**Experiment No: 04**

**Aim:** **To study and implement FIFO and LRU Page Replacement policy.**

**Code:**

**FIFO**

from queue import Queue

def pageFaults(incomingStream, n, frames):

    print("Incoming \t pages")

    s = set()

    queue = Queue()

    page\_faults = 0

    for i in range(n):

        if len(s) < frames:

            if incomingStream[i] not in s:

                s.add(incomingStream[i])

                page\_faults += 1

                queue.put(incomingStream[i])

        else:

            if incomingStream[i] not in s:

                val = queue.queue[0]

                queue.get()

                s.remove(val)

                s.add(incomingStream[i])

                queue.put(incomingStream[i])

                page\_faults += 1

        print(incomingStream[i], end="\t\t")

        for q\_item in queue.queue:

            print(q\_item, end="\t")

        print()

    return page\_faults

incomingStream = [7, 0, 1, 2, 0, 3, 0, 4, 2, 3, 0, 3, 2, 1]

n = len(incomingStream)

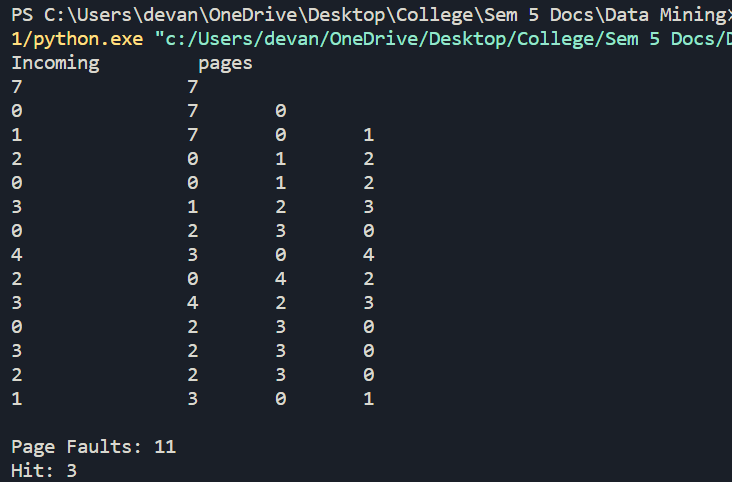
frames = 3

page\_faults = pageFaults(incomingStream, n, frames)

hits = n - page\_faults

print("\nPage Faults: " + str(page\_faults))

print("Hit: " + str(hits))

****

**LRU:**

def pageFaults(pages\_seq, n, C):

  s = set()

  indexes = {}

  faults = 0

  for i in range(n):

    if len(s) < C:

      if pages\_seq[i] not in s:

        s.add(pages\_seq[i])

        faults += 1

      indexes[pages\_seq[i]] = i

    else:

      if pages\_seq[i] not in s:

        lru = float('inf')

        for p in s:

          if indexes[p] < lru:

            lru = indexes[p]

            v = p

        s.remove(v)

        s.add(pages\_seq[i])

        faults += 1

      indexes[pages\_seq[i]] = i

    print('s =', s)

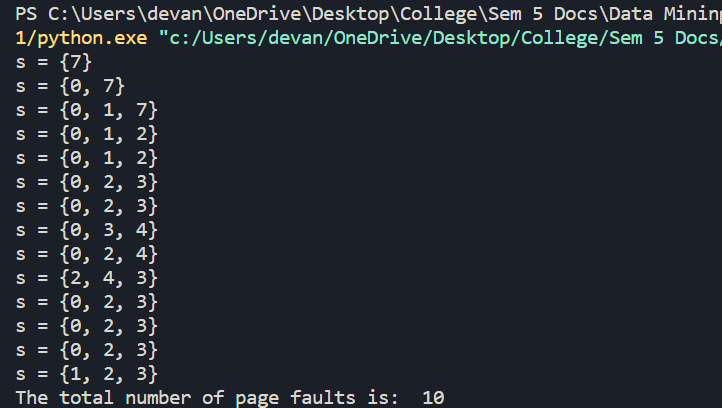
  return faults

pages\_seq = [7,0,1,2,0,3,0,4,2,3,0,3,2,1]

n = len(pages\_seq)

C = 3

print('The total number of page faults is: ', pageFaults(pages\_seq, n, C))

****