Ex. No.: 11a)

FIFO PAGE REPLACEMENT

Aim:

To find out the number of page faults that occur using First-in First-out (FIFO) page replacement technique.

Algorithm:

1. Declare the size with respect to page length
2. Check the need of replacement from the page to memory
3. Check the need of replacement from old page to new page in memory 4.
   Form a queue to hold all pages
5. Insert the page require memory into the queue
6. Check for bad replacement and page fault
7. Get the number of processes to be inserted
8. Display the values

Program Code:

```
def fifo_page_replacement():
    reference = []
    size = int(input("Enter the size of reference string: "))
    for i in range(size):
        ref = int(input(f"Enter [{i+1}] : "))
        reference.append(ref)
    frame_size = int(input("Enter page frame size : "))
    frames = []
    page_faults = 0

    print()
    for page in reference:
        if page not in frames:
            if len(frames) < frame_size:
                frames.append(page)
            else:
                frames.pop(0)
                frames.append(page)
            page_faults += 1
            print(f"{page} ->", ' '.join(map(str, frames)).ljust(10))
        else:
            print(f"{page} -> No Page Fault")
    print(f"\nTotal page faults: {page_faults}.")

fifo_page_replacement()
```

Sample Output:

```
[root@localhost student]# python fifo.py

Enter the size of reference string: 20
Enter [ 1] : 7
Enter [ 2] : 0
Enter [ 3] : 1
Enter [ 4] : 2
Enter [ 5] : 0
Enter [ 6] : 3
Enter [ 7] : 0
Enter [ 8] : 4
Enter [ 9] : 2
Enter [10] : 3
Enter [11] : 0
Enter [12] : 3
Enter [13] : 2
Enter [14] : 1
Enter [15] : 2
Enter [16] : 0
Enter [17] : 1
Enter [18] : 7
Enter [19] : 0
Enter [20] : 1

Enter page frame size : 3

        7
-> 7 - -  0
-> 7 0 -  1
-> 7 0 1
        2 -> 2 0 1
        0 -> No Page Fault


        3 -> 2 3 1
        0 -> 2 3 0
        4 -> 4 3 0
2 -> 4 2 0
3 -> 4 2 3
        0 -> 0 2 3
        3 -> No Page Fault
        2 -> No Page
Fault   1 -> 0 1 3
        2 -> 0 1 2
```

0 -> No Page Fault
1 -> No Page Fault
7 -> 7 1 2
0 -> 7 0 2
1 -> 7 0 1 Total page faults: 15.
[root@localhost student]#

Output:

Enter the size of reference string: 20
Enter [1] : 7
Enter [2] : 0
Enter [3] : 1
Enter [4] : 2
Enter [5] : 0
Enter [6] : 3
Enter [7] : 0
Enter [8] : 4
Enter [9] : 2
Enter [10] : 3
Enter [11] : 0
Enter [12] : 3
Enter [13] : 2
Enter [14] : 1
Enter [15] : 2
Enter [16] : 0
Enter [17] : 1
Enter [18] : 7
Enter [19] : 0
Enter [20] : 1
Enter page frame size : 3

7 -> 7
0 -> 7 0
1 -> 7 0 1
2 -> 0 1 2
0 -> No Page Fault
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 -> No Page Fault
2 -> No Page Fault
1 -> 3 0 1
2 -> 0 1 2
0 -> No Page Fault

1 -> No Page Fault
7 -> 1 2 7
0 -> 2 7 0
1 -> 7 0 1

Total page faults: 15.

Result :

The program to find the number of page faults that occur using First-in First-out (FIFO) page replacement technique has been implemented successfully and the output has been verified.