Experiment No. 8

Title: LRU page replacement algorithm

Batch: B2 **Roll No: 16010420117** **Experiment No: 8**

Aim: To implement Least Recently Used (LRU) page replacement algorithm.

Resources needed: Text editor and JAVA/C compiler.

Theory:

Pre lab/Prior concepts:

Algorithm:

1. Accept the number of frames from user in n.
2. Accept the size of the reference string in m.
3. Read reference string in to array s.
4. Take a stack of maximum size=n.
5. Read one page number from s and enter it in stack. Increment stack pointer.
6. Read second page number from s. compares it with page number existing in stack. If matches then hit++ and goto step 7 else goto step 8.
7. Remove that page from stack. Shift all the content of stack down by one. And put the current page on stack top.(least recently used page will be at the bottom of stack)
8. If stack is empty(frame is free) then insert current page on the stack top else shift the content of the stack down by one and insert current page on stack top.
9. Print the current stack.
10. Repeat steps 6 to 9 till all the content of reference string is processed
11. Calculate and print Hit ratio.
12. Stop.

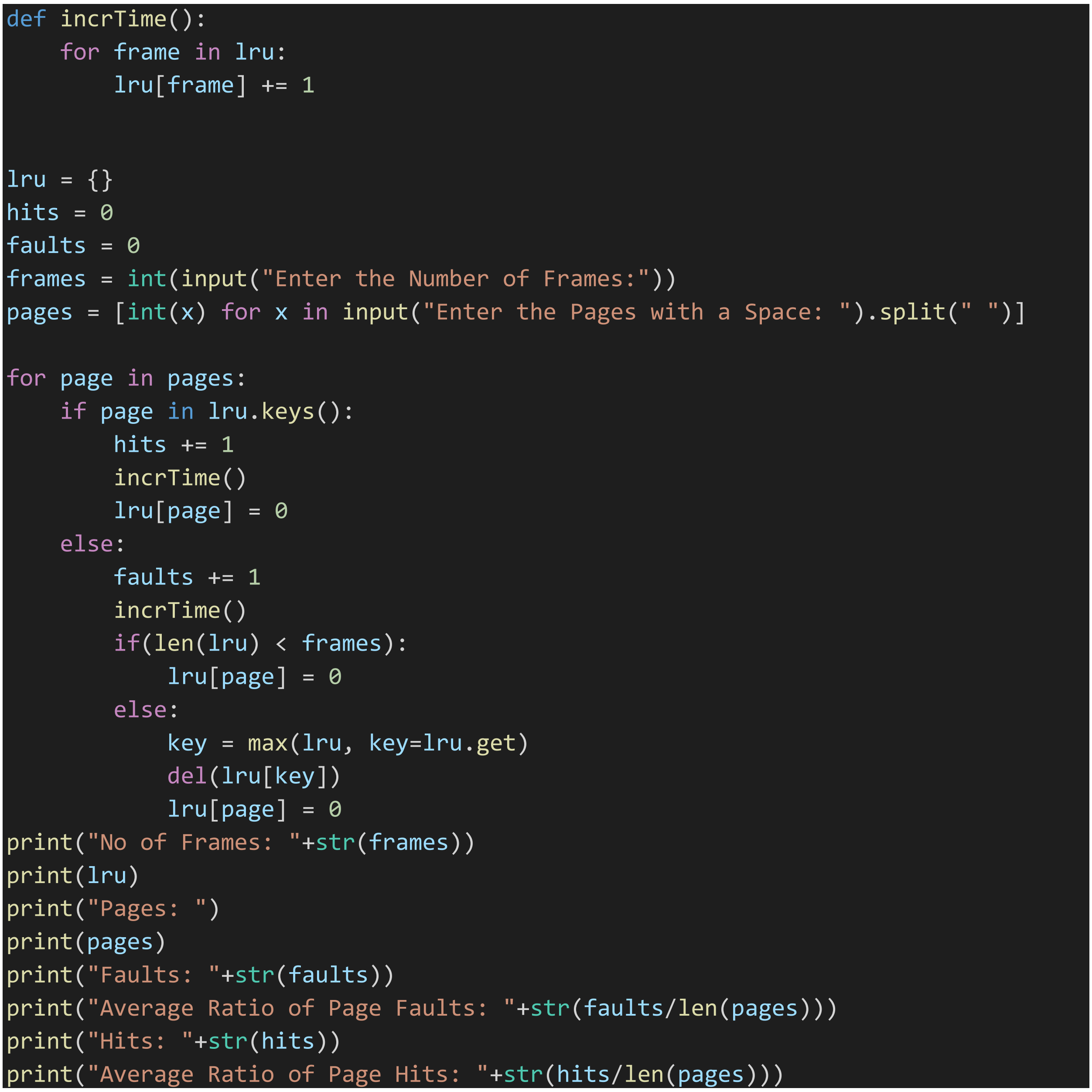
Results: Attach the results here.

The assignment submitted should be e- media saved as <Roll No_Batch No_Date>

This file must contain on the top:
Name:
Roll No.
Exp No.
Batch:
Date:

And Students have to upload this document electronically.

CODE(Python):

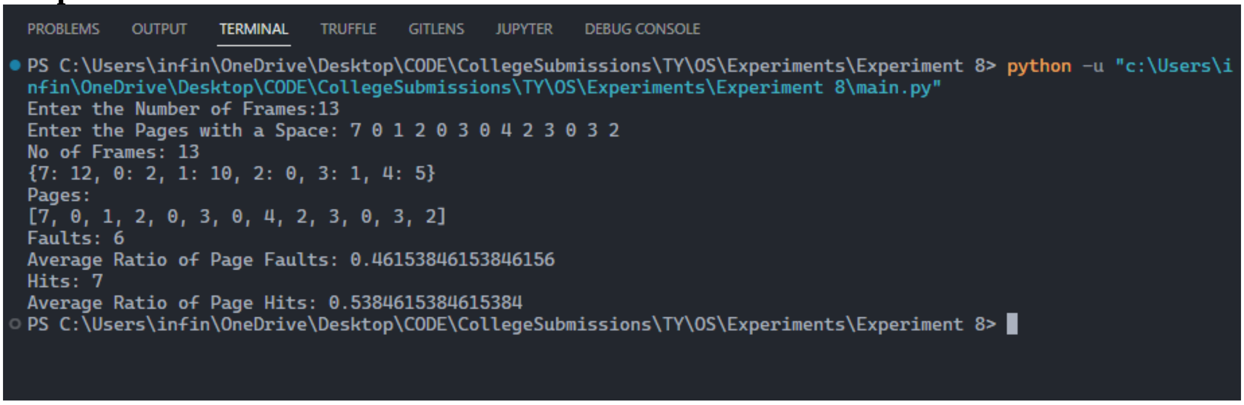


```
def incrTime():
    for frame in lru:
        lru[frame] += 1


lru = {}
hits = 0
faults = 0
frames = int(input("Enter the Number of Frames:"))
pages = [int(x) for x in input("Enter the Pages with a Space: ").split(" ")]

for page in pages:
    if page in lru.keys():
        hits += 1
        incrTime()
        lru[page] = 0
    else:
        faults += 1
        incrTime()
        if(len(lru) < frames):
            lru[page] = 0
        else:
            key = max(lru, key=lru.get)
            del(lru[key])
            lru[page] = 0
print("No of Frames: "+str(frames))
print(lru)
print("Pages: ")
print(pages)
print("Faults: "+str(faults))
print("Average Ratio of Page Faults: "+str(faults/len(pages)))
print("Hits: "+str(hits))
print("Average Ratio of Page Hits: "+str(hits/len(pages)))
```

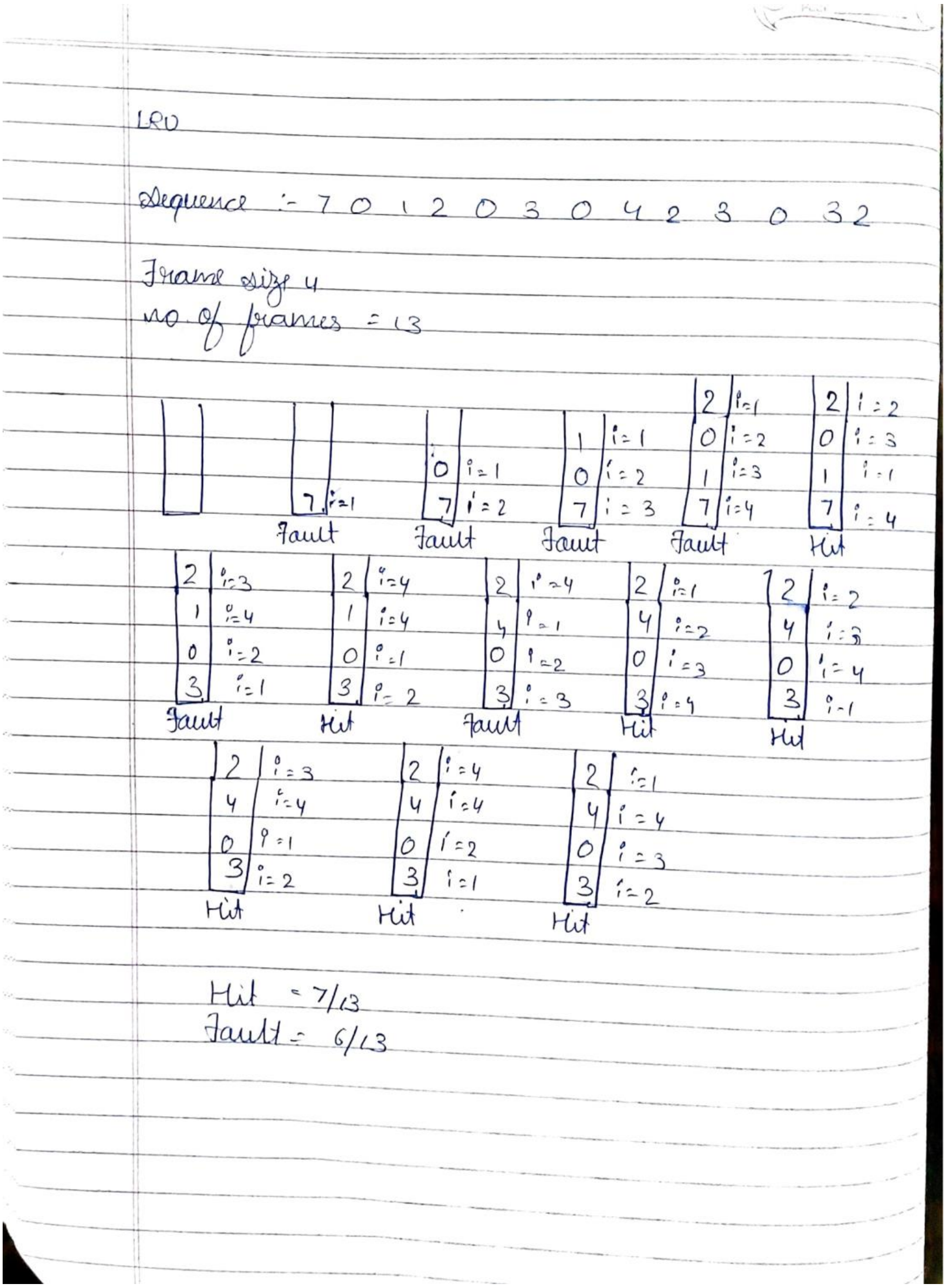
Output:



```
PS C:\Users\infin\OneDrive\Desktop\CODE\CollegeSubmissions\TY\OS\Experiments\Experiment 8> python -u "c:\Users\i
nfin\OneDrive\Desktop\CODE\CollegeSubmissions\TY\OS\Experiments\Experiment 8\main.py"
Enter the Number of Frames:13
Enter the Pages with a Space: 7 0 1 2 0 3 0 4 2 3 0 3 2
No of Frames: 13
{7: 12, 0: 2, 1: 10, 2: 0, 3: 1, 4: 5}
Pages:
[7, 0, 1, 2, 0, 3, 0, 4, 2, 3, 0, 3, 2]
Faults: 6
Average Ratio of Page Faults: 0.46153846153846156
Hits: 7
Average Ratio of Page Hits: 0.5384615384615384
PS C:\Users\infin\OneDrive\Desktop\CODE\CollegeSubmissions\TY\OS\Experiments\Experiment 8>
```

Questions: Solve the same problem manually.

Ans:



LRU

Sequence :- 7 0 1 2 0 3 0 4 2 3 0 3 2

Frame size 4
no. of frames = 13

(empty)

7 i=1
Fault

0 i=1
7 i=2
Fault

1 i=1
0 i=2
7 i=3
Fault

2 i=1
0 i=2
1 i=3
7 i=4
Fault

2 i=2
0 i=3
1 i=1
7 i=4
Hit

2 i=3
1 i=4
0 i=2
3 i=1
Fault

2 i=4
1 i=4
0 i=1
3 i=2
Hit

2 i=4
4 i=1
0 i=2
3 i=3
Fault

2 i=1
4 i=2
0 i=3
3 i=4
Hit

2 i=2
4 i=3
0 i=4
3 i=1
Hit

2 i=3
4 i=4
0 i=1
3 i=2
Hit

2 i=4
4 i=4
0 i=2
3 i=1
Hit

2 i=1
4 i=4
0 i=3
3 i=2
Hit

Hit = 7/13
Fault = 6/13

Outcomes: CO3: Understand I/O management, memory management and file management

Conclusion: (Conclusion to be based on outcomes achieved)

We learnt and implemented LRU (Least Recently Used) Page replacement Algorithm and calculated the average ratio of page faults and Page Hits.

Grade: AA / AB / BB / BC / CC / CD /DD

Signature of faculty in-charge with date

References:

Books:

1. Applied Operating System Concepts, 1st ed. Silberschatz, Galvin and Gagne, John Wiley Publishers.
2. Modern Operating Systems, Tanenbaum, PHI.
3. Operating System, 4th Edition, William Stallings, Pearson