

Akki maan

Roll no - 2023023

Sid 20051071

Practical - OS

```
#include <stdio.h>
```

```
int absolute Value(int);
```

```
void main()
```

```
{
```

```
int
```

```
queue [25], n, headPosition, i, j, k, seek = 0, maxRange,  
difference, temp, queue 1 [20], queue 2 [20], temp 1 = 0,  
temp 2 = 0;
```

```
float average Seek Time;
```

```
printf("Enter the maximum range of Disk: ");
```

```
scanf("%d", &maxRange);
```

```
printf("Enter the number of queue requests: ");
```

```
scanf("%d", &n);
```

```
printf("Enter the initial head Position: ");
```

```
scanf("%d", &headPosition);
```

```
printf("Enter the initial headdisk Position to be ready (queue): ");
```

```
scanf("%d",
```

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

```
{
```

```
scanf("%d", &temp);
```

```
if(temp > headPosition)
```

```
{
```

```
queue1[temp1] = temp;
```

```
temp1++
```

```
}
```

```
else
```

```
{
```

```
queue 2 [temp 2] = temp;
```

```
temp 2++;
```

```
}
```

```

}
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;
        }
    }
}

```

```

}
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;
        }
    }
}

```

```

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

```

```

for (i = temp1 + 2, j = 0; j < temp2; i++, j++)
{
    queue[i] = queue2[j];
}
queue[i] = 0;
queue[0] = headPosition;
for (j = 0; j <= n; j++)
{
    difference = absoluteValue(queue[j+1] - queue[j]);

    seek = seek + difference;
    printf("Disk head moves from position %d to %d with  

    seek %d \n",
           queue[j], queue[j+1], difference);

    seek = seek + difference;
    printf("Disk head moves from position %d to %d  

    with seek %d \n",
           queue[j], queue[j+1], difference);
}
averageSeekTime = seek / (float)n;
printf("Total Seek Time = %d \n", seek);
printf("Average Seek Time = %f \n", averageSeekTime);
}

int absoluteValue(int x)
{
    if (x > 0)
        return x;
    else
        return -x;
}

```

else

{

return $x^* - 1$

}

}