Nagar Yuwak Shikshan Sanstha’s

Yeshwantrao Chavan College of Engineering

(An Autonomous Institution affiliated to Rashtrasant Tukadoji Maharaj Nagpur University)

Hingna Road, Wanadongri, Nagpur - 441 110
NAAC A++

Department of Computer Technology

Vision of the Department

To be a well-known centre for pursuing computer education through innovative pedagogy, value-based education and industry collaboration.

Mission of the Department

To establish learning ambience for ushering in computer engineering professionals in core and multidisciplinary area by developing Problem-solving skills through emerging technologies.

Session 2025-2026

Vision: Dream of where you want.	**Mission:** Means to achieve Vision

Program Educational Objectives of the program (PEO): (broad statements that describe the professional and career accomplishments)

PEO1	**Preparation**	**P: Preparation**	**Pep-CL abbreviation pronounce as Pep-si-lL easy to recall**
PEO2	**Core Competence**	**E: Environment (Learning Environment)**	
PEO3	**Breadth**	**P: Professionalism**	
PEO4	**Professionalism**	**C: Core Competence**	
PEO5	**Learning Environment**	**L: Breadth (Learning in diverse areas)**	

Program Outcomes (PO): (statements that describe what a student should be able to do and know by the end of a program)

Keywords of POs:

Engineering knowledge, Problem analysis, Design/development of solutions, Conduct Investigations of Complex Problems, Engineering Tool Usage, The Engineer and The World, Ethics, Individual and Collaborative Team work, Communication, Project Management and Finance, Life-Long Learning

PSO Keywords: Cutting edge technologies, Research

“I am an engineer, and I know how to apply engineering knowledge to investigate, analyse and design solutions to complex problems using tools for entire world following all ethics in a collaborative way with proper management skills throughout my life.” *to contribute to the development of cutting-edge technologies and Research*.

Integrity: I will adhere to the Laboratory Code of Conduct and ethics in its entirety.

Name and Signature of Student and Date

(Signature and Date in Handwritten)

Nagar Yuwak Shikshan Sanstha's

Yeshwantrao Chavan College of Engineering

(An Autonomous Institution affiliated to Rashtrasant Tukadoji Maharaj Nagpur University)

Hingna Road, Wanadongri, Nagpur - 441 110
NAAC A++

Department of Computer Technology

Vision of the Department

To be a well-known centre for pursuing computer education through innovative pedagogy, value-based education and industry collaboration.

Mission of the Department

To establish learning ambience for ushering in computer engineering professionals in core and multidisciplinary area by developing Problem-solving skills through emerging technologies.

Session	**2025-26 (ODD)**	**Course Name**	**Operating System Lab**
Semester	**5**	**Course Code**	**23IOT1504**
Roll No	**42**	**Name of Student**	**Karan Chopkar**

Practical Number	7
Course Outcome	1. **CO4:** Simulate and implement disk scheduling algorithms such as FCFS and SSTF. 2. **CO4:** Calculate and compare total head movement to evaluate the efficiency of different scheduling techniques. 3. **CO4:** Understand how effective disk scheduling enhances overall system performance and resource utilization.
Aim	Implement a program to simulate disk scheduling algorithms. A. FCFS B. SSTF
Problem Definition	Design and implement a program to simulate different disk scheduling algorithms that determine the order in which pending disk I/O requests are processed. The program should calculate the total head movement and display the sequence in which tracks are accessed using the FCFS and SSTF methods.
Theory (100 words)	Disk scheduling algorithms are used in operating systems to decide the order of servicing I/O requests from multiple processes. When several processes request access to different tracks on the disk, the scheduling method determines how the disk head moves to fulfill these requests efficiently. **Objective:** • To minimize total head movement, average seek time, and response time. • To ensure fairness and improve the overall efficiency of disk operations.

1. First Come First Serve (FCFS):
   - In FCFS, the requests are handled in the exact order they arrive. The disk head moves from one requested track to the next without reordering.
   - Advantage: Simple to implement and fair to all requests.
   - Disadvantage: Can result in large total head movement if requests are far apart, leading to poor performance.
2. Shortest Seek Time First (SSTF):
   - In SSTF, the request that is closest to the current head position is selected next. This reduces the overall seek time since the nearest track is always served first.
   - Advantage: Minimizes total head movement and average seek time.
   - Disadvantage: May lead to starvation of requests that are far from the current head position.

Applications:

These algorithms are essential in optimizing disk performance, improving throughput, and reducing latency in systems where multiple processes compete for disk access.

Procedure and Execution

(100 Words)

Step for Implementation:

1. Start the program and input the total number of requests and their track numbers.
2. Enter the initial position of the disk head.
3. Implement two functions:
   - Simulate_fcfs() for the FCFS algorithm.
   - simulate_sstf() for the SSTF algorithm.
4. In FCFS:
   - Service requests in the order they are received.
   - Calculate total head movement.
5. In SSTF:
   - Select the request closest to the current head each time.
   - Mark served requests to avoid repetition.
6. Display the order of head movement and the total number of tracks traversed.
7. End the program after displaying both algorithm results.

Code:

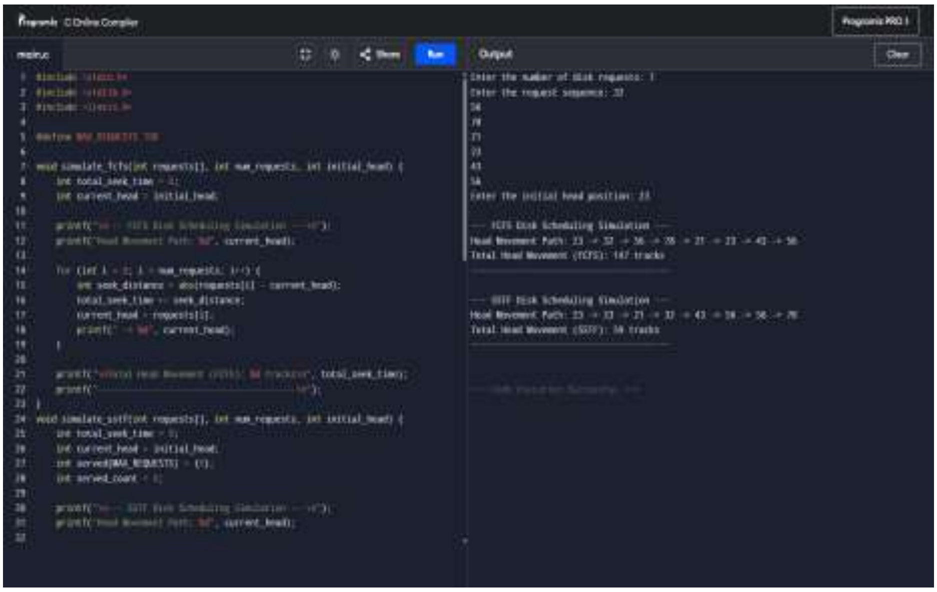
```
#include <stdio.h>

#include <stdlib.h>

#include <limits.h>
```

```
#define MAX_REQUESTS 100



void simulate_fcfs(int requests[], int num_requests,
int initial_head) {

    int total_seek_time = 0;

    int current_head = initial_head;



    printf("\n--- FCFS Disk Scheduling Simulation --
-\n");

    printf("Head Movement Path: %d",
current_head);



    for (int i = 0; i < num_requests; i++) {

        int seek_distance = abs(requests[i] -
current_head);

        total_seek_time += seek_distance;

        current_head = requests[i];

        printf(" -> %d", current_head);

    }



    printf("\nTotal Head Movement (FCFS): %d
tracks\n", total_seek_time);

    printf("--------------------------------------\n");

}

void simulate_sstf(int requests[], int num_requests,
int initial_head) {
    int total_seek_time = 0;

    int current_head = initial_head;

    int served[MAX_REQUESTS] = {0};

    int served_count = 0;



    printf("\n--- SSTF Disk Scheduling Simulation --
-\n");

    printf("Head Movement Path: %d",
current_head);



    while (served_count < num_requests) {

        int min_seek_distance = INT_MAX;

        int next_index = -1;

        for (int i = 0; i < num_requests; i++) {

            if (served[i] == 0) {

                int seek_distance = abs(requests[i] -
current_head);

                if (seek_distance < min_seek_distance) {

                    min_seek_distance = seek_distance;

                    next_index = i;

                }

            }

        }



        total_seek_time += min_seek_distance;

        current_head = requests[next_index];
        served[next_index] = 1;

        served_count++;



        printf(" -> %d", current_head);

    }



    printf("\nTotal Head Movement (SSTF): %d
tracks\n", total_seek_time);

    printf("--------------------------------------\n");

}

int main() {

    int num_requests, initial_head;

    int requests[MAX_REQUESTS];



    printf("Enter the number of disk requests: ");

    scanf("%d", &num_requests);



    printf("Enter the request sequence: ");

    for (int i = 0; i < num_requests; i++) {

        scanf("%d", &requests[i]);

    }

   printf("Enter the initial head position: ");

    scanf("%d", &initial_head);

    simulate_fcfs(requests, num_requests,
initial_head);
```

```
    simulate_sstf(requests, num_requests,
initial_head);



    return 0;
}
```

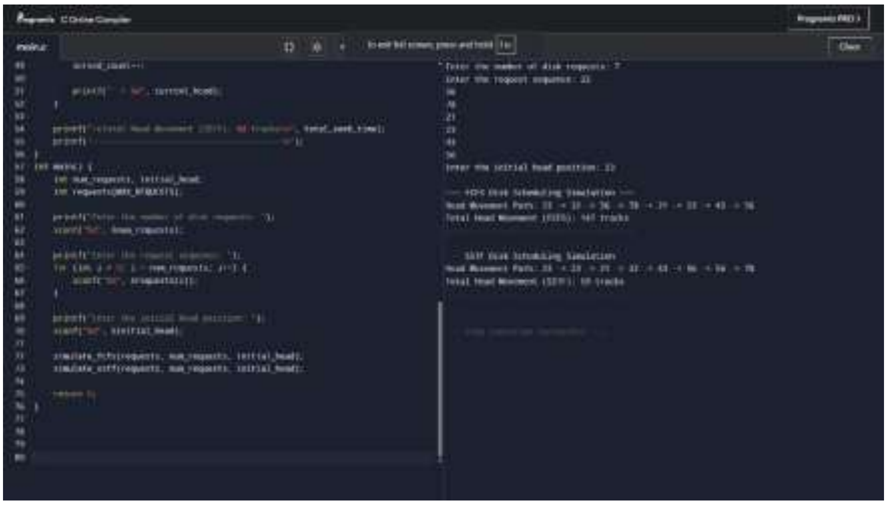
Nagar Yuwak Shikshan Sanstha's
Yeshwantrao Chavan College of Engineering
(An Autonomous Institution affiliated to Rashtrasant Tukadoji Maharaj Nagpur University)
Hingna Road, Wanadongri, Nagpur - 441 110
NAAC A++
Department of Computer Technology
Vision of the Department
To be a well-known centre for pursuing computer education through innovative pedagogy, value-based education and industry collaboration.
Mission of the Department
To establish learning ambience for ushering in computer engineering professionals in core and multidisciplinary area by developing Problem-solving skills through emerging technologies.

Output:



```c
#include <stdio.h>
#include <stdlib.h>
#include <limits.h>

#define MAX_REQUESTS 100

void simulate_fcfs(int requests[], int num_requests, int initial_head) {
    int total_seek_time = 0;
    int current_head = initial_head;

    printf("\n--- FCFS Disk Scheduling Simulation ---\n");
    printf("Head Movement Path: %d", current_head);

    for (int i = 0; i < num_requests; i++) {
        int seek_distance = abs(requests[i] - current_head);
        total_seek_time += seek_distance;
        current_head = requests[i];
        printf(" -> %d", current_head);
    }

    printf("\nTotal Head Movement (FCFS): %d tracks\n", total_seek_time);
    printf("----------------------------------------\n");
}
void simulate_sstf(int requests[], int num_requests, int initial_head) {
    int total_seek_time = 0;
    int current_head = initial_head;
    int served[MAX_REQUESTS] = {0};
    int served_count = 0;

    printf("\n--- SSTF Disk Scheduling Simulation ---\n");
    printf("Head Movement Path: %d", current_head);
```

```
Enter the number of disk requests: 7
Enter the request sequence: 32
56
78
21
23
43
56
Enter the initial head position: 23

--- FCFS Disk Scheduling Simulation ---
Head Movement Path: 23 -> 32 -> 56 -> 78 -> 21 -> 23 -> 43 -> 56
Total Head Movement (FCFS): 147 tracks
----------------------------------------


--- SSTF Disk Scheduling Simulation ---
Head Movement Path: 23 -> 23 -> 21 -> 32 -> 43 -> 56 -> 56 -> 78
Total Head Movement (SSTF): 59 tracks
----------------------------------------


=== Code Execution Successful ===
```

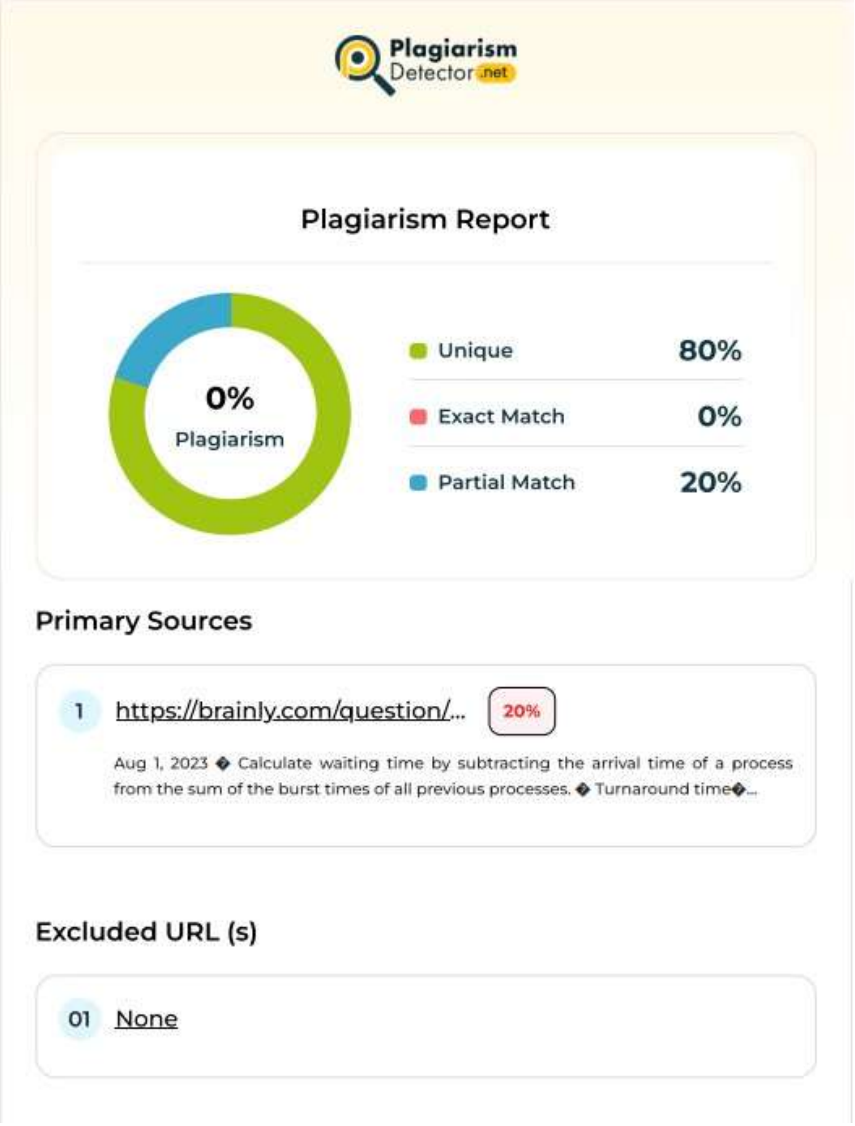


Programiz C Online Compiler
Programiz PRO
main.c
To exit full screen, press and hold Esc
Clear

```c
            served_count++;

            printf(" -> %d", current_head);
        }

    printf("\nTotal Head Movement (SSTF): %d tracks\n", total_seek_time);
    printf("----------------------------------------\n");
}
int main() {
    int num_requests, initial_head;
    int requests[MAX_REQUESTS];

    printf("Enter the number of disk requests: ");
    scanf("%d", &num_requests);

    printf("Enter the request sequence: ");
    for (int i = 0; i < num_requests; i++) {
        scanf("%d", &requests[i]);
    }

    printf("Enter the initial head position: ");
    scanf("%d", &initial_head);

    simulate_fcfs(requests, num_requests, initial_head);
    simulate_sstf(requests, num_requests, initial_head);

    return 0;
}
```

```
Enter the number of disk requests: 7
Enter the request sequence: 32
56
78
21
23
43
56
Enter the initial head position: 23

--- FCFS Disk Scheduling Simulation ---
Head Movement Path: 23 -> 32 -> 56 -> 78 -> 21 -> 23 -> 43 -> 56
Total Head Movement (FCFS): 147 tracks


--- SSTF Disk Scheduling Simulation ---
Head Movement Path: 23 -> 23 -> 21 -> 32 -> 43 -> 56 -> 56 -> 78
Total Head Movement (SSTF): 59 tracks


=== Code Execution Successful ===
```

Output Analysis	• FCFS services requests in the arrival order, resulting in higher total head movement. • SSTF chooses the nearest track each time, significantly reducing the total seek time.
Link of student Github profile where lab assignment has been uploaded	https://github.com/karan-0123/OS_LAB

Conclusion	The simulation demonstrates how different disk scheduling algorithms affect disk performance. - **FCFS** is simple and fair but not efficient in minimizing head movement. - **SSTF** improves efficiency by always selecting the closest request, leading to faster disk access. Thus, proper selection of disk scheduling algorithms can improve system throughput and overall performance.
Plag Report (Similarity index < 12%)	 Plagiarism Detector .net Plagiarism Report 0% Plagiarism 🟩 Unique 80% 🟥 Exact Match 0% 🟦 Partial Match 20% Primary Sources 1 https://brainly.com/question/... 20% Aug 1, 2023 � Calculate waiting time by subtracting the arrival time of a process from the sum of the burst times of all previous processes. � Turnaround time�... Excluded URL (s) 01 None
Date	01/11/25