Practical No 9

Aim : Develop, debug and execute a C program to simulate FIFO page replacement algorithms.

Apparatus: Mingw compiler for C/C++, and a text editor for developing C code file (Dev C++).

Theory :

What is Page Replacement algorithm?

- In an operating system that uses paging for memory management, a page replacement algorithm is needed to decide which page needs to be replaced when new page comes in.
- The page replacement algorithms do this task of deciding which page needs to be replaced when a new page arrives in the memory.

What are the components of Page Replacement Algorithms?

Page fault:

- A page fault happens when a running program accesses a memory page that is mapped into the virtual address space but is not loaded in physical memory.
- Since actual physical memory is much smaller than virtual memory, page faults can happen.
- In case of page faults, the operating system might have to replace one of the existing pages with the newly needed page.
- Different page replacement algorithms suggest different ways to decide which page to replace.
- The target for all algorithms is to reduce the number of page faults.

Page Hit:

- When we want to load the page on the memory, and the page is already available on memory, then it is called page hit.

What is FIFO Page Replacement?

- This is the Simplest page replacement algorithm.
- In this algorithm, the operating system keeps track of all pages in the memory in a queue, the oldest page is in the front of the queue.
- When a page needs to be replaced page in the front of the queue is selected for removal.

<u>Example</u>:

Reference string: 7,0,1,2,0,3,0,4,2,3,0,3,1,2,0

No of frames: 3

F3			1	1	1	1	0	0	0	3	3	3	3	2	2
F2		0	0	0	0	3	3	3	2	2	2	2	1	1	1
F1	7	7	7	2	2	2	2	4	4	4	0	0	0	0	0
	*	*	*	*	HIT	*	*	*	*	*	*	HIT	*	*	HIT

Page fault (*): 12

Page hit (HIT): 3

Page fault ratio = No. of page fault / No. of reference string

= 12/15

= 80%

Page hit ratio = No. of page ratio / No. of reference string

= 3/15

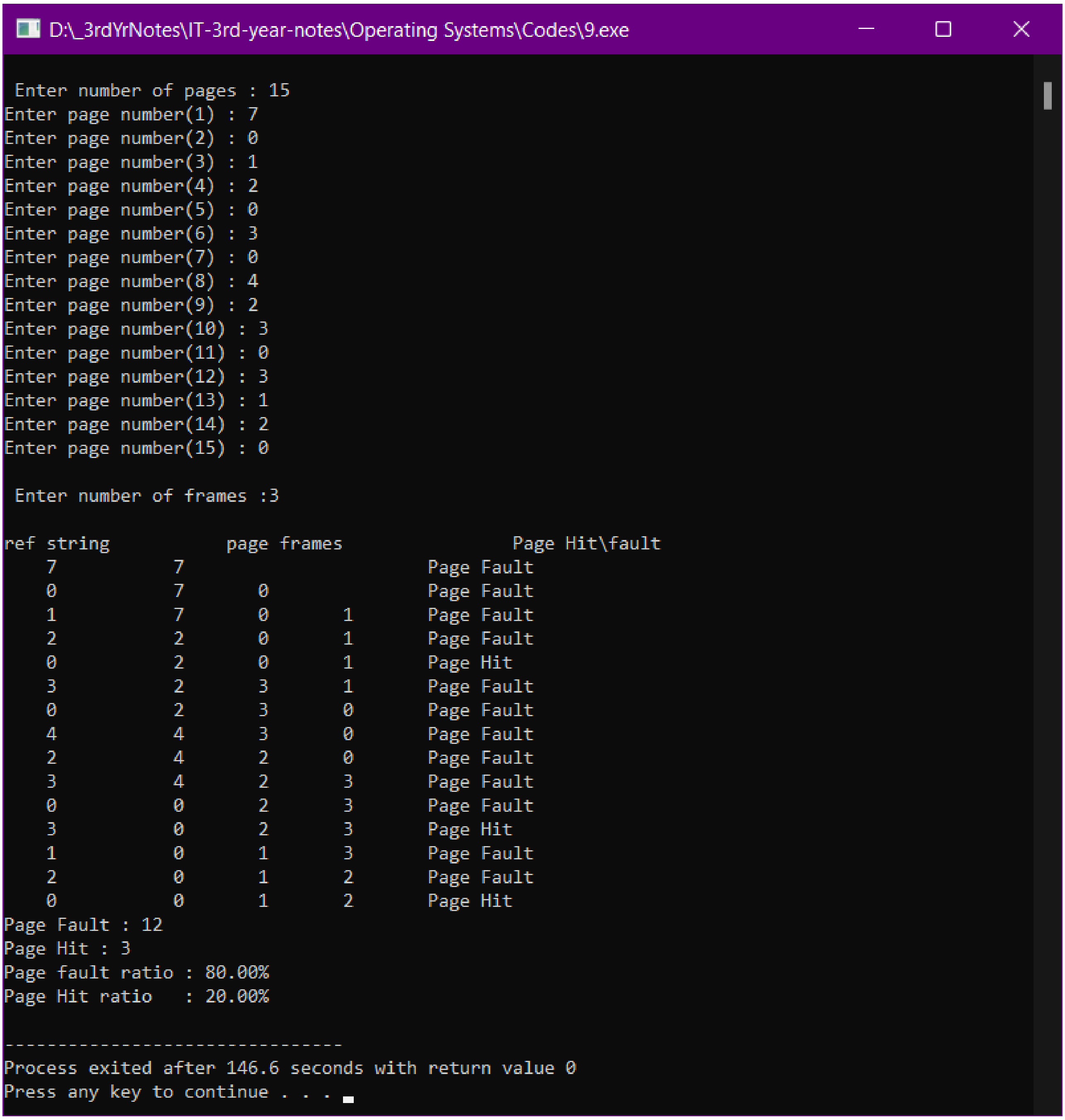
= 20%

Code:

```
#include<stdio.h>
int main()
{
int i,j,n,a[50],frame[10],no,k,avail,count=0;
    printf("\n Enter number of pages : ");
      scanf("%d",&n);
    for(i=1;i<=n;i++){
      printf("\n Enter page number(%d) : ",i+1);
    scanf("%d",&a[i]);
      }
    printf("\n Enter number of frames :");
      scanf("%d",&no);
      printf("\n");
for(i=0;i<no;i++)
    frame[i]= -1;
      j=0;
      printf("ref string\t     page frames\t\tPage Hit\\fault\n");
for(i=1;i<=n;i++){
      printf("%5d\t\t",a[i]);
      avail=0;
      for(k=0;k<no;k++)
if(frame[k]==a[i])
      avail=1;
    if (avail==0){
            frame[j]=a[i];
        j=(j+1)%no;
        count++;
      }
    for(k=0;k<no;k++){
      if(frame[k] == -1)
      printf(" \t");
      else
          printf("%d\t",frame[k]);
      }
      printf("%s",avail==1 ? "Page Hit" : "Page Fault");
      printf("\n");
}

    printf("Page Fault : %d\n",count);
    printf("Page Hit : %d\n",n-count);
    printf("Page fault ratio : %.02f%%\n",count/(float)n * 100);
    printf("Page Hit ratio   : %.02f%%\%\n",(n-count)/(float)n * 100);
    return 0;
}
```

Output:


D:_3rdYrNotes\IT-3rd-year-notes\Operating Systems\Codes\9.exe

```
 Enter number of pages : 15
Enter page number(1) : 7
Enter page number(2) : 0
Enter page number(3) : 1
Enter page number(4) : 2
Enter page number(5) : 0
Enter page number(6) : 3
Enter page number(7) : 0
Enter page number(8) : 4
Enter page number(9) : 2
Enter page number(10) : 3
Enter page number(11) : 0
Enter page number(12) : 3
Enter page number(13) : 1
Enter page number(14) : 2
Enter page number(15) : 0

 Enter number of frames :3

ref string           page frames                Page Hit\fault
    7           7                       Page Fault
    0           7       0               Page Fault
    1           7       0       1       Page Fault
    2           2       0       1       Page Fault
    0           2       0       1       Page Hit
    3           2       3       1       Page Fault
    0           2       3       0       Page Fault
    4           4       3       0       Page Fault
    2           4       2       0       Page Fault
    3           4       2       3       Page Fault
    0           0       2       3       Page Fault
    3           0       2       3       Page Hit
    1           0       1       3       Page Fault
    2           0       1       2       Page Fault
    0           0       1       2       Page Hit
Page Fault : 12
Page Hit : 3
Page fault ratio : 80.00%
Page Hit ratio   : 20.00%

--------------------------------
Process exited after 146.6 seconds with return value 0
Press any key to continue . . .
```

Conclusion:

Hence, by performing this practical I got to know about the concept of Page replacement, page fault and page hit. I also learnt about FIFO page replacement algorithm. I also developed, debugged and executed a C program to simulate FIFO page replacement algorithms.