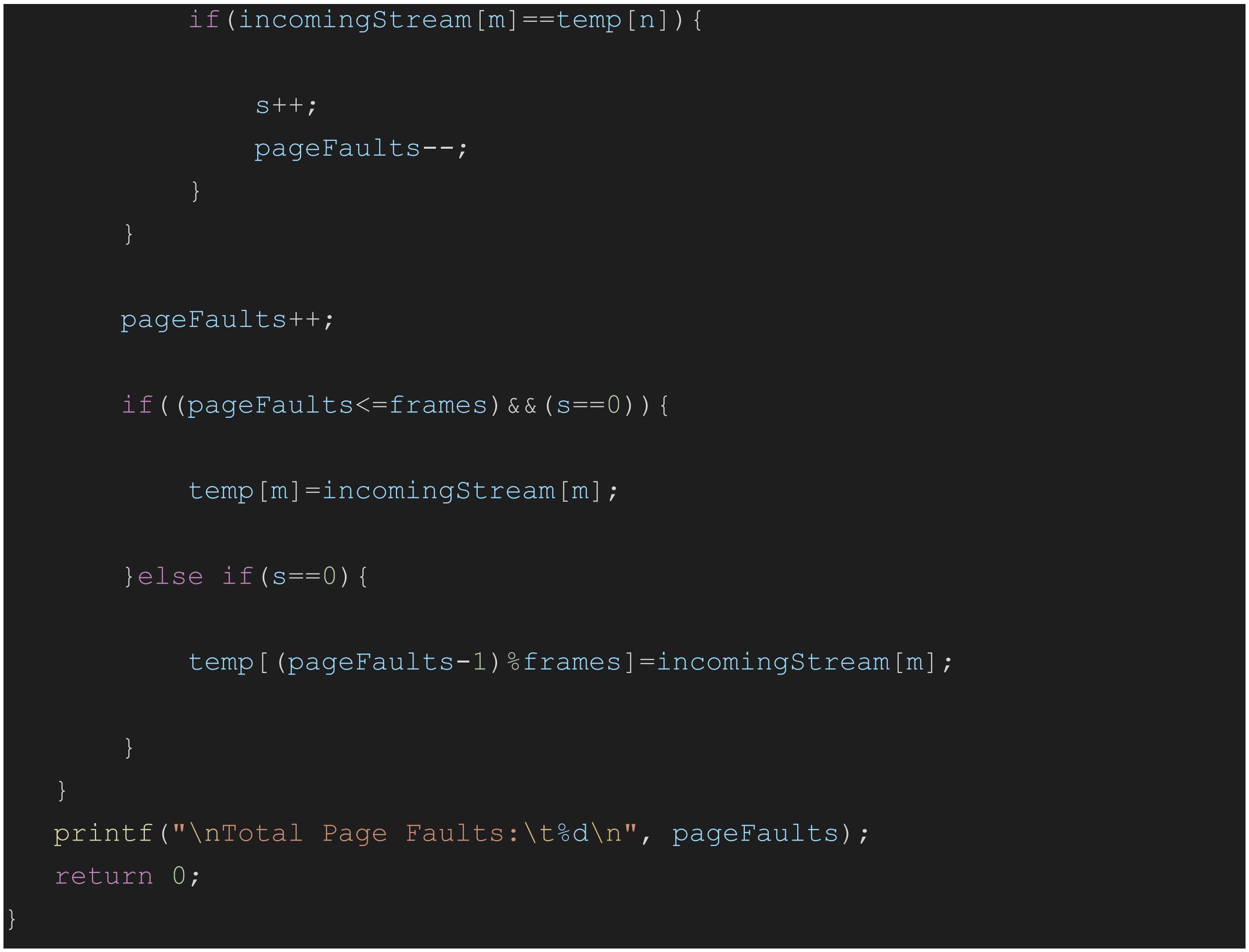
CS353
Lab6
200001053
Nilay Ganvit

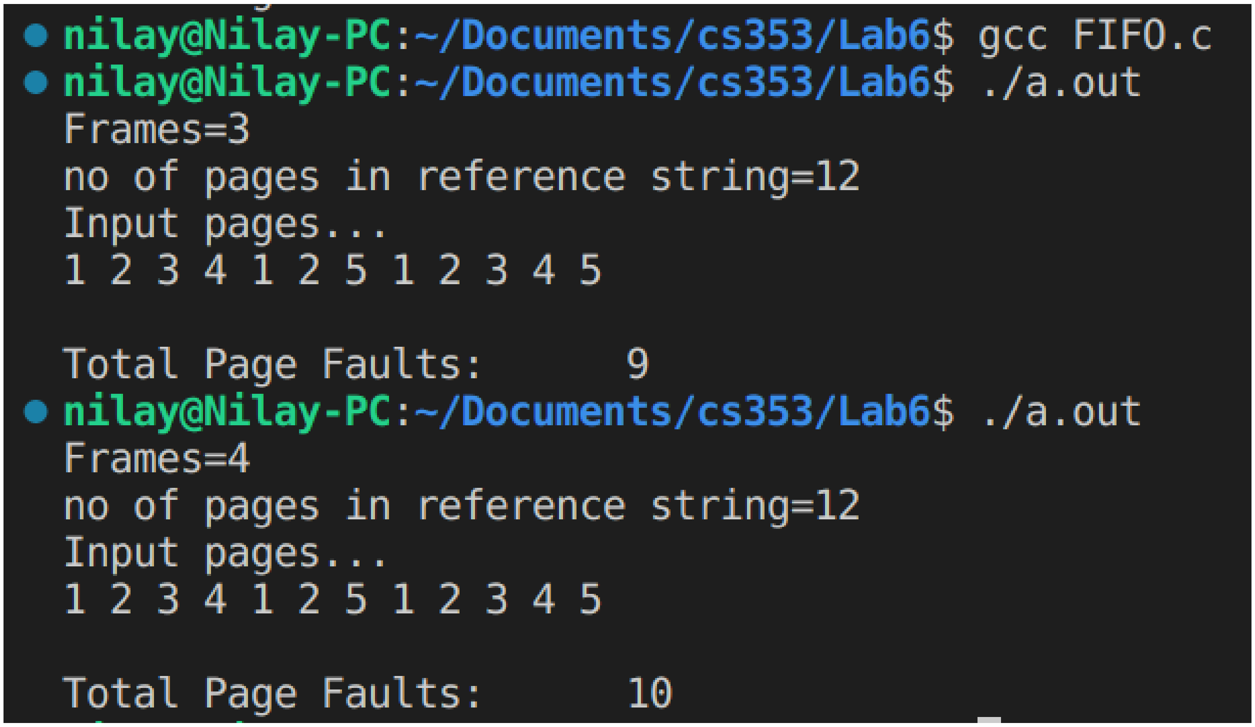
1. Write a program that simulates the following algorithms.
  i) FIFO(first in first out)

```
#include<stdio.h>

int main(){

   printf("Frames=");
   int frames;
   scanf("%d",&frames);
   printf("no of pages in reference string=");
   int pages;
   scanf("%d",&pages);

   int incomingStream[pages];
   printf("Input pages...\n");
   for(int i=0;i<pages;i++){
       scanf("%d",&incomingStream[i]);
   }
   int pageFaults=0;
   int m,n,s;

   int temp[frames];
   for(m=0;m<frames;m++){

       temp[m]=-1;

   }

   for(m=0;m<pages;m++){

       s=0;

       for(n=0;n<frames;n++){
```



```
            if(incomingStream[m]==temp[n]){

                s++;
                pageFaults--;
            }
        }

        pageFaults++;

        if((pageFaults<=frames)&&(s==0)){

            temp[m]=incomingStream[m];

        }else if(s==0){

            temp[(pageFaults-1)%frames]=incomingStream[m];

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

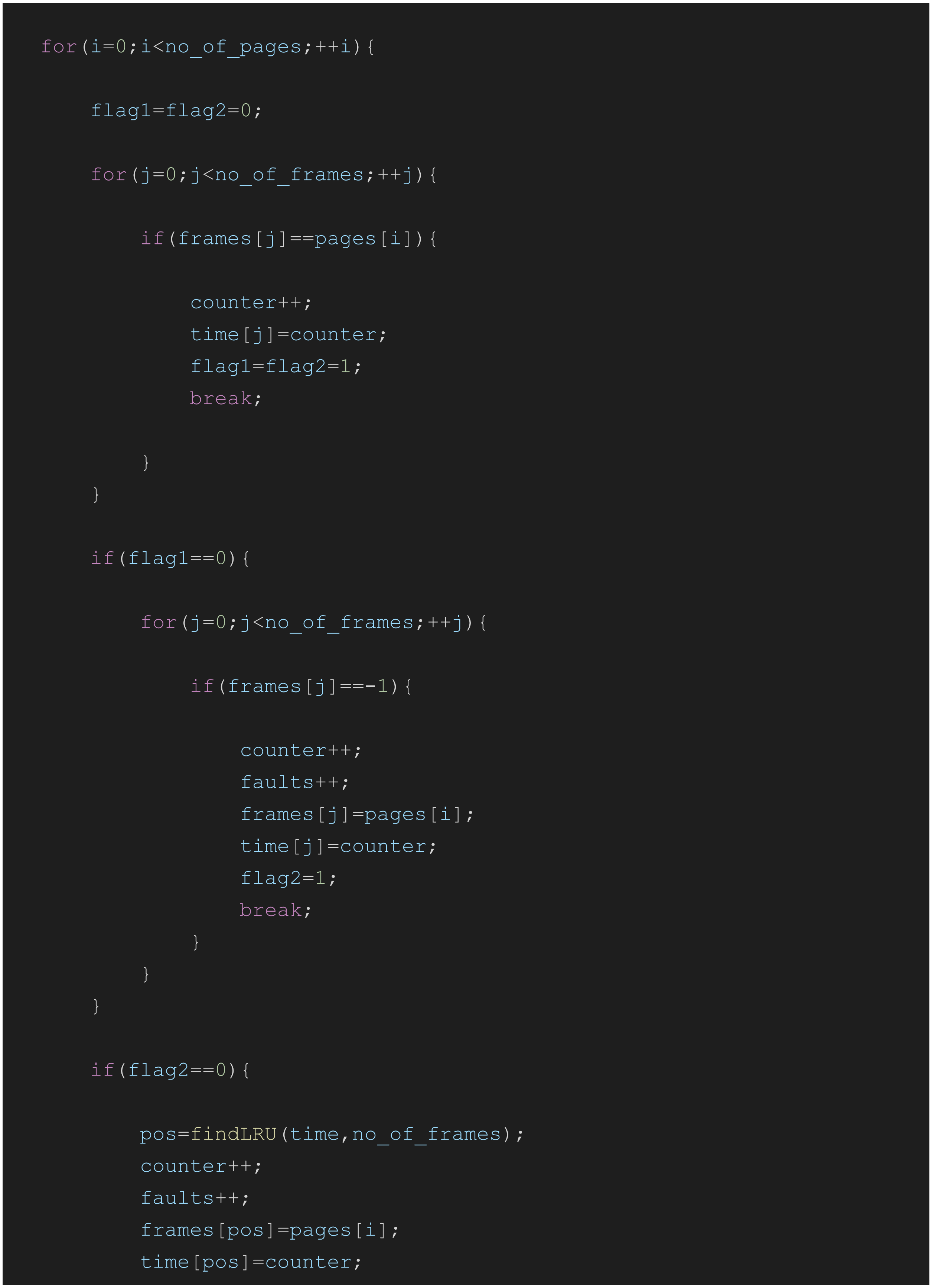
Input/Output:



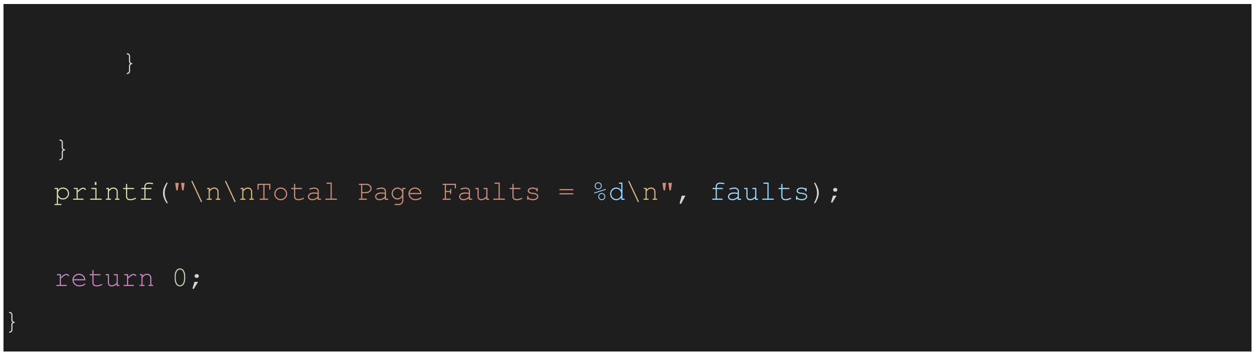
```
● nilay@Nilay-PC:~/Documents/cs353/Lab6$ gcc FIFO.c
● nilay@Nilay-PC:~/Documents/cs353/Lab6$ ./a.out
  Frames=3
  no of pages in reference string=12
  Input pages...
  1 2 3 4 1 2 5 1 2 3 4 5

  Total Page Faults:      9
● nilay@Nilay-PC:~/Documents/cs353/Lab6$ ./a.out
  Frames=4
  no of pages in reference string=12
  Input pages...
  1 2 3 4 1 2 5 1 2 3 4 5

  Total Page Faults:      10
```

ii) LRU(least recently used)

```
#include <stdio.h>

int findLRU(int time[],int n){

    int i,minimum=time[0],pos=0;

    for(i=1;i<n;++i){

        if(time[i]<minimum){

            minimum=time[i];
            pos=i;

        }
    }
    return pos;
}

int main(){

    int no_of_frames,no_of_pages,frames[10],pages[30],counter=0,time[10],flag1,flag2,i,j,pos,faults=0;
    printf("frames=");
    scanf("%d",&no_of_frames);
    printf("Pages=");
    scanf("%d",&no_of_pages);
    printf("reference string...\n");
    for(i=0;i<no_of_pages;++i){

        scanf("%d",&pages[i]);

    }

    for(i=0;i<no_of_frames;++i){

        frames[i]=-1;

    }
```

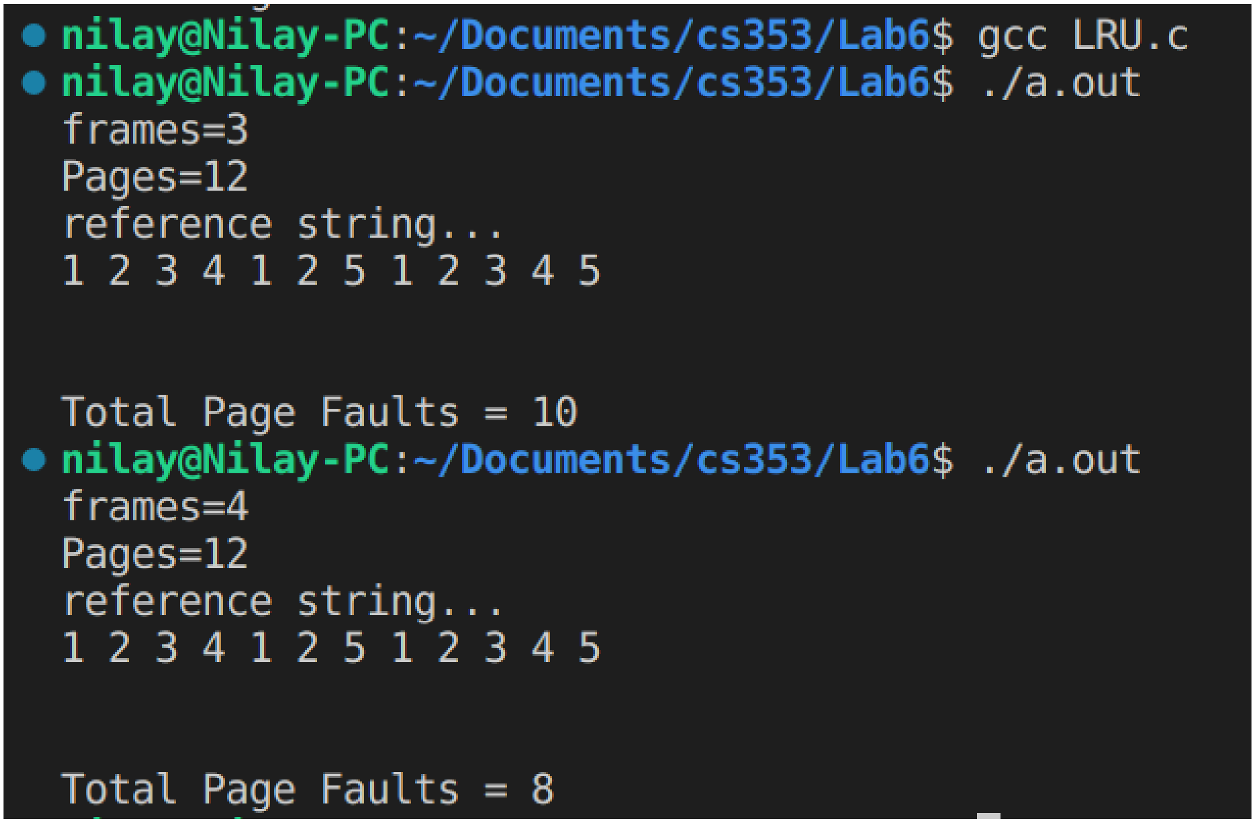


```
for(i=0;i<no_of_pages;++i){

    flag1=flag2=0;

    for(j=0;j<no_of_frames;++j){

        if(frames[j]==pages[i]){

            counter++;
            time[j]=counter;
            flag1=flag2=1;
            break;

        }
    }

    if(flag1==0){

        for(j=0;j<no_of_frames;++j){

            if(frames[j]==-1){

                counter++;
                faults++;
                frames[j]=pages[i];
                time[j]=counter;
                flag2=1;
                break;
            }
        }
    }

    if(flag2==0){

        pos=findLRU(time,no_of_frames);
        counter++;
        faults++;
        frames[pos]=pages[i];
        time[pos]=counter;
```



```
        }

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

    return 0;
}
```

Input/Output:



```
nilay@Nilay-PC:~/Documents/cs353/Lab6$ gcc LRU.c
nilay@Nilay-PC:~/Documents/cs353/Lab6$ ./a.out
frames=3
Pages=12
reference string...
1 2 3 4 1 2 5 1 2 3 4 5


Total Page Faults = 10
nilay@Nilay-PC:~/Documents/cs353/Lab6$ ./a.out
frames=4
Pages=12
reference string...
1 2 3 4 1 2 5 1 2 3 4 5


Total Page Faults = 8
```