Ex. No.: 11c)
Date: 12-04-2025

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 predict(int pages[], int frames[], int n, int
index, int frameSize) {          int res = -1, farthest = index;   for (int
i = 0; i < frameSize; i++) {              int j;              for (j = index; j
< n; j++) {              if (frames[i] == pages[j]) {              if (j >
farthest) {              farthest = j;                        res = i;
                    }
break;
        }
          }
          if (j == n)
 return i;
          }
          return (res == -1) ? 0 : res;
} int main()
{
```

```
        int frames[10], pages[30];      int i,
j, k, n, frameSize, faults = 0;             int
hit;
        printf("Enter number of frames: ");
scanf("%d", &frameSize);   printf("Enter
number of pages: ");   scanf("%d", &n);
        printf("Enter reference string: ");
        for (i = 0; i < n; i++)
scanf("%d", &pages[i]);         for (i = 0; i
< frameSize; i++)       frames[i]
= -1;   printf("\n");   for (i = 0; i < n; i++)
{       hit = 0;            for (j = 0; j <
frameSize; j++) {                   if
(frames[j] == pages[i]) {                   hit
= 1;            break;
      }
        }       if (!hit) {
        int empty = -1;           for (j
= 0; j < frameSize; j++) {
        if (frames[j] == -1) {
empty = j;                  break;
                }
      }         if (empty != -1) {              frames[empty]
= pages[i];
      } else {                  int pos = predict(pages, frames, n, i
+ 1, frameSize);                  frames[pos] = pages[i];
      }         faults++;         }
        for (k = 0; k < frameSize; k++)
{       if (frames[k] != -1)
printf("%d ", frames[k]);
else            printf("-1 ");
        }
        printf("\n");
        }

 printf("\nTotal Page Faults = %d\n", faults);    return 0;
}
```

OUTPUT:

```
Enter number of frames: 3
Enter number of pages: 10
Enter reference string: 3
2
6
8
3
4
1
2
2
6

3 -1 -1
3 2 -1
3 2 6
3 2 8
3 2 8
4 2 8
1 2 8
1 2 8
1 2 8
6 2 8

Total Page Faults = 7
```

RESULT:

Hence, page faults that occur using OPTIMAL page replacement technique has been found.