

NAME: Siddhesh Avinash Dhinge
ROLL NO.: 33310
CLASS: TE11
BATCH: L11

CODE:

```
#include<stdio.h>
int absoluteValue(int); // Declaring function absoluteValue

void main()
{
    int queue[25],n,headposition,i,j,k,seek=0, maxrange,
    difference,temp,queue1[20],queue2[20],temp1=0,temp2=0;
    float averageSeekTime;

    // Reading the maximum Range of the Disk.
    printf("Enter the maximum range of Disk: ");
    scanf("%d",&maxrange);

    // Reading the number of Queue Requests(Disk access requests)
    printf("Enter the number of queue requests: ");
    scanf("%d",&n);

    // Reading the initial head position.(ie. the starting point of execution)
    printf("Enter the initial head position: ");
    scanf("%d",&headposition);

    // Reading disk positions to be read in the order of arrival
    printf("Enter the disk positions to be read(queue): ");
    for(i=1;i<=n;i++) // Note that i varies from 1 to n instead of 0 to n-1
    {
        scanf("%d",&temp); //Reading position value to a temporary variable

        //Now if the requested position is greater than current headposition,
        //then pushing that to array queue1
        if(temp>headposition)
        {
            queue1[temp1]=temp; //temp1 is the index variable of queue1[]
            temp1++; //incrementing temp1
        }
        else //else if temp < current headposition,then push to array queue2[]
        {
            queue2[temp2]=temp; //temp2 is the index variable of queue2[]
            temp2++;
        }
    }

    //Now we have to sort the two arrays
    //SORTING array queue1[] in ascending order
    for(i=0;i<temp1-1;i++)
    {
```

```

    for(j=i+1;j<temp1;j++)
    {
        if(queue1[i]>queue1[j])
        {
            temp=queue1[i];
            queue1[i]=queue1[j];
            queue1[j]=temp;
        }
    }
}

```

//SORTING array queue2[] in ascending order

```

for(i=0;i<temp2-1;i++)
{
    for(j=i+1;j<temp2;j++)
    {
        if(queue2[i]>queue2[j])
        {
            temp=queue2[i];
            queue2[i]=queue2[j];
            queue2[j]=temp;
        }
    }
}

```

//Copying first array queue1[] into queue[]

```

for(i=1,j=0;j<temp1;i++,j++)
{
    queue[i]=queue1[j];
}

```

//Moving Disk head to the inner most requested cylinder,

//because this is Circular LOOK.

```
queue[i]=queue2[0];
```

//Copying second array queue2[] after that first one is copied, into queue[]

```

for(i=temp1+1,j=0;j<temp2;i++,j++)
{
    queue[i]=queue2[j];
}

```

//At this point, we have the queue[] with the requests in the

//correct order of execution as per C-LOOK algorithm.

//Now we have to set 0th index of queue[] to be the initial headposition.

```
queue[0]=headposition;
```

// Calculating SEEK TIME. seek is initially set to 0 in the declaration part.

for(j=0; j<n; j++) //Loop starts from headposition. (ie. 0th index of queue)

```

{
    // Finding the difference between next position and current position.
    difference = absoluteValue(queue[j+1]-queue[j]);
}

```

```

// Adding difference to the current seek time value
seek = seek + difference;

// Displaying a message to show the movement of disk head
printf("Disk head moves from position %d to %d with Seek %d \n",
queue[j], queue[j+1], difference);
}

// Calculating Average Seek time
averageSeekTime = seek/(float)n;

//Display Total and Average Seek Time(s)
printf("Total Seek Time= %d\n", seek);
printf("Average Seek Time= %f\n", averageSeekTime);
}

// Defining function absoluteValue
int absoluteValue(int x)
{
    if(x>0)
    {
        return x;
    }
    else
    {
        return x*-1;
    }
}

```

OUTPUT:

```

Enter the maximum range of Disk: 199
Enter the number of queue requests: 8
Enter the initial head position: 53
Enter the disk positions to be read(queue): 98
183
41
122
14
124
65
67
Disk head moves from position 53 to 65 with Seek 12
Disk head moves from position 65 to 67 with Seek 2
Disk head moves from position 67 to 98 with Seek 31
Disk head moves from position 98 to 122 with Seek 24
Disk head moves from position 122 to 124 with Seek 2
Disk head moves from position 124 to 183 with Seek 59
Disk head moves from position 183 to 14 with Seek 169
Disk head moves from position 14 to 41 with Seek 27
Total Seek Time= 326
Average Seek Time= 40.750000

```