PAGE NO :
DATE :

Q) write a C program to stimulate page replacement algorithms
a) FIFO
b) LRU
c) Optimal

```
#include<stdio.h>
#include<stdbool.h>
void fifoPageReplacement(int pages, int incomingStream[],
    int frames){
  int pageFaults=0;
  int temp[frames];
  int front=0;
  for(int m=0; m<frames; m++){
      temp[m]=-1;
  }
  printf("\n FIFO Page Replacement Algorithm\n");
  printf("\n Incoming \t Frame 1 \t Frame 2 \t Frame
      3 \t Frame \n");
  for(int m=0; m<pages; m++){
    bool pageFault= true;
    for(int n=0; n<Frames; n++){
      if( temp[n] == incomingStream[m]){
          pageFault= false;
          break;
      }
    }
    if(pageFault){
        temp[front]= incomingStream[m];
        front= (front+1) % frames;
        pageFaults++;
    }
```

```
printf("%d \t \t", incomingStream[m]);
for(int n=0; n<frames; n++){
    if(temp[n] != -1)
        printf("%d \t \t", temp[n]);
    else
        printf("- \t \t");
}
printf("\n");
}
printf("\n Total Page Faults: \t %d \n",
                pageFaults);
}
void lruPageReplacement(int pages, int incomingSt
ream[], int frames){
int pageFaults = 0;
int temp[frames];
int used[frames];
for(int m=0; m<frames; m++){
    temp[m] = -1;
    used[m] = 0;
}
printf("\n LRU PageReplacement Algorithm \n");
printf("\n Incoming \t Frame 1 \t Frame 2 \t
  Frame 3 \t Frame ");
for(int m=0; m<pages; m++){
  bool pagefault = true;
  int leastUsedIdx = 0;
  for(int n=0; n<frames; n++){
    if(temp[n] == incomingStream[m]){
        pagefault = false;
        used[n] = m+1;
        break;
    }
```

```
        if(used[n] < used[leastUsedIdx]){
            leastUsedIdx = n;
        }
    }
    if(pageFault){
        temp[leastUsedIdx] = incomingStream[m];
        used[leastUsedIdx] = m+1;
        pageFaults++;
    }
    printf("%d \t \t", incomingStream[m]);
    for(int n=0; n < frames; n++){
        if(temp[n] != -1)
            printf("%d \t \t", temp[n]);
        else
            printf("- \t \t");
    }
    printf("\n");
}
printf("\n Total page faults: \t %d\n",
        pageFaults);
}

void optimalPageReplacement(int pages, int
    incomingStream[], int frames){
    int pageFaults = 0;
    int temp[frames];
    for(int m=0; m < frames; m++){
        temp[m] = -1;
    }
    printf("\n Optimal page Replacement \n");
    printf("\n Incoming \t Frame 1 \t Frame 2 \t
        Frame 3 \t Frame");
```

PAGE NO :
DATE :

```
for(int m=0; m<pages; m++){
    bool pagefault=true;
    int replaceIdx=-1;
    int farthest=m;
    for(int n=0; n<frames; n++){
        if(temp[n]==incomingStream[m]){
            pagefault=false;
            break;
        }
    }
    if(pagefault){
        for(int n=0; n<frames; n++){
            int j;
            for(j=m+1; j<pages; j++){
                if(temp[n]==incomingStream[j]){
                    if(j>farthest){
                        farthest=j;
                        replaceIdx=n;
                    }
                    break;
                }
            }
            if(j==pages){
                replaceIdx=n;
                break;
            }
        }
        if(replaceIdx==-1){
            replaceIdx=0;
        }
        temp[replaceIdx]=incomingStream[m];
        pagefaults++;
    }
```

```
printf("%d\t\t", incomingStream[m]);
for(int n=0; n<frames; n++){
    if(temp[n] != -1){
        printf("%d\t\t", temp[n]);
    else
        printf(" :\t\t");
    }
    printf("\n");
}
printf("\nTotal page faults:\t%d\n",
        pagefaults);
}
int main(){
    int pages;
    printf("Enter number of pages in incoming
            stream: ");
    scanf("%d", &pages);
    int incomingStream[pages];
    printf("Enter the sequence of pages:\n");
    for(int i=0; i<pages; i++){
        scanf("%d", &incomingStream[i]);
    }
    int frames;
    printf("Enter no.of frames: ");
    scanf("%d", &frames);
    fifoPageReplacement(pages, incomingStream, fra
    lruPageReplacement(pages, incomingStream, fra
    OptimalPageReplacement(pages, incomingStream, fra
    return 0;
}
```

PAGE NO.:
DATE:

Output:

Enter no. of pages in memory: [illegible]
Enter the sequence of pages:
0 9 0 1 8 1 8 7 8 7 1 2 8 2 7 8 2 3 8 3
Enter number of frames: 3

LRU Page Replacement Algorithm

Incoming	Frame1	Frame2	Frame3
0	0	-	-
9	0	9	-
0	0	9	-
1	0	9	1
8	8	9	1
1	8	9	1
8	8	9	1
7	8	7	1
8	8	7	1
7	8	7	1
1	8	7	1
2	8	7	2
8	8	7	2
2	8	7	2
7	8	7	2
8	8	7	2
2	8	7	2
3	3	7	2
8	3	8	2
3	3	8	2

Total page faults: 8

PAGE NO:
DATE:

LRU page Replacement algorithm

Incoming	Frame1	Frame2	Frame3
0	0	-	-
9	0	9	-
0	0	9	-
1	0	9	1
8	0	8	1
1	0	8	1
8	0	8	1
7	7	8	1
8	7	8	1
7	7	8	1
1	7	8	1
2	7	2	1
8	8	2	1
2	8	2	1
7	8	2	7
8	8	2	7
2	8	2	7
3	8	2	7
8	8	2	3
3	8	2	3
	8	2	3

Page Faults

page faults: 9

PAGE NO :
DATE :

Optimal page Replacement algorithm

Incoming	Frame1	Frame2	Frame3
0	0	—	—
9	0	9	—
0	0	9	—
1	1	9	—
8	1	8	—
1	1	8	—
8	1	8	—
7	1	8	7
8	1	8	7
7	1	8	7
1	1	8	7
2	2	8	7
8	2	8	7
2	2	8	7
7	2	8	7
8	2	8	7
2	2	8	7
3	3	8	7
8	3	8	7
3	3	8	7

Total page faults: 7

10/7/24