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);
for (i=1; i<n; i++){
  total movement+= abs(request_queue[i] :
   printf(" Total head movement %d \n", total)
}
int main(){
    int n, head;
    printf(" Enter no of requests: ");
    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
```

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 (j=0 ; j<n ; j++){
   if (request_queue[j] == i){
     total_movement += head;
     printf ("0");
     for (i=0 ; i<head ; i++){
       if (i== head-1){
         printf ("%d", i);
   for (j=0 ; j<n ; j++){
     if (request_queue [j] = = 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 (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 requests :");
 scanf("%d", &n);
 int requests_queue[n];
 printf("Enter initial head position");
 scanf("%d", &head);
 Scan(request_queue, n, head);
 return 0;
```

4.

Output:

```
Enter no of request : 8
Enter request sequence 90 120 30 60 50 80
Enter initial head position 70
Enter total disk size 200
Enter the head movement direction for high & low
 0.0
Total head movement is 190
```

5) 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 (j = 0 ; j < n ; j++){
      if (request_queue[j] = 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++){
    if (i = 199){
       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 requst-queue \n");
 for(i=0; i<n; i++){
   scanf("%d", &request-queue[i]);
printf("Enter initial head position");
scanf("%d", &head);
 CSAN(requst-queue, n, head);
 return 0;

}
```

Output:

```
Enter no of requst: 6
Enter the requsts sequence: 210
Enter initial head position: 1
Enter total disk size: 3
Enter the head movement direction for high 1&
for slow 0 -> 1
Total head movement :5
```

10/10

28/8/23