

```

//FIFO Disk Scheduling Algorithm :

#include <stdio.h>
#include <stdlib.h>
int main() {
int n, i, head, total_movement = 0;
printf("Enter number of disk requests: ");
scanf("%d", &n);
int requests[n];
printf("Enter the request sequence: ");
for (i = 0; i < n; i++) {
scanf("%d", &requests[i]);
}
printf("Enter initial head position: ");
scanf("%d", &head);
printf("\nDisk Sequence: %d", head);
for (i = 0; i < n; i++) {
total_movement += abs(requests[i] - head);
head = requests[i];
printf(" %d", head);
}
printf("\n\nTotal Head Movement: %d", total_movement);
printf("\nAverage Seek Time: %.2f\n", (float)total_movement / n);
return 0;
}

```

SCAN Disk Scheduling Algorithm :

```

#include <stdio.h>
#include <stdlib.h>
int main() {
int n, i, j, head, total_movement = 0, disk_size, direction, index = 0;
printf("Enter number of disk requests: ");
scanf("%d", &n);
int requests[n];
printf("Enter the request sequence: ");
for (i = 0; i < n; i++)
scanf("%d", &requests[i]);
printf("Enter initial head position: ");
scanf("%d", &head);
printf("Enter total disk size: ");
scanf("%d", &disk_size);
printf("Enter head movement direction (1 for high/right, 0 for low/left): ");
scanf("%d", &direction);
// Add only the required boundary (0 or max disk size)
int size = n + 1;
int arr[size];
for (i = 0; i < n; i++)
arr[i] = requests[i];
if (direction == 1)
arr[n] = disk_size - 1; // only max end
else
arr[n] = 0; // only min end
// Sort the array
for (i = 0; i < size - 1; i++) {
for (j = i + 1; j < size; j++) {
if (arr[i] > arr[j]) {
int temp = arr[i];
arr[i] = arr[j];
arr[j] = temp;
}
}
}

```

```
}

// Find the index where head should start
for (i = 0; i < size; i++) {
if (head < arr[i]) {
index = i;
break;
}
printf("\nSeek Sequence: %d", head);
if (direction == 1) { // Move right first
for (i = index; i < size; i++) {
total_movement += abs(head - arr[i]);
head = arr[i];
printf(" %d", head);
}
for (i = index - 1; i >= 0; i--) {
total_movement += abs(head - arr[i]);
head = arr[i];
printf(" %d", head);
}
} else { // Move left first
for (i = index - 1; i >= 0; i--) {
total_movement += abs(head - arr[i]);
head = arr[i];
printf(" %d", head);
}
}
printf("\n\nTotal Head Movement: %d", total_movement);
printf("\nAverage Seek Time: %.2f\n", (float)total_movement / n);
return 0;
}
```