:

```
def find_closest(queue,head):
    min = 100000
    min_track = -1
    for track in queue:
        if abs(head-track) < min:
            min = abs(head-track)
            min_track = track
    return min_track
```

```
def find_total_tracks(queue,head,tail=0):
    total_track = 0
    while(len(queue) > 0):
        min_track = find_closest(queue,head)
        total_track += abs(head-min_track)
        print(min_track, total_track)
        head = min_track
        queue.remove(min_track)
    print(total_track)

def t1():
    queue = [98, 183, 37, 122, 14, 124, 65, 67]
    head = 53
    find_total_tracks(queue,head)

def t2():
    queue = [54, 95, 180, 34, 119, 11, 123, 62, 64 ,76 ]
    head = 50
    tail = 199
    find_total_tracks(queue,head,tail)

#t1()
t2()
```

SCAN:

```
def has(queue,element):
    for i in queue:
        if i == element:
            return True
    return False

def find_total_tracks(queue,head,tail=0):
    total_track = 0
    i_head = head
    while(head > 0):
        if has(queue,head):
            total_track += abs(head-i_head)
            print("Serviced ", head)
            queue.remove(head)
            i_head = head
        head = head - 1
    i_head = 0
    while (len(queue) > 0):
        if has(queue, head):
            total_track += abs(head-i_head)
            i_head = head
            print("Serviced ", head)
            queue.remove(head)
        head = head + 1
    print(total_track)

def t1():
    queue = [98, 183, 37, 122, 14, 124, 65, 67]
    head = 53
```

```
    find_total_tracks(queue,head)

def t2():
    queue = [54, 95, 180, 34, 119, 11, 123, 62, 64 ,76 ]
    head = 50
    tail = 199
    find_total_tracks(queue,head,tail)

#t1()
t2()
```