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:

```c
#include <stdio.h>
int main(){
    int frames[10], pages[30], nf, np, i, j, k, pageFaults=0, flag,
    index, farthest;

    printf("Enter number of frames: ");
    scanf("%d", &nf);

    printf("Enter number of pages:");
    scanf("%d", &np);
    printf("Enter reference string:");
    for(i=0; i<np; i++)
        scanf("%d", &pages[i]);
```

```
for (i=0; i<nf; i++)
    frames[i]=-1;
for( i=0; i <np; i++)
    flag=0;
for( j=0; j<nf; j++){
    if (frames[j] == pages[i]){
        flag=1;
        break;
    }
}

if (!flag){
    for(j=0; j<nf; j++){
        if(frames[j]==-1){
            frames[j]=pages[i];
            flag=1;
            break;
        }
    }
}

if (!flag){
    int used[10]={0};
    for(j=0; j<nf; j++)
    for(k=i+1; k<np; k++){
        if(frames[j]==pages[k]){
            used[j]=k;
            break;
        }
    }
    if(k==np) used[j]=999;
}
```

```
farthest = 0;
for(j = 1; j < nf ; j++) {
    if ( used [j] > used [farthest])
        farthest = j;
}
    frames [farthest] = pages[i];
    pageFaults++;
} else if (flag == 0) {
    page Faults++;
}
for(j = 0; j < nf; j++) {
    printf ("%d", frames [j]);
}
printf ("\n");
}
printf (" Total Page Faults = %d\n", pageFaults);
return 0;
}
```

replacement

Enter number of frames: 3

Enter number of pages : 6

Enter reference string: 4 1 2 4 5 1

4 -1 -1

4 1 -1

4 1 2

4 1 2

5 1 2

5 1 2

Total Page Faults = 4

Result:

Hence the Optimal algorithm for page replacement has been executed an successfully.