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 the Size of reference String:");
    Scanf ("%d", &n);
   for ( i=0 ; i<n; i++){
     Printf ("Enter [%d] : ", i+1);
     Scanf ("%d", &refStr[i]);
  }
```

```
Printf ("Enter page fram size: ");
Scanf ("%d", & f);
for (i=0; i<f; i++)
    frames[i] = -1;

Print f (" \n");

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;
}
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;
            }
            break;
        }
    }
    if (! found){
        idx = j ;
        break ;
    }
  }
  frames [idx] = refStr[i];
}
 PageFaults ++;
 Printf ("%2d ->", refStr[i]);
 for (k= 0; k < f; k++){
     if (frames [k] != -1)
        Printf ("%d", frames[k]);
 }
 Printf("=> Page Fault\n");
}
Print f("\n Total Page Faults : %d\n", PageFaults);
Print f ("Total Page Hits : %d\n", n- Page Faults);
return 0;
```

Output:

```
Enter the size of reference String : 10

Enter [1] : 7
Enter [2] : 0
Enter [3] : 1
Enter [4] : 2
Enter [5] : 0
Enter [6] : 3
Enter [7] : 0
Enter [8] : 4
Enter [9] : 2
Enter [10] : 3

Enter Page Frame Size : 3

7 → 7 ⇒ Page Fault
0 → 7 0 ⇒ Page Fault
1 → 7 0 1 ⇒ Page Fault
2 → 2 0 1 ⇒ Page Fault
0 → No Page Fault
3 → 2 0 3 ⇒ Page Fault
0 → 2 0 3 ⇒ No Page Fault
4 → 2 4 3 ⇒ Page Fault
2 → No Page Fault
3 → No Page Fault

Total Page Faults : 6
Total Page hit : 4
```

[illegible]

Result:

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