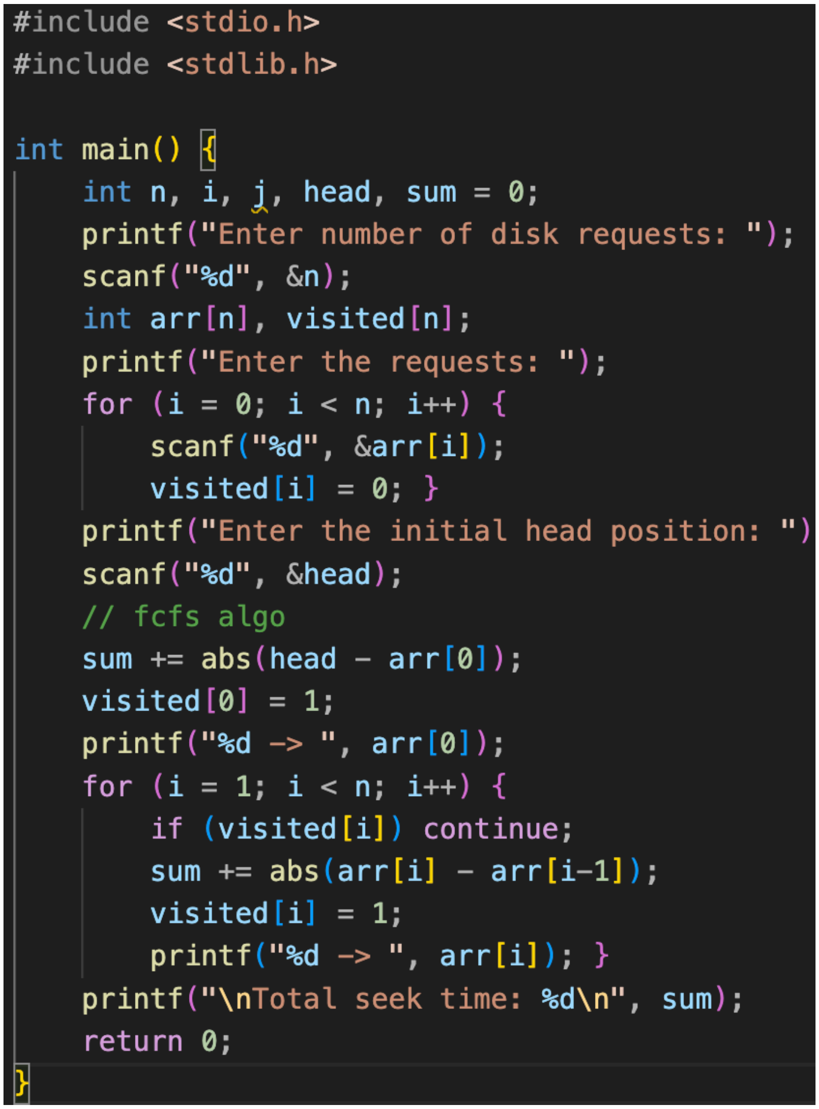
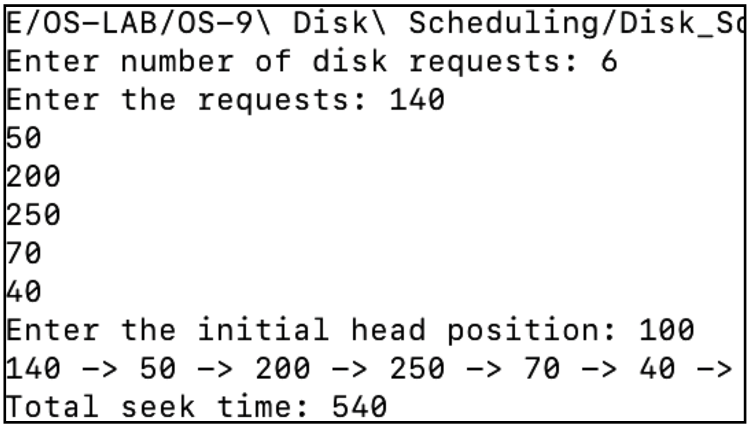
EXPERIMENT - 9 :DISK SCHEDULING ALGORITHM

1). First Come First Serve (FCFS)

it’s a disk scheduling algorithm that services disk requests in the order in which they arrive. It is one of the simplest disk scheduling algorithms and is easy to implement. However, it can result in long seek times if the requests are not ordered optimally. In FCFS, the head moves from its current position to the first request in the queue, and then to the second request, and so on until all the requests have been serviced. This can lead to a phenomenon called the "convoy effect", where a large number of requests located far apart from each other can cause the head to move back and forth across the disk, resulting in long seek times and slow performance.



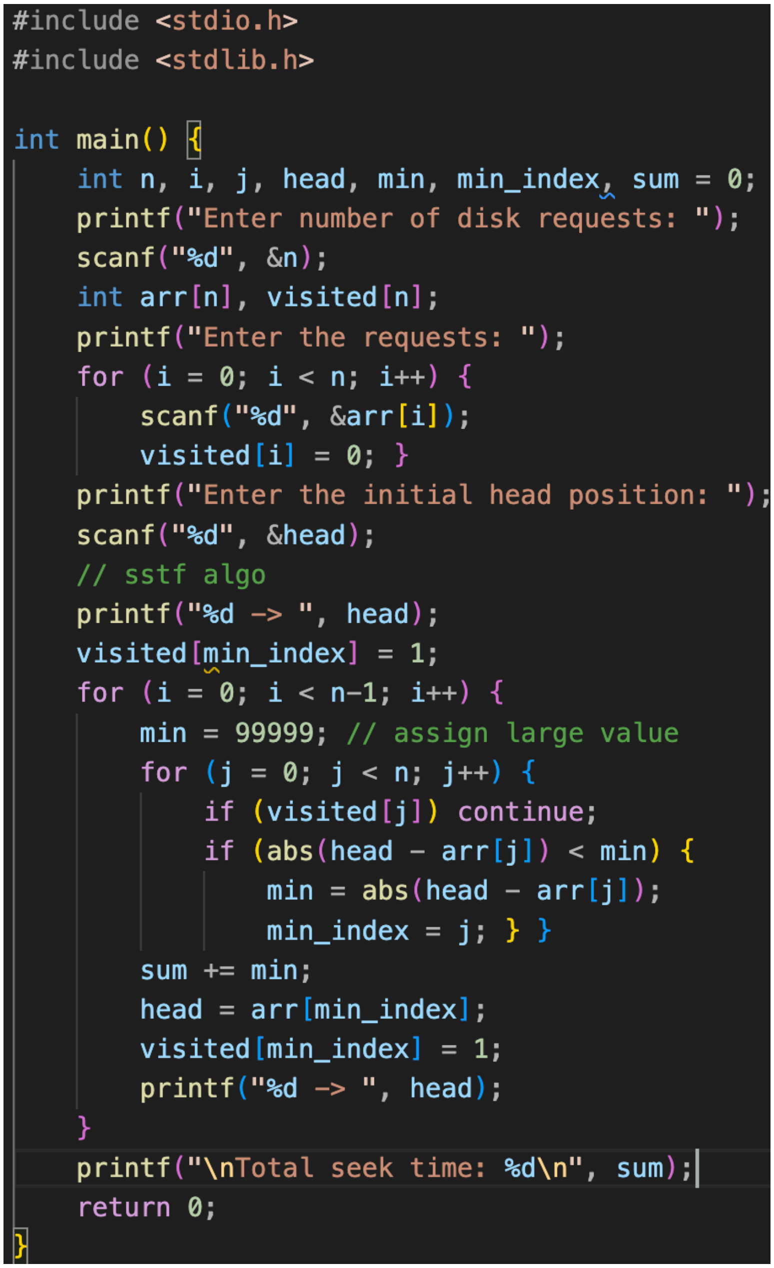
```c
#include <stdio.h>
#include <stdlib.h>

int main() {
    int n, i, j, head, sum = 0;
    printf("Enter number of disk requests: ");
    scanf("%d", &n);
    int arr[n], visited[n];
    printf("Enter the requests: ");
    for (i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
        visited[i] = 0; }
    printf("Enter the initial head position: ")
    scanf("%d", &head);
    // fcfs algo
    sum += abs(head - arr[0]);
    visited[0] = 1;
    printf("%d -> ", arr[0]);
    for (i = 1; i < n; i++) {
        if (visited[i]) continue;
        sum += abs(arr[i] - arr[i-1]);
        visited[i] = 1;
        printf("%d -> ", arr[i]); }
    printf("\nTotal seek time: %d\n", sum);
    return 0;
}
```



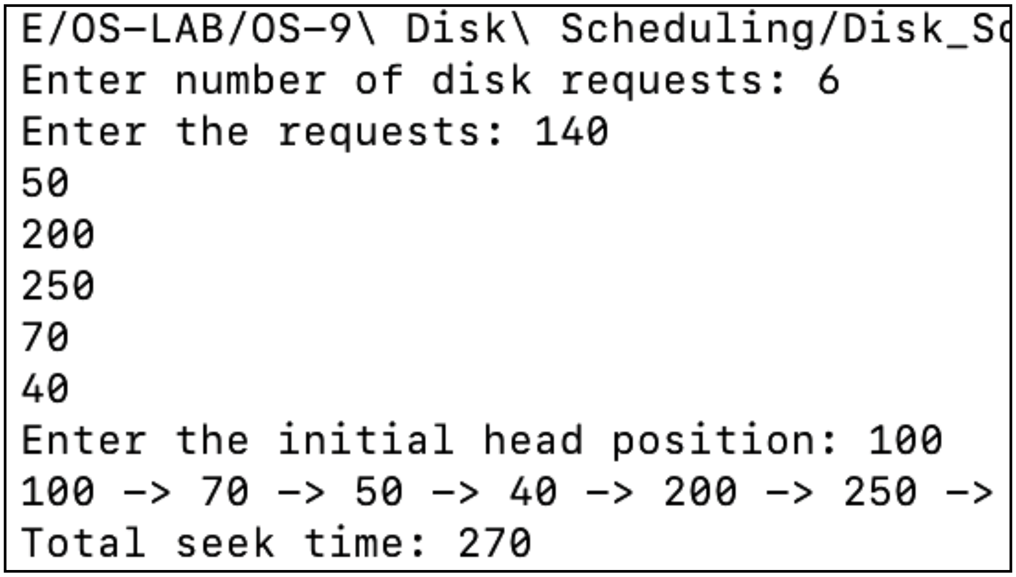
```
E/OS-LAB/OS-9\ Disk\ Scheduling/Disk_S
Enter number of disk requests: 6
Enter the requests: 140
50
200
250
70
40
Enter the initial head position: 100
140 -> 50 -> 200 -> 250 -> 70 -> 40 ->
Total seek time: 540
```

2). Shortest Seek Time First (SSTF)

It’s a disk scheduling algorithm that services disk requests based on the shortest seek time between the current head position and the next request. This algorithm always chooses the request that is closest to the head position, which minimizes the total seek time and improves the overall performance of the system. The SSTF algorithm is more efficient than the FCFS algorithm and can help to reduce the convoy effect. However, it can also lead to starvation of some requests if the closest request to the head position is constantly changing, which can cause long wait times for requests located far away from the current head position



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

int main() {
    int n, i, j, head, min, min_index, sum = 0;
    printf("Enter number of disk requests: ");
    scanf("%d", &n);
    int arr[n], visited[n];
    printf("Enter the requests: ");
    for (i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
        visited[i] = 0; }
    printf("Enter the initial head position: ");
    scanf("%d", &head);
    // sstf algo
    printf("%d -> ", head);
    visited[min_index] = 1;
    for (i = 0; i < n-1; i++) {
        min = 99999; // assign large value
        for (j = 0; j < n; j++) {
            if (visited[j]) continue;
            if (abs(head - arr[j]) < min) {
                min = abs(head - arr[j]);
                min_index = j; } }
        sum += min;
        head = arr[min_index];
        visited[min_index] = 1;
        printf("%d -> ", head);
    }
    printf("\nTotal seek time: %d\n", sum);
    return 0;
}
```



```
E/OS-LAB/OS-9\ Disk\ Scheduling/Disk_Sc
Enter number of disk requests: 6
Enter the requests: 140
50
200
250
70
40
Enter the initial head position: 100
100 -> 70 -> 50 -> 40 -> 200 -> 250 ->
Total seek time: 270
```