Ex. No.: 11c)
Date: 23/4/25

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 main(){
    int refstr[100], frames[10];
    int n, f, i, j, k, pagefaults=0, hit;
    printf("Enter thesize of reference string: ");
    scanf("%d", &n);
    for (i=0; i<n; i++) {
        printf("Enter[%d]: ", i+1);
        Scanf("%d", &refStr[i]);
    }
    print("Enter page frame size:");
    Scanf("%d", &f);
    for (i=0; i<ff ; i++)
        frames[i] = -1;
    printf ("\n");
    for
```

```
for(i=0; i<n; i++){
    hit=0;
    for(j=0; j<f; j++){
        if(frames[j]==refStr[i]){
            hit=1;
            break;
        }
    }
    if(hit){
        printf("%2d -> No Page Fault\n", refStr[i]);
        continue;
    }
    int empty=-1;
    for(j=0; j<f; j++){
        if(frames[j]==-1){
            empty=j;
            break;
        }
    }
    if(empty!=-1){
        frames[empty]=refStr[i];
    }else{
        int farthest=-1, idx=-1;
        for(j=0; j<f; j++){
            int found=0;
            for(k=i+1; k<n; k++){
                if(frames[j]==refStr[k]){
                    found=1;
                    if(k>farthest){
                        farthest=k;
                        idx=j;
                    }
                }
            }
            frames[idx]=refStr[i];
        }
```

```
        pageFault++;
        printf("%2d->", refStr[i]);
        for(k=0; k<f; k++)
            if(frames[k]!=-1)
                printf("%d", frames[k]);
        }
        printf("=> Page Fault\n");
    }
    printf("\nTotal Page Faults: %d\n", pageFaults);
    printf("Total Page Hits: %d\n", n - pageFaults);
    return 0;
}
```

Output:

```
Enter the size of reference string: 10
Enter page frame size: 3

Enter[1]: 7        7 → 7 = page fault
Enter[2]: 0        0 → 70 ⇒ page fault
Enter[3]: 1        1 → 701 ⇒ page fault
Enter[4]: 2        2 → 201 ⇒ page fault
Enter[5]: 0        0 → No page fault
Enter[6]: 3        3 → 203 ⇒ page fault
Enter[7]: 0        0 → 203 ⇒ No page fault
Enter[8]: 4        4 → 243 ⇒ page fault
Enter[9]: 2        2 → No page fault
Enter[10]: 3       3 → No page fault

Total page faults: 6
Total page hits: 4
```

Result:

A C program for finding the page fault using optimal page replacement technique is implemented successfully.