|  |  |
| --- | --- |
| **Lab 10:** | **Program to implement Disk Scheduling Algorithm: FCFS** |

#include<stdio.h>

#include<stdlib.h>

int main()

{

    int RQ[100],i,n,TotalHeadMoment=0,initial;

    printf("Enter the number of Requests\n");

    scanf("%d",&n);

    printf("Enter the Requests sequence\n");

    for(i=0;i<n;i++)

     scanf("%d",&RQ[i]);

    printf("Enter initial head position\n");

    scanf("%d",&initial);

    // logic for FCFS disk scheduling

    for(i=0;i<n;i++)

    {

        TotalHeadMoment=TotalHeadMoment+abs(RQ[i]-initial);

        initial=RQ[i];

    }

    printf("Total head moment is %d",TotalHeadMoment);

    return 0;

}



|  |  |
| --- | --- |
| **Lab 11:** | **Program to implement Disk Scheduling Algorithm: FCFS** |

#include<stdio.h>

#include<stdlib.h>

int main()

{

    int RQ[100],i,n,TotalHeadMoment=0,initial,count=0;

    printf("Enter the number of Requests\n");

    scanf("%d",&n);

    printf("Enter the Requests sequence\n");

    for(i=0;i<n;i++)

     scanf("%d",&RQ[i]);

    printf("Enter initial head position\n");

    scanf("%d",&initial);

    // logic for sstf disk scheduling

        /\* loop will execute until all process is completed\*/

    while(count!=n)

    {

        int min=1000,d,index;

        for(i=0;i<n;i++)

        {

           d=abs(RQ[i]-initial);

           if(min>d)

           {

               min=d;

               index=i;

           }

        }

        TotalHeadMoment=TotalHeadMoment+min;

        initial=RQ[index];

        // 1000 is for max

        // you can use any number

        RQ[index]=1000;

        count++;

    }

    printf("Total head movement is %d",TotalHeadMoment);

    return 0;

}