LAB - 11

Write a C program to simulate disk scheduling algorithms
a) SSTF
b) LOOK
c) C-LOOK

Source Code:

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

void SSTF() {
   int RQ[100], i, n, TotalHeadMoment = 0, initial, count = 0;

   printf("Enter the number of Requests\n");
   scanf("%d", &n);
   printf("Enter the Requests sequence\n");
   for (i = 0; i < n; i++)
      scanf("%d", &RQ[i]);
   printf("Enter initial head position\n");
   scanf("%d", &initial);

   // Logic for SSTF disk scheduling

   while (count != n) {
      int min = 1000, d, index;
      for (i = 0; i < n; i++) {
         d = abs(RQ[i] - initial);
         if (min > d) {
            min = d;
            index = i;
         }
      }
      TotalHeadMoment = TotalHeadMoment + min;
      initial = RQ[index];
      RQ[index] = 1000;
      count++;
```

```
    }
    printf("Total head movement is %d\n", TotalHeadMoment);
}

void C_LOOK() {
    int RQ[100], i, j, n, TotalHeadMoment = 0, initial, move;

    printf("Enter the number of Requests\n");
    scanf("%d", &n);
    printf("Enter the Requests sequence\n");
    for (i = 0; i < n; i++)
        scanf("%d", &RQ[i]);
    printf("Enter initial head position\n");
    scanf("%d", &initial);
    printf("Enter the head movement direction for high (1) and for low
(0)\n");
    scanf("%d", &move);

    // Logic for C-LOOK disk scheduling

    for (i = 0; i < n; i++) {
        for (j = 0; j < n - i - 1; j++) {
            if (RQ[j] > RQ[j + 1]) {
                int temp;
                temp = RQ[j];
                RQ[j] = RQ[j + 1];
                RQ[j + 1] = temp;
            }
        }
    }

    int index;
    for (i = 0; i < n; i++) {
        if (initial < RQ[i]) {
            index = i;
            break;
        }
    }

    if (move == 1) {
```

```
        for (i = index; i < n; i++) {
            TotalHeadMoment = TotalHeadMoment + abs(RQ[i] - initial);
            initial = RQ[i];
        }
        for (i = 0; i < index; i++) {
            TotalHeadMoment = TotalHeadMoment + abs(RQ[i] - initial);
            initial = RQ[i];
        }
    } else {
        for (i = index - 1; i >= 0; i--) {
            TotalHeadMoment = TotalHeadMoment + abs(RQ[i] - initial);
            initial = RQ[i];
        }
        for (i = n - 1; i >= index; i--) {
            TotalHeadMoment = TotalHeadMoment + abs(RQ[i] - initial);
            initial = RQ[i];
        }
    }

    printf("Total head movement is %d\n", TotalHeadMoment);
}

void LOOK() {
    int RQ[100], i, j, n, TotalHeadMoment = 0, initial, move;

    printf("Enter the number of Requests\n");
    scanf("%d", &n);
    printf("Enter the Requests sequence\n");
    for (i = 0; i < n; i++)
        scanf("%d", &RQ[i]);
    printf("Enter initial head position\n");
    scanf("%d", &initial);
    printf("Enter the head movement direction for high (1) and for low
(0)\n");
    scanf("%d", &move);

    // Logic for LOOK disk scheduling

    for (i = 0; i < n; i++) {
        for (j = 0; j < n - i - 1; j++) {
```

```
            if (RQ[j] > RQ[j + 1]) {
                int temp;
                temp = RQ[j];
                RQ[j] = RQ[j + 1];
                RQ[j + 1] = temp;
            }
        }
    }

    int index;
    for (i = 0; i < n; i++) {
        if (initial < RQ[i]) {
            index = i;
            break;
        }
    }

    if (move == 1) {
        for (i = index; i < n; i++) {
            TotalHeadMoment = TotalHeadMoment + abs(RQ[i] - initial);
            initial = RQ[i];
        }
        for (i = index - 1; i >= 0; i--) {
            TotalHeadMoment = TotalHeadMoment + abs(RQ[i] - initial);
            initial = RQ[i];
        }
    } else {
        for (i = index - 1; i >= 0; i--) {
            TotalHeadMoment = TotalHeadMoment + abs(RQ[i] - initial);
            initial = RQ[i];
        }
        for (i = index; i < n; i++) {
            TotalHeadMoment = TotalHeadMoment + abs(RQ[i] - initial);
            initial = RQ[i];
        }
    }

    printf("Total head movement is %d\n", TotalHeadMoment);
}
```

```
int main() {
   int ch;
   printf("\n 1.SSTF\t 2.LOOK\t 3.C-LOOK\t 4.EXIT\n");
   while (1) {
      printf("\nEnter your choice\n");
      scanf("%d", &ch);
      switch (ch) {
         case 1:
            SSTF();
            break;
         case 2:
            LOOK();
            break;
         case 3:
            C_LOOK();
            break;
         case 4:
            exit(0);
         default:
            printf("Invalid choice\n");
      }
   }
}
```

Output:

SSTF:

```
Enter your choice
1
Enter the number of Requests
8
Enter the Requests sequence
95 180 34 119 11 123 62 64
Enter initial head position
50
Total head movement is 236
```

LOOK:

```
Enter your choice
2
Enter the number of Requests
8
Enter the Requests sequence
95 180 34 119 11 123 62 64
Enter initial head position
50
Enter the head movement direction for high 1 and for low 0
1
Total head movement is 299
```

C-LOOK:

```
Enter your choice
3
Enter the number of Requests
8
Enter the Requests sequence
95 180 34 119 11 123 62 64
Enter initial head position
50
Enter the head movement direction for high 1 and for low 0
1
Total head movement is 322
```