Name:- Pritika Chaudhary
Roll No:- 2023078
Student Id:- 20052056
Date:- 27 August, 2021
Course:- B.Sc (H)
Semester:- 2
Section:- B
Subject Name:- Operating System Practical
Subject Code:- PBII-202

Q2.

Algorithm

Step 1:- Start
Step 2:- Let request array represent an array storing indexes of tracks that have been requested in ascending order of their time of arrival... 'head' is the position of disk head.
Step 3:- Let direction represents whether the head is moving towards left or right.
Step 4:- In the direction in which head is moving service all tracks one by one.
Step 5:- Calculate the absolute distance and the tracks from the head.
Step 6:- Increment the total seek count with this instance
Step 7:- Currently serviced track position now becomes the new head position.
Step 8:- Go to step 4 until we reach at one of the end of the disk
Step 9:- If we reach at the end of the disk reverse the direction & go to step 3 until all tracks in request array have not been serviced.
Step 10:- Stop

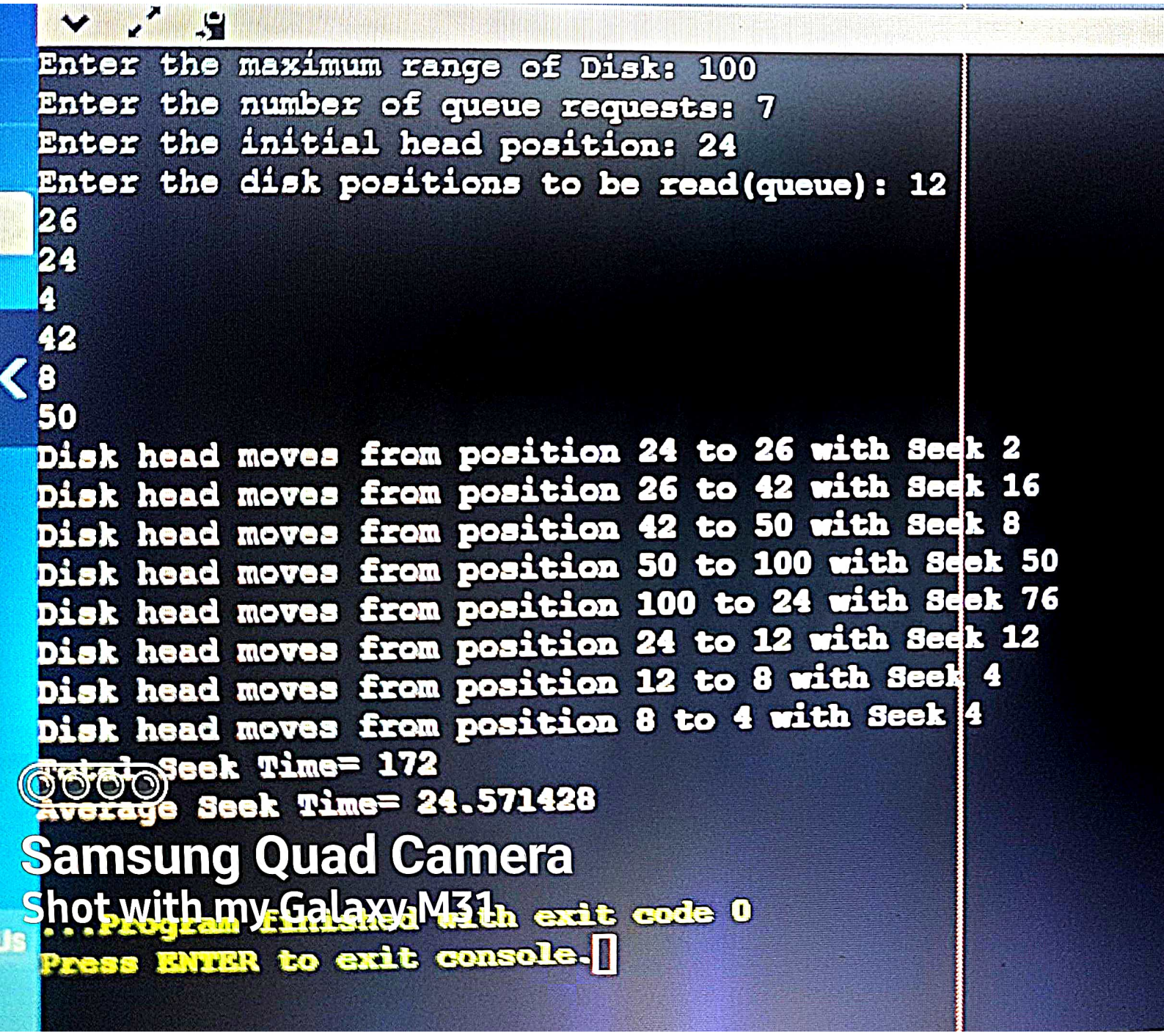
Pritika

Coding

```
#include<stdio.h>
int absolute value(int);
void main()
{
int queue[25], n, headposition, i, j, k, seek=0;
maxrange, diff, temp, que1[50], que2[20], temp1,
temp2=0;
float avg seek time;
printf("Enter the max range of disk:");
scanf("%d",&n);
printf("Enter the number of queue request:");
scanf("%d",&n);
printf("Enter the initial head position:");
scanf("%d",&headposition);
printf("Enter the disk position to be read(queue
for(i=1;i<n;i++)
{
scanf("%d",&temp);
if temp > headposition)
{
queue1[temp1]=temp;
temp1++;
}
else
{
queue2[temp2]=temp;
temp2++;
}}
```

```
for(i=0; i<temp1-1; i++)
{
for(j=i+1; j<temp1; j++)
{
if queue1[i]>queue1[i]
{
temp=queue1[i];
queue1[i]=queue[i];
queue[i]=temp;
}
}
}
for(i=0; i<temp2-1; i++)
{
for(j=i+1; j<temp2; j++)
{
if queue2[i]>queue2[i]
{
temp=queue2[i];
queue2[i]=queue2[i];
queue2[i]=temp;
}
}
}
for(i=1; j=0; j<temp1; i++, j++)
{
queue[i]=queue1[j];
}
queue[i]=maxrange;
for(j=temp1+2 j=0; j<temp2; i++, j++)
{
```

```
{
queue[i] = queue2[j-1];
}
queue[i] = 0;
queue[0] = headposition;
for(j=0; j<=n; j++)
{
diff = absoluteValue(queue[j+1] - queue[j]);
seek = seek + difference;
printf("\nDisk head moves from position %d
to %d with seek %d \n", queue[j-1], queue
[j+1], difference);
}
average seektime = seek/(float) n;
printf("\nTotal seek time = %d\n", seek);
printf("\nAverage Seek Time = %f\n", average
seektime);
}
int absoluteValue(int x)
{
if(x>0)
{
return x;
}
else
{
return x*-1;
}
}
```


Enter the maximum range of Disk: 100
Enter the number of queue requests: 7
Enter the initial head position: 24
Enter the disk positions to be read(queue): 12
26
24
4
42
8
50
Disk head moves from position 24 to 26 with Seek 2
Disk head moves from position 26 to 42 with Seek 16
Disk head moves from position 42 to 50 with Seek 8
Disk head moves from position 50 to 100 with Seek 50
Disk head moves from position 100 to 24 with Seek 76
Disk head moves from position 24 to 12 with Seek 12
Disk head moves from position 12 to 8 with Seek 4
Disk head moves from position 8 to 4 with Seek 4
Total Seek Time= 172
Average Seek Time= 24.571428
Samsung Quad Camera
Shot with my Galaxy M31
...Program finished with exit code 0
Press ENTER to exit console.