To stimulate disk scheduling algorithms.

1) fcfs

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

void fcfs (int request_queue[], int n, int head){
    int total_movement = abs(head - request_queue[0]);
    for (i=1; i<n; i++){
    total_movement += abs(request_queue[i] - request_queue[i-1]);
    printf("Total head movement %d\n", total);
    }

int main(){
    int n, head;
    printf("Enter no of request: ");
    scanf("%d", &n);
    int request_queue[n];
    printf("Enter request queue\n");
    for(i=0; i<n; i++){
    scanf("%d", &request_queue[i]);
    }
    printf("Enter the initial head position");
    scanf("%d", &head);
    fcfs(request_queue, n, head);
    return 0;
}
```

output

```
Enter no of requests: 8
Enter request sequence 95 180 34 119 11 123 62 64
Enter initial head position 50
Total head movement is 644
```

PAGE NO :
DATE :

2) SCAN

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

void scan(int request_queue[], int n, int head);
  int total_movement = 0;
  int direction;
  printf("Enter the direction:");
  scanf("%d", &direction);
if (direction == 0) {
    for (i=head ; i>=0 ; i--)
      if (i == head)!
        printf("%d", i); {
for (i=0; i<n; i++) {
    if (request_queue[i] == i) {
      total_movement += head;
      printf("0");
      for (i=0; i<head; i++) {
        if (i == head-1) {
          printf("%d", i);
for (i=0; i<n; i++) {
    if (request_queue[i] == i) {
      total_movement += abs(head-i)
      head = i
    printf("%d", i) {{{
    else {
    for (i=200; i=head; i--) {
        if (i == head+1) {
printf("%d", i);
  for (i=0; i<n; i++) {
      if (request_queue[i] == i) {
        total_movement += abs(head-i)
          head = i;
```

```
printf ("%d", i);
for (j=0; j<n; j++) {
    if (request_queue[j] == i) {
        total_movement += abs(head-i);
        head = i;
        printf ("%d", i);
    }
}
printf ("\n Total head movement %d", total_movement);
}
int main() {
    int n, head;
    printf ("Enter no. of request: ");
    scanf ("%d", &n);
    int request_queue[n];
    printf ("Enter initial head position");
    scanf ("%d", &head);
    scanf (request_queue, n, head);
    return 0;
}
```

Output:

Enter no. of request: 8
Enter request sequence: 70 120 30 60 90
Enter initial head position 70
Enter total disk size 200
Enter the head movement direction for hi
0.0

Total head movement is 190

3) C-SCAN

```
#include <stdio.h>
#include <stdlib.h>
void CSCAN(int request_queue[], int n, int head){
  int total_movement = 0;
  for (i = head; i < 200; i++){
      if (i == head){
    printf(" %d", i);}
  for (i = 0; i < n; i++){
   if (request_queue[i] = i){
     total_movement += abs(head - i)
        head = i;
    printf("%d", i);}}}
 total_movement += 200 - head;
   printf("200. 0");
    head = 0;
 for (i = 0; i <= 200; i++){
   printf(" %d", i);}
   for (j = 0; j < n; j++){
      if (request_queue[j] == i){
        total_movement += abs(head - i);
        head = i;
        printf("%d", i);}}}
  printf("\n Total head movement %d\n",
   Total movement);}
    int main(){
  int n, head;
  printf("Enter no of requests: ");
    Scanf("%d", &n);
 int request_queue[n];
  Printf("Enter the request_queue \n");
```

```
for (i=0; i<n; i++){
    scanf ("%d", &request_queue[i]);
printf ("Enter initial head position");
scanf ("%d", &head);
CSCAN (request_queue, n, head);
return 0;
}
```

Output:

```
Enter no. of request: 6
Enter the requests sequence: 21
Enter initial head position: 1
Enter total disk size: 3
Enter the head movement direction for high
for loop 0=1
Total head movement is 5.
```

10/10

29/8/23

```
Enter the maximum range of Disk: 200
Enter the number of queue requests: 7
Enter the initial head position: 50
Enter the disk positions to be read(queue): 82 170 43 140 24 16 19
Disk head moves from position 50 to 82 with Seek 32
Disk head moves from position 82 to 140 with Seek 58
Disk head moves from position 140 to 170 with Seek 30
Disk head moves from position 170 to 190 with Seek 20
Disk head moves from position 190 to 200 with Seek 10
Disk head moves from position 200 to 43 with Seek 157
Disk head moves from position 43 to 24 with Seek 19
Disk head moves from position 24 to 16 with Seek 8
Total Seek Time= 334
Average Seek Time= 47.714287
```

```
Enter the maximum range of Disk: 200
Enter the number of queue requests: 7
Enter the initial head position: 50
Enter the disk positions to be read(queue): 82 170 43 140 24 16 1
Disk head moves from position 50 to 82 with Seek 32
Disk head moves from position 82 to 140 with Seek 58
Disk head moves from position 140 to 170 with Seek 30
Disk head moves from position 170 to 190 with Seek 20
Disk head moves from position 190 to 200 with Seek 10
Disk head moves from position 200 to 43 with Seek 157
Disk head moves from position 43 to 24 with Seek 19
Disk head moves from position 24 to 16 with Seek 8
Total Seek Time= 334
Average Seek Time= 47.714287
```

```
Enter the number of Requests: 7
Enter the Requests sequence: 82 170 43 140 24 16 190
Enter initial head position: 50
Enter total disk size: 200
Enter the head movement direction (high = 1 and low = 0): 0
Total head movement is: 366
```