Name: Om Jadhav **Roll No.:** I3275 **Div:** 2

Statement: Implement the C program for Disk Scheduling Algorithms: SSTF, SCAN, C-Look consideringthe initial head position moving away from the spindle

Code:

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

int SSTF();
int SCAN();
int CLOOK();

int main(){

        int ch, YN = 1, i, l, f=10;
        char F[10], s[25];
        for (i = 0; i < f; i++) F[i] = -1;
        do{
                system("clear");
                printf("\n\n\t********** MENU **********");
                printf("\n\n\t1:SSTF\n\n\t2:SCAN\n\n\t3:CLOOK\n\n\t4:EXIT");
                printf("\n\n\tEnter your choice: ");
                scanf("%d", &ch);
                switch (ch){
                                case 1:
                                for (i = 0; i < f; i++){
                                        F[i] = -1;
                                }
                                SSTF();
                                break;

                                case 2:
                                for (i = 0; i < f; i++){
                                        F[i] = -1;
                                }
                                SCAN();
                                break;

                                case 3:
                                for (i = 0; i < f; i++){
                                F[i] = -1;
                                }
                                CLOOK();
                                break;

                                case 4:
                                exit(0);
                }

        printf("\n\n\tDo u want to continue IF YES PRESS 1\n\n\tIF NO PRESS 0: ");
        scanf("%d", &YN);
```

```
        } while (YN == 1);

        return (0);
}

//SSTF Algorithm
int 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);

        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", TotalHeadMoment);
        return 0;
}

//SCAN Algorithm
int SCAN(){
        int RQ[100], i, j, n, TotalHeadMoment = 0, initial, size, 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 total disk size\n"); scanf("%d", &size);
        printf("Enter the head movement direction for high 1 and for low 0\n");
        scanf("%d", &move);
        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];
                }

                TotalHeadMoment = TotalHeadMoment + abs(size - RQ[i - 1] - 1);
                initial = size - 1;
                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];
                }

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

printf("Total head movement is %d", TotalHeadMoment);
return 0;
}
//C-LOOK Algorithm
int CLOOK(){

                int RQ[100], i, j, n, TotalHeadMoment = 0, initial, size, 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 total disk size\n"); scanf("%d", &size);
        printf("Enter the head movement direction for high 1 and for low 0\n");
        scanf("%d", &move);

        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", TotalHeadMoment);
return 0;
}

/*OUTPUT:


        ********** MENU **********
```

1:SSTF

2:SCAN

3:CLOOK

4:EXIT

Enter your choice: 1
Enter the number of Requests
8
Enter the Requests sequence
370
30
390
130
310
170
340
180
Enter initial head position
160
Total head movement is 530

Do u want to continue IF YES PRESS 1

IF NO PRESS 0: 1

********** MENU **********

1:SSTF

2:SCAN

3:CLOOK

4:EXIT

Enter your choice: 2
Enter the number of Requests
8
Enter the Requests sequence
370
30
390
130
310
170
340
180
Enter initial head position
160

```
Enter total disk size
400
Enter the head movement direction for high 1 and for low 0
1
Total head movement is 608

        Do u want to continue IF YES PRESS 1

        IF NO PRESS 0: 1


        ********** MENU **********

        1:SSTF

        2:SCAN

        3:CLOOK

        4:EXIT

        Enter your choice: 3
Enter the number of Requests
8
Enter the Requests sequence
370
30
390
130
310
170
340
180
Enter initial head position
160
Enter total disk size
400
Enter the head movement direction for high 1 and for low 0
1
Total head movement is 690

        Do u want to continue IF YES PRESS 1

        IF NO PRESS 0: 0
*/
```