**Lab no:  Date:2079/**

**Title: Write a program to simulate OPR page replacement algorithm.**

**Optimal Page Replacement:**

**In this algorithm, the pages are replaced with the ones that will not be used for the longest duration of time in the future. Optimal page replacement is the best page replacement algorithm as this algorithm results in the least number of page faults.**

**Algorithm:**

Step 1: Start

Step 2: Input the page sequence

Step 3: Insert the pages directly if frame is empty

Step 4: Insert the pages by replacing the least appearing one in the later sequence

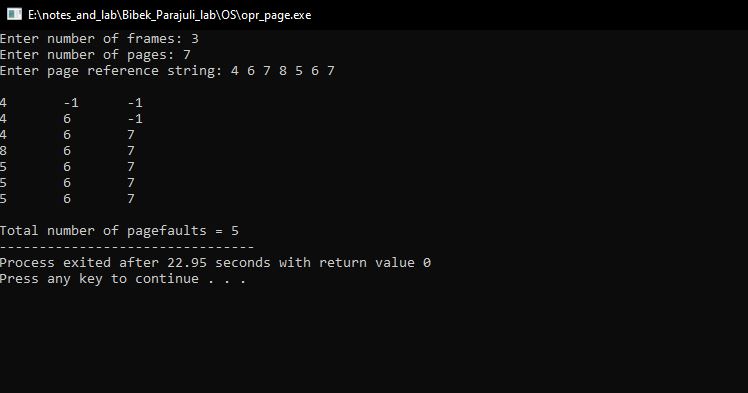
Step 5: Repeat step 2,3,4 until the input sequence is empty

Step 6: Stop

**Source code:**

|  |
| --- |
| #include<stdio.h>  int main()  {  int nof, nop, frames[5], pages[20], temp[20], f1, f2, f3, i, j, k, pos, max, pgfaults = 0;  printf("Enter number of frames: ");  scanf("%d", &nof);  printf("Enter number of pages: ");  scanf("%d", &nop);  printf("Enter page reference string: ");  for(i = 0; i < nop; i++)  {  scanf("%d", &pages[i]);  }  for(i = 0; i < nof; i++)  {  frames[i] = -1;  }  for(i = 0; i < nop; i++)  {  f1 = f2 = 0; //f1,f2 and f3 are the flag  for(j = 0; j < nof; j++)  {  if(frames[j] == pages[i])  {  f1 = f2 = 1;  break;  }  }  if(f1 == 0)  {  for(j = 0; j < nof; j++)  {  if(frames[j] == -1)  {  pgfaults++;  frames[j] = pages[i];  f2 = 1;  break;  }  }  }  if(f2 == 0)  {  f3 =0;  for(j = 0; j < nof; j++)  {  temp[j] = -1;  for(k = i + 1; k < nop; k++)  {  if(frames[j] == pages[k])  {  temp[j] = k;  break;  }  }  }  for(j = 0; j < nof; j++)  {  if(temp[j] == -1)  {  pos = j;  f3 = 1;  break;  }  }  if(f3 ==0)  {  max = temp[0];  pos = 0;  for(j = 1; j < nof; j++)  {  if(temp[j] > max)  {  max = temp[j];  pos = j;  }  }  }  frames[pos] = pages[i];  pgfaults++;  }  printf("\n");  for(j = 0; j < nof; j++)  {  printf("%d\t", frames[j]);  }  }  printf("\n\nTotal number of pagefaults = %d", pgfaults);  return 0;  } |

**Output:**



OPR output