**NAME- DHRUV DHABALIA**

**SRN- 202101281**

**ROLL NO- 27**

**DIV- E**

**ASSIGNMENT-11**

**Implement in C language following disk scheduling algorithm**

1. **First Come First Serve (FCFS) disk scheduling algorithm**
2. **Shortest Seek Time First (SSTF) disk scheduling algorithm**
3. **SCAN disk scheduling algorithm**
4. **Circular SCAN (C-SCAN) disk scheduling algorithm**
5. **First Come First Serve (FCFS) disk scheduling algorithm**

CODE:

#include <stdio.h>

#include <stdlib.h>

void main()

{

    int n, i, totalDiff = 0;

    float avgSeekTime;

    printf("\nFirst Come First Serve (FCFS) Disk Scheduling Algorithm\n");

    printf("\nEnter the number of tracks: ");

    scanf("%d", &n);

    int track[n];

    printf("Enter the requested tracks:\n");

    for (i = 0; i < n; i++)

    {

        printf("Track %d: ", i + 1);

        scanf("%d", &track[i]);

    }

    for (i = 0; i < n - 1; i++)

    {

        totalDiff += abs(track[i] - track[i + 1]);

    }

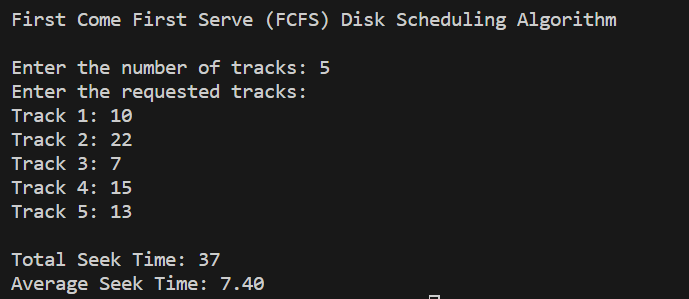
    avgSeekTime = (float)totalDiff / n;

    printf("\nTotal Seek Time: %d", totalDiff);

    printf("\nAverage Seek Time: %.2f\n", avgSeekTime);

}

OUTPUT:



**II. Shortest Seek Time First (SSTF) disk scheduling algorithm**

CODE:

#include <stdio.h>

#include <stdlib.h>

void main()

{

    int n, i, totalDiff = 0, startTrack;

    float avgSeekTime;

    printf("\nShortest Seek Time First (SSTF) Disk Scheduling Algorithm\n");

    printf("\nEnter the number of tracks: ");

    scanf("%d", &n);

    int track[n];

    printf("Enter the requested tracks:\n");

    for (i = 0; i < n; i++)

    {

        printf("Track %d: ", i + 1);

        scanf("%d", &track[i]);

    }

    printf("Enter the starting track: ");

    scanf("%d", &startTrack);

    int trackDiff[n];

    for (i = 0; i < n; i++)

    {

        trackDiff[i] = abs(startTrack - track[i]);

    }

    for (i = 0; i < n - 1; i++)

    {

        totalDiff += trackDiff[i];

    }

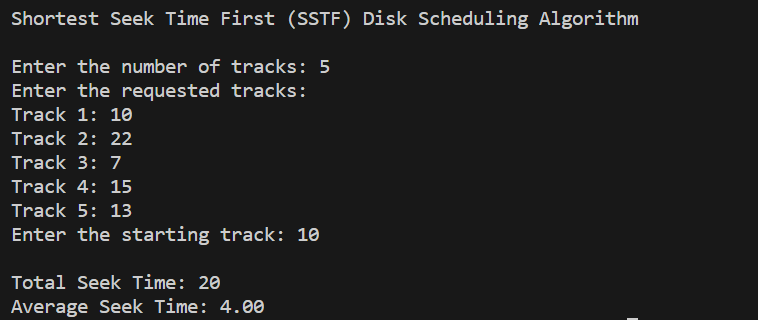
    avgSeekTime = (float)totalDiff / n;

    printf("\nTotal Seek Time: %d", totalDiff);

    printf("\nAverage Seek Time: %.2f\n", avgSeekTime);

}

OUTPUT:



**III. SCAN disk scheduling algorithm**

CODE:

#include <stdio.h>

#include <stdlib.h>

void main()

{

    int n, i, totalDiff = 0, startTrack;

    float avgSeekTime;

    printf("\nSCAN Disk Scheduling Algorithm\n");

    printf("\nEnter the number of tracks: ");

    scanf("%d", &n);

    int track[n];

    printf("Enter the requested tracks:\n");

    for (i = 0; i < n; i++)

    {

        printf("Track %d: ", i + 1);

        scanf("%d", &track[i]);

    }

    printf("Enter the starting track: ");

    scanf("%d", &startTrack);

    int trackDiff[n];

    for (i = 0; i < n; i++)

    {

        trackDiff[i] = abs(startTrack - track[i]);

    }

    for (i = 0; i < n - 1; i++)

    {

        for (int j = i + 1; j < n; j++)

        {

            if (trackDiff[i] > trackDiff[j])

            {

                int temp = trackDiff[i];

                trackDiff[i] = trackDiff[j];

                trackDiff[j] = temp;

            }

        }

    }

    for (i = 0; i < n - 1; i++)

    {

        totalDiff += trackDiff[i];

    }

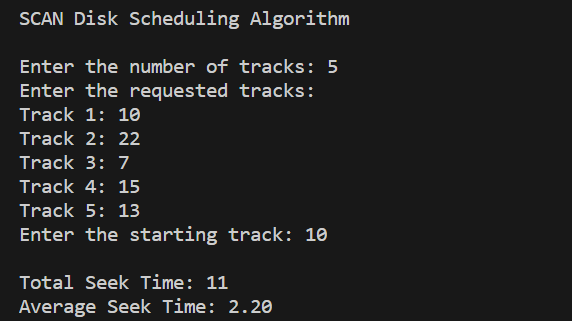
    avgSeekTime = (float)totalDiff / n;

    printf("\nTotal Seek Time: %d", totalDiff);

    printf("\nAverage Seek Time: %.2f\n", avgSeekTime);

}

OUTPUT:



**IV. Circular SCAN (C-SCAN) disk scheduling algorithm**

CODE:

#include <stdio.h>

#include <stdlib.h>

void main()

{

    int n, i, totalDiff = 0, startTrack;

    float avgSeekTime;

    printf("\nCircular SCAN (C-SCAN) Disk Scheduling Algorithm\n");

    printf("\nEnter the number of tracks: ");

    scanf("%d", &n);

    int track[n];

    printf("Enter the requested tracks:\n");

    for (i = 0; i < n; i++)

    {

        printf("Track %d: ", i + 1);

        scanf("%d", &track[i]);

    }

    printf("Enter the starting track: ");

    scanf("%d", &startTrack);

    int trackDiff[n];

    for (i = 0; i < n; i++)

    {

        trackDiff[i] = abs(startTrack - track[i]);

    }

    for (i = 0; i < n - 1; i++)

    {

        for (int j = i + 1; j < n; j++)

        {

            if (trackDiff[i] > trackDiff[j])

            {

                int temp = trackDiff[i];

                trackDiff[i] = trackDiff[j];

                trackDiff[j] = temp;

            }

        }

    }

    for (i = 0; i < n - 1; i++)

    {

        totalDiff += trackDiff[i];

    }

    totalDiff += (startTrack - 0) + (n - 1);

    avgSeekTime = (float)totalDiff / n;

    printf("\nTotal Seek Time: %d", totalDiff);

    printf("\nAverage Seek Time: %.2f\n", avgSeekTime);

}

OUTPUT:

