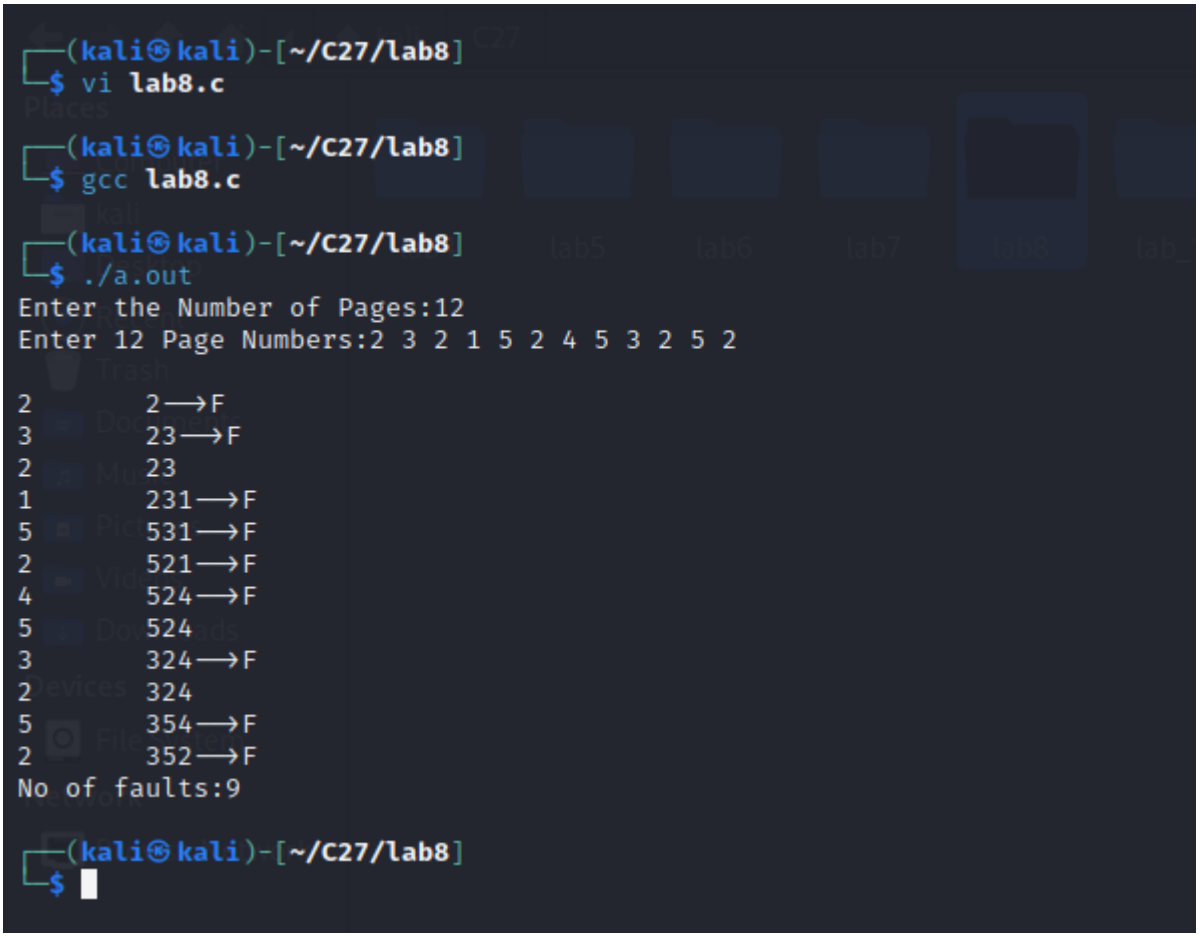
Student Name:	Aditya.Rajesh.Wanwade
Roll No:	C2-27
Practical No:	7
Aim:	Write C programs to simulate Page Replacement Algorithms: FIFO, LRU. Page Replacement Algorithms:

A)To Simulate FIFO page replacement algorithms.

```
#include <stdio.h>
int main()
{
    int a[5], b[20], n, p = 0, q = 0, m = 0, h, k, i, q1 = 1;
    char f = 'F';
    printf("Enter the Number of Pages:");
    scanf("%d", &n);
    printf("Enter %d Page Numbers:", n);
    for (i = 0; i < n; i++)
        scanf("%d", &b[i]);
    for (i = 0; i < n; i++)
    {
        if (p == 0)
        {
            if (q >= 3)
                q = 0;
            a[q] = b[i];
            q++;
            if (q1 < 3)
            {
                q1 = q;
            }
        }
        printf("\n%d", b[i]);
        printf("\t");
        for (h = 0; h < q1; h++)
            printf("%d", a[h]);
        if ((p == 0) && (q <= 3))
        {
```

```
            printf("-->%c", f);
            m++;
        }
        p = 0;
        for (k = 0; k < q1; k++)
        {
            if (b[i + 1] == a[k])
                p = 1;
        }
    }
    printf("\nNo of faults:%d", m);
}
```

<u>OUTPUT:</u> (All the test cases are included):



B)To Simulate LRU page replacement algorithms.

```
#include <stdio.h>
int main()
{
    int a[5], b[20], p = 0, q = 0, m = 0, h, k, i, q1 = 1, j, u, n;
    char f = 'F';
    printf("Enter the number of pages:");
    scanf("%d", &n);
    printf("Enter %d Page Numbers:", n);
    for (i = 0; i < n; i++)
        scanf("%d", &b[i]);
    for (i = 0; i < n; i++)
    {
        if (p == 0)
        {
            if (q >= 3)
                q = 0;
            a[q] = b[i];
            q++;
            if (q1 < 3)
            {
                q1 = q;
            }
        }
        printf("\n%d", b[i]);
        printf("\t");
        for (h = 0; h < q1; h++)
            printf("%d", a[h]);
        if ((p == 0) && (q <= 3))
        {
            printf("-->%c", f);
            m++;
        }
        p = 0;
```

```
        if (q1 == 3)
        {
            for (k = 0; k < q1; k++)
            {
                if (b[i + 1] == a[k])
                    p = 1;
            }
            for (j = 0; j < q1; j++)
            {
                u = 0;
                k = i;
                while (k >= (i - 1) && (k >= 0))
                {
                    if (b[k] == a[j])
                        u++;
                    k--;
                }
                if (u == 0)
                    q = j;
            }
        }
        else
        {
            for (k = 0; k < q; k++)
            {
                if (b[i + 1] == a[k])
                    p = 1;
            }
        }
    }
    printf("\nNo of faults:%d", m);
}
```

<u>OUTPUT:</u> (All the test cases are included):

```
┌──(kali㉿kali)-[~/C27/lab8]
└─$ vi lab8b.c

┌──(kali㉿kali)-[~/C27/lab8]
└─$ gcc lab8b.c

┌──(kali㉿kali)-[~/C27/lab8]
└─$ ./a.out
Enter the number of pages:12
Enter 12 Page Numbers:2 3 2 1 5 2 4 5 3 2 5 2

2       2⟶F
3       23⟶F
2       23
1       231⟶F
5       251⟶F
2       251
4       254⟶F
5       254
3       354⟶F
2       352⟶F
5       352
2       352
No of faults:7

┌──(kali㉿kali)-[~/C27/lab8]
└─$
```

<u>Result</u>: Thus the program was executed and verified successfully.