OS LAB MANUAL (CS23431)

Roll No:230701263

EX.NO:11(C)

Optimal

Aim: To write a c program to implement Optimal page replacement algorithm

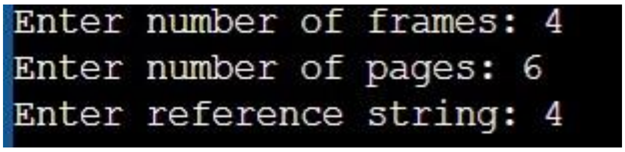
Program:

```
#include <stdio.h> int findOptimal(int pages[], int frames[], int
n, int index, int frameSize) { int farthest = index; int pos = -1; 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; pos = i;
        }
        break;
      }
    }
    if (j == n)
      return i;
  }
  if (pos == -1)
  return     0;
  else
    return pos;
}
int main() { int frames[10], pages[30], n, frameSize, i, j, k, pageFaults =
  0, found; 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;
for (i = 0; i < n; i++) {
   found = 0;
   for (j = 0; j < frameSize; j++) {
       if (frames[j] == pages[i]) {
       found = 1; break;
       }
   }
   if (!found) { int replaceIndex =
      -1;  for  (j  =  0;  j  <  frameSize;
      j++)  {  if  (frames[j]  ==  -1)  {
      replaceIndex = j; break;
          }
      }
      if   (replaceIndex   ==   -1)   {   replaceIndex   =   findOptimal(pages,
         frames, n, i + 1, frameSize);
      }
      frames[replaceIndex] = pages[i];
      pageFaults++;
   }
   for (k = 0; k < frameSize; k++) {
      if      (frames[k]       !=       -1)
      printf("%d     ",     frames[k]);
      else
          printf("-1 ");
   }
   printf("\n");
}
```

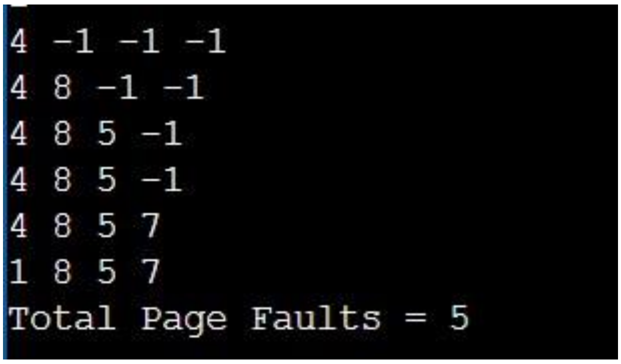
```
    printf("Total Page Faults = %d\n", pageFaults); return

    0;

}
```

Input:



```
Enter number of frames: 4
Enter number of pages: 6
Enter reference string: 4
```

Output:



```
4 -1 -1 -1
4 8 -1 -1
4 8 5 -1
4 8 5 -1
4 8 5 7
1 8 5 7
Total Page Faults = 5
```