Ex. No.: 11c)
Date: 18.04.25

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 findoptimal (int pages[], int frames[], int
                                    currentIndex,
                 int totalpages, int totalframes ){
  int pos = -1, farthest = currentIndex;
  for (int i = 0; i < totalframes ; i++){
      int j;
      for (j= currentIndex +1; j< totalpages ; j++) {
          if (frames [i] == pages [j]){
              if (j< farthest) {
                  farthest = j;
                  pos = i;
              }
```

```
        break;
    }
}
if (j == totalpages){
    return 1; }
if (pos == -1) {
    pos = 0; }
}
return pos;
}
int main() {
    int framescount, pagescount;
    int pages[50], frames[10], faults = 0;
    printf("Enter number of frames: ");
    scanf("%d", &framescount);
    printf("Enter number of pages: ");
    scanf("%d", &pagecount);
    printf("Enter the page reference string: ");
    for(int i = 0; i< pagescount; i++){
        scanf("%d", &pages[i]);
    }
    for(int i=0; i< framescount; i++){
        frames[i] = -1;
    }
    for(int i= 0; i< pagescount; i++) {
        int found = 0;
        for(int j=0; j< framescount; j++){
            if(frames[j] == pages[i]){
                found = 1;
                break;
            }}
```

```
if (!found){
    int replaceIndex = -1;
    for(int j=0; j<framescount; j++){
        if(frames[j] == -1){
            replaceIndex = j;
            break;
        }
    }
    if (replaceIndex = -1){
        replaceIndex = findoptimal(pages, frames, i,
                                   pagecount, framescount); }
        frames[replaceIndex] = pages[i];
        faults++;
    }
    printf("Frame after page %d: ", pages[i]);
    for(int j=0; j<framescount; j++){
        if(frames[j] != -1){
            printf("%d", frames[j]); }
        else{
            printf("-");
        }
        printf("\n");
    }
    printf("Total page faults: %d\n", faults);
    return 0;
}
```

Enter the number of frames : 3
Enter number of pages : 12
Enter the page reference string : 7 0 12 0 3 0 4 2 3
03

Output:

```
Frame after page 7: 7 - -
Frame after page 0: 7 0 -
Frame after page 1: 7 0 1
Frame after page 2: 0 1 2
Frame after page 0: 0 1 2
Frame after page 3: 0 2 3
Frame after page 0: 0 2 3
Frame after page 4: 0 3 4
Frame after page 2: 0 2 4
Frame after page 3: 2 3 4
Frame after page 0: 3 0 4
Frame after page 3: 0 3 4
Total page faults: 9
```

Result:

Thus the code to implement optimal page replacement algorithm has been executed successfully.