Ex. No.: 11c)
Date: 23/04/2025

Optimal

Aim:
To write a c program to implement Optimal page replacement algorithm.

ALGORITHM:

1. Start the process
2. Declare the size
3. Get the number of pages to be inserted
4. Get the value
5. Declare counter and stack
6. Select the least frequently used page by counter value
7. Stack them according the selection.
8. Display the values
9. Stop the process

PROGRAM:

```
# include <stdio.h>
int search (int key, int frames [], int f ) {
        for ( int i=0 ; i<f ; i++) {
            if (frame [i] ==key)
                return 1;
        }
        return 0;
}
int predict ( int pages[], int frame[], int n,
    index, int f) {
        int res = -1, farthest = index;
        for ( int i =0; i < f; i++) {
            int i;
```

```
        for (j = index; j < n; j++){
            if ( frame[i] == pages[j]){
                if (j > farthest){
                    farthest = j;
                }
                break;
            }
        }
        if(j == n)
            return i;
    }

    return (res == -1)? 0 : res;
}

int main (){
        int n, f;
        printf(" Enter no. of frames: ")
        scanf (" %d", &f);
        printf (" Enter no. of pages: ")
        scanf ("%d", &n);
        int pages[n];
        printf ("Enter references string :\n");
        for(int i =0; i<n ; i++)
                scanf ("%d", & pages[i]);
        int frame[f];
        int count=0; index =0;
        for (int i=0; i<f; i++)
                frame[i] = -1;
        printf("\n Page Replacement Process :\n");
```

```
for (int i=0; i<n; i++) {
        if (!search(pages[i], frame, f)){
                if(index < f) {
                        frame[index-1] = pages[i];
        } else {
                int pos = predict(pages, frame, n, i+1, f);
                frame[pos]=pages[i];
        }
        count++;
}
for (int j = 0; j<f; j++){
        if (frame[j] != -1)
                printf("%d", frame[j]);
        else
                printf("-1");
        }
        printf("\n");
}
printf("\n Total page faults %d\n", count);
return 0;
}
```

Output:

Enter no. of frames: 3
Enter no. of pages: 12
Enter reference string: 7 0 1 2 0 3 0 4 2 3 0 3

```
7 -1 -1
7 -0 -1
 7 0  1
 2 0  1
 2 0  1
 2 0  3
 2 0  3
 4 0  3
 4 0  2
 4 3  2
 0 3  2
 0 3  2
```

Result:

The optimal pages replacement algorithm has been successfully implemented.