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

Aim: **Optimal**

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>
#define MAX 100
int predict(int pages[], int frames[], int n, int index, int capacity){
    int result = -1, farthest = index;
    for(int i=0; i<capacity; i++){
        int j;
        for(int j=0; j<n; j++){
            if(frames[i]==pages[j]){
                if(j>farthest){
                    farthest=j;
                    result=i;
                }
                break;
            }
        }
        if(j==n)
            return i;
    }
    return (result==-1)?0:result;
```

```
int main(){
    int pages[MAX], frames[MAX];
    int n, capacity, faults=0, hit=0;
    int i, j, k, filled=0;
    printf("Enter the no. of Pages : ");
    scanf("%d", &n);
    printf("Enter the ~~no. of~~ reference string : ");
    ~~scanf~~ for(i=0; i<n; i++){
        scanf("%d", &pages[i]);
    }
    printf("Enter the no. of frames: ");
    scanf("%d", &capacity);
    for(i=0; i<capacity; i++){
        frames[i];
    }
for(i=0; i<n; i++){
    int found=0;
    for(j=0; j<capacity; j++){
        if(frames[j]==pages[i]){
            found=1;
            hit++;
            break;
    }   }   }
if(!found){
    if(filled<capacity){
        frames[filled++]=pages[i];
    }
    else{
        int pos = predict(pages, frames, n, i+1, capacity);
        frames[pos]=pages[i];
    }
    faults++;
}
printf("%d", pages[i]);
for(k=0; k<capacity; k++){
    if(frames[k]!=-1){
        printf("%d", frame[k]); }
    else{
        printf("-");
    }
    printf("\n");
}
printf("Total Page faults : %d \n", faults);
printf("Total Page hits : %d \n", hit);
return 0;
}
```

Output:

Enter the no. of pages : 12
Enter the reference string : 1 3 0 3 5 6 3 0 6 4 1 7
Enter the no. of frames : 3

1 : 1 _ _
3 : 1 3 _
0 : 1 3 0
3 : 1 3 0
5 : 5 3 0
6 : 5 6 0
3 : 5 6 3
0 : 5 0 3
6 : 6 0 3
4 : 6 0 4
1 : 1 0 4
7 : 1 7 4

Total Page faults = 9
Total Page hits = 3

Result:

Thus, optimal page replacement algorithm is implemented and executed succesfully.