Ex. No.: 11c)
Date:

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>

#include <conio.h>


int i, j, nof, nor, flag = 0, ref[50], frm[50], pf = 0, victim = -1;

int recent[10], optcal[50], count = 0;


int optvictim(int);


void main() {

   clrscr();

   printf("\n OPTIMAL PAGE REPLACEMENT ALGORITHM\n");

   printf("\n----------------------------\n");
```

```
printf("\nEnter the no. of frames: ");
scanf("%d", &nof);
printf("Enter the no. of reference string: ");
scanf("%d", &nor);
printf("Enter the reference string:\n");
for (i = 0; i < nor; i++) {
   scanf("%d", &ref[i]);
}
clrscr();
printf("\n OPTIMAL PAGE REPLACEMENT ALGORITHM");
printf("\n----------");
printf("\nThe given string:\n");
for (i = 0; i < nor; i++)
   printf("%4d", ref[i]);
for (i = 0; i < nof; i++) {
   frm[i] = -1;
   optcal[i] = 0;
}
for (i = 0; i < 10; i++)
   recent[i] = 0;
printf("\n");
for (i = 0; i < nor; i++) {
   flag = 0;
   printf("\nref no %d ->\t", ref[i]);
   for (j = 0; j < nof; j++) {
```

```
            if (frm[j] == ref[i]) {
                flag = 1;
                break;
            }
        }
        if (flag == 0) {
            count++;
            if (count <= nof)
                victim++;
            else
                victim = optvictim(i);
            pf++;
            frm[victim] = ref[i];
        }
        for (j = 0; j < nof; j++)
            printf("%4d", frm[j]);
    }
    printf("\n\nNumber of page faults: %d", pf);
    getch();
}

int optvictim(int index) {
    int i, j, temp, notfound;
    for (j = 0; j < nof; j++) {
        notfound = 1;
```

```
        for (i = index; i < nor; i++) {
            if (frm[j] == ref[i]) {
                notfound = 0;
                optcal[j] = i;
                break;
            }
        }
        if (notfound == 1)
            return j;
    }
    temp = optcal[0];
    for (i = 1; i < nof; i++) {
        if (temp < optcal[i])
            temp = optcal[i];
    }
    for (i = 0; i < nof; i++) {
        if (temp == optcal[i])
            return i;
    }
    return 0;
}
```

Output:

```c
#include <stdio.h>
#include <conio.h>

int i, j, nof, nor, flag = 0, ref[50], frm[50], pf = 0, victim = -1;
int recent[10], optcal[50], count = 0;

int optvictim(int);

void main() {

    printf("\n OPTIMAL PAGE REPLACEMENT ALGORITHM\n");
    printf("\n---------------------------\n");

    printf("\nEnter the no. of frames: ");
    scanf("%d", &nof);

    printf("Enter the no. of reference string: ");
    scanf("%d", &nor);

    printf("Enter the reference string:\n");
    for (i = 0; i < nor; i++) {
        scanf("%d", &ref[i]);
```

input

```
ref no 3 ->        3  -1  -1  -1  -1  -1  -1
ref no 4 ->        3   4  -1  -1  -1  -1  -1

Number of page faults: 2

...Program finished with exit code 255
Press ENTER to exit console.
```

Result:
Program is successfully executed and output is verified.