OPERATING SYSTEM - CS23431

EXP 11(A)

FIFO PAGE REPLACEMENT

NAME: S.Manicka Meenakshi **ROLL NO: 230701173**

PROGRAM:

```
#include <stdio.h>

int main() {
    int n,frame_size,front=0,count=0,page_faults=0;
    printf("Enter size of reference string: ");
    scanf("%d",&n);
    int pages[n];
    for(int i=0;i<n;i++)
    {
        printf("Enter [%d]: ",i+1);
        scanf("%d",&pages[i]);
    }
    printf("Enter page frame size: ");
    scanf("%d",&frame_size);
    int mem[frame_size];
    for (int i = 0; i < n; i++) {
        int found = 0;
        for (int j = 0; j < count; j++) {
            if (mem[j] == pages[i]) {
                found = 1;
                break;
            }
        }

        printf("%d -> ", pages[i]);
        int f=1;
        if (!found) {
            if (count < frame_size) {
                mem[count++] = pages[i];
            } else {
                mem[front] = pages[i];
                front = (front + 1) % frame_size;
```

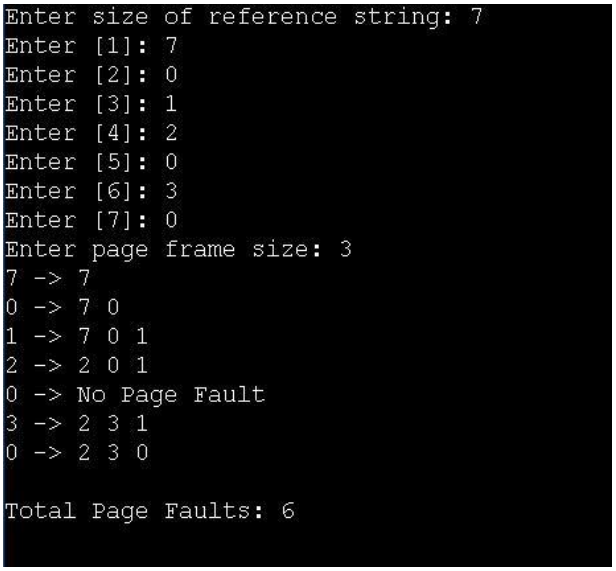
```
            }
            page_faults++;
        }
        else
        {
            f=0;
            printf("No Page Fault ");
        }
        if(f)
        {
        for (int j = 0; j < count; j++) {
            printf("%d ", mem[j]);
        }
        }

        printf("\n");
    }

    printf("\nTotal Page Faults: %d\n", page_faults);

    return 0;
}
```

OUTPUT:



OPERATING SYSTEM - CS23431

EXP 11(B)

LRU PAGE REPLACEMENT

NAME: S.Manicka Meenakshi **ROLL NO: 230701173**

PROGRAM:

```
#include <stdio.h>

int main() {
   int n,frame_size,count=0,page_faults=0;
   printf("Enter size of reference string: ");
   scanf("%d",&n);
   int page[n];
   for(int i=0;i<n;i++)
   {
      printf("Enter [%d]: ",i+1);
      scanf("%d",&page[i]);
   }
   printf("Enter page frame size: ");
   scanf("%d",&frame_size);
   int mem[frame_size];
   for (int i = 0; i < n; i++) {
      int top = -1;
      int f=0;
      for (int j = 0; j < count; j++) {
         if (mem[j] == page[i]) {
            top = j;
            break;
         }
      }

      printf("%d -> ", page[i]);
      if (top!=-1) {
            for(int j=0;j<count-1;j++)
            {
               mem[j]=mem[j+1];
            }
            mem[count-1]=page[i];
```

```
            printf("No page fault\n");
        }
        else
        {
            f=1;
            if(count<frame_size)
            {
                mem[count++]=page[i];
            }
            else
            {
                for(int j=0;j<frame_size;j++)
                {
                    mem[j]=mem[j+1];
                }
                mem[frame_size-1]=page[i];
            }
            page_faults++;
        }
        if(f)
        for (int j = 0; j < count; j++) {
            printf("%d ", mem[j]);
        }
        printf("\n");
    }

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

OUTPUT:

```
Enter size of reference string: 6
Enter [1]: 5
Enter [2]: 7
Enter [3]: 5
Enter [4]: 6
Enter [5]: 7
Enter [6]: 3
Enter page frame size: 3
5 -> 5
7 -> 5 7
5 -> No page fault

6 -> 7 5 6
7 -> No page fault

3 -> 6 7 3

Total Page Faults: 4
```

OPERATING SYSTEM - CS23431

EXP 11(C)

OPTIMAL PAGE REPLACEMENT

NAME: S.Manicka Meenakshi **ROLL NO: 230701173**

PROGRAM:

```
#include <stdio.h>
int findreplacementindex(int n,int frame_size,int page[],int mem[],int current)
{
   int ind[frame_size];
   for(int i=0;i<frame_size;i++)
   {
      ind[i]=-1;
      for(int j=current+1;j<n;j++)
      {
         if(mem[i]==page[j])
         {
            ind[i]=j;
            break;
         }
      }
   }
   int dist=-1,reqind=-1;;
   for(int i=0;i<frame_size;i++)
   {
      if(ind[i]==-1)
      {
         return i;
      }
      else if(ind[i]>dist)
      {
            dist=ind[i];
            reqind=i;
      }
   }
   return reqind;
}
int main() {
```

```
int n,frame_size,front=0,count=0,page_faults=0;
printf("Enter size of reference string: ");
scanf("%d",&n);
int page[n];
for(int i=0;i<n;i++)
{
   printf("Enter [%d]: ",i+1);
   scanf("%d",&page[i]);
}
printf("Enter page frame size: ");
scanf("%d",&frame_size);
int mem[frame_size];
for (int i = 0; i < n; i++) {
   int found = 0;
   for (int j = 0; j < count; j++) {
      if (mem[j] == page[i]) {
         found = 1;
         break;
      }
   }

   printf("%d -> ", page[i]);
   int f=1;
   if (!found) {
      if (count < frame_size) {
         mem[count++] = page[i];
      } else {
         int index=findreplacementindex(n,frame_size,page,mem,i);
         mem[index]=page[i];
      }
      page_faults++;
   }
   else
   {
      f=0;
      printf("No Page Fault ");
   }
   if(f)
   {
   for (int j = 0; j < count; j++) {
```

```
            printf("%d ", mem[j]);
        }
        }

        printf("\n");
    }

    printf("\nTotal Page Faults: %d\n", page_faults);

    return 0;
}
```

OUTPUT: