Subject: Process Organization and Architecture **SAP ID: 60004220253 – Devansh Mehta**

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))
```

```
PS C:\Users\devan\OneDrive\Desktop\College\Sem 5 Docs\Data Mining>
1/python.exe "c:/Users/devan/OneDrive/Desktop/College/Sem 5 Docs/D
Incoming         pages
7                7
0                7       0
1                7       0       1
2                0       1       2
0                0       1       2
3                1       2       3
0                2       3       0
4                3       0       4
2                0       4       2
3                4       2       3
0                2       3       0
3                2       3       0
2                2       3       0
1                3       0       1

Page Faults: 11
Hit: 3
```

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))
```

```
PS C:\Users\devan\OneDrive\Desktop\College\Sem 5 Docs\Data Mining
1/python.exe "c:/Users/devan/OneDrive/Desktop/College/Sem 5 Docs/
s = {7}
s = {0, 7}
s = {0, 1, 7}
s = {0, 1, 2}
s = {0, 1, 2}
s = {0, 2, 3}
s = {0, 2, 3}
s = {0, 3, 4}
s = {0, 2, 4}
s = {2, 4, 3}
s = {0, 2, 3}
s = {0, 2, 3}
s = {0, 2, 3}
s = {1, 2, 3}
The total number of page faults is:  10
```