

WEEK 11

Write a C program to simulate disk scheduling algorithms

a) FCFS

b) SCAN

c) C-SCAN

CODE :

```
#include<stdio.h>

#include<stdlib.h>

int disks;

void quicksort(int number[25], int first, int last)
{
    int i, j, pivot, temp;
    if (first < last)
    {
        pivot = first;
        i = first;
        j = last;
        while (i < j)
        {
            while (number[i] <= number[pivot] && i < last)
                i++;
            while (number[j] > number[pivot])
                j--;
```

```

        if (i < j)
        {
            temp = number[i];
            number[i] = number[j];
            number[j] = temp;
        }
    }

    temp = number[pivot];
    number[pivot] = number[j];
    number[j] = temp;
    quicksort(number, first, j - 1);
    quicksort(number, j + 1, last);
}
}

void fcfs(int arr[],int src, int n)
{
    int sseq[20],i;
    sseq[0]=abs(arr[0]-src);
    for(i=1;i<n;i++)
    sseq[i]=abs(arr[i]-arr[i-1]);
    int sum=0;
    for(i=0;i<n;i++)
    sum+=sseq[i];
    printf("\nFCFS \nTotal seek sequence: %d \nSeek Sequence: \n",sum);
    for(i=0;i<n;i++)
    printf("%d ",sseq[i]);
    printf("\n");
}

```

```

}

void cscan(int arr[], int src, int n)
{
    int i,sum=0,j,sseq[20];

    quicksort(arr, 0, n-1);

    int index;

    for (index = 0; index < n; index++) {

        if (arr[index] == src) {

            break;

        }

    }

    i=index+1;

    j=0;

    while(i<=n)

    {

        sseq[j]=abs(arr[i]-arr[i-1]);

        i++;

        j++;

    }

    sseq[j++]=abs(disks-arr[i-1]);

    i=0;

    sseq[j++]=abs(disks);

    while(i<index)

    {

        sseq[j++]=abs(arr[i]-arr[i-1]);

        i++;

    }

```

```

    for(i=0;i<(n+2);i++)
    sum+=sseq[i];

    printf("\nC-SCAN \nTotal seek sequenece: %d \nSeek Sequence: \n",sum);

    for(i=0;i<n+2;i++)

    printf("%d ",sseq[i]);

    printf("\n");
}

void scan(int arr[], int src, int n)
{
    int i,sum=0,j,sseq[20];

    quicksort(arr, 0, n-1);

    int index;

    for (index = 0; index < n; index++) {

        if (arr[index] == src) {

            break;

        }

    }

    i=index-1;

    j=0;

    while(i>=0)

    {

        sseq[j]=abs(arr[i]-arr[i+1]);

        i--;

        j++;

    }

    i=index+1;

    sseq[j++]=abs(arr[i++]-arr[0]);

```

```

while(i<=n)
{
    sseq[j++]=abs(arr[i]-arr[i-1]);
    i++;
}

for(i=0;i<n;i++)
sum+=sseq[i];

printf("\nSCAN \nTotal seek sequenece: %d \nSeek Sequence: \n",sum);

for(i=0;i<n;i++)
printf("%d ",sseq[i]);

printf("\n");
}

void main()
{
    int source, arr[20],i,n,copy[20];
    printf("Enter numebr of disks: ");
    scanf("%d",&n);
    printf("\nEnter %d values: ",n);
    for(i=0;i<n;i++)
    scanf("%d",&arr[i]);
    printf("\nEnter source position: ");
    scanf("%d",&source);
    printf("\nEnter number disks: ");
    scanf("%d",&disks);
    for(i=0;i<n;i++)
    copy[i]=arr[i];
    arr[n]=source;

```

```

copy[n]=arr[n];

fcfs(copy , source , n);

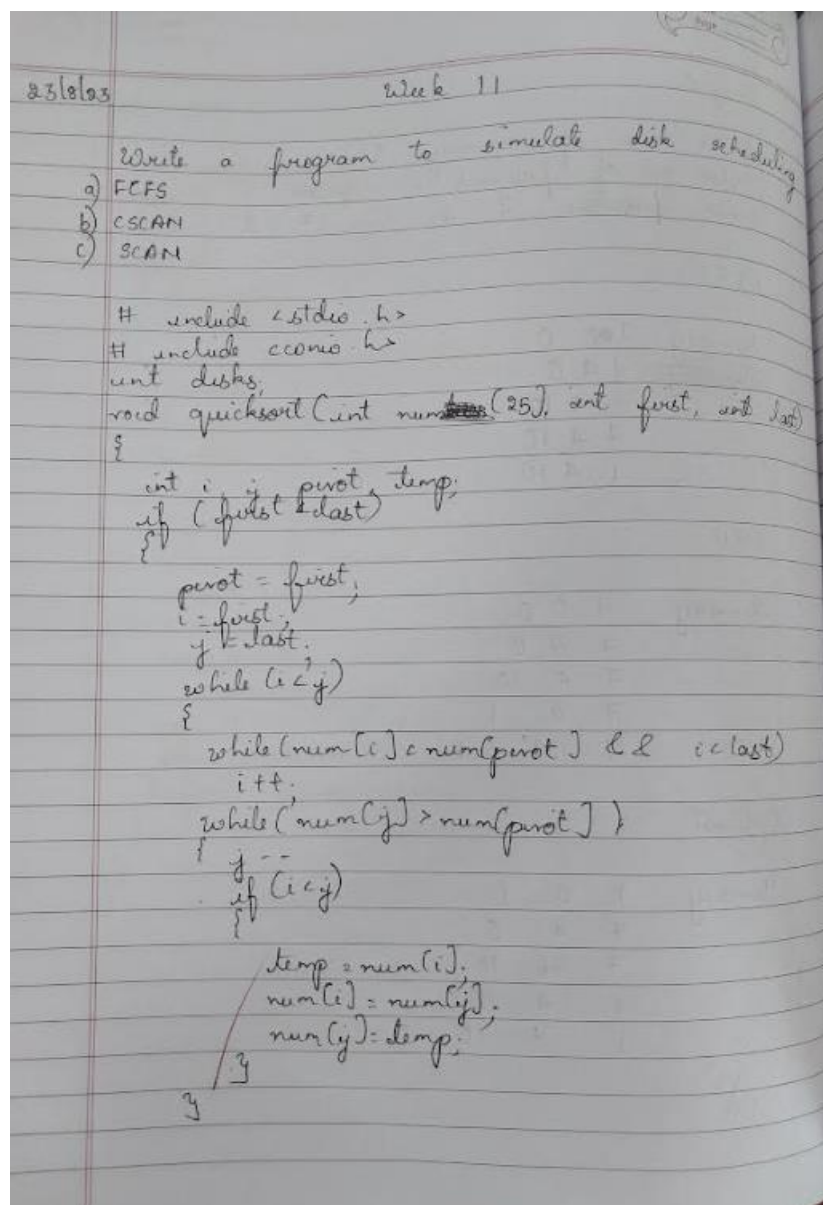
scan(copy , source , n);

cscan(arr , source , n);

}

```

OBSERVATION :



```

temp = num[prev];
num[prev] = num[y];
num[y] = temp;
quicksort (num, first, y-1);
quicksort (num, y+1, last);
}

void fdfs (int arr[], int src, int n)
{
    int sseq[20], i;
    sseq[0] = abs(arr[0] - src);
    for (i = 1; i < n; i++)
        sseq[i] = abs(arr[i] - arr[i-1]);
    int sum = 0;
    for (i = 0; i < n; i++)
        sum += sseq[i];
    int s, sumf;
    for (i = 0; i < n; i++)
        printf ("FDFS in Total seek sequence: %d in seek sequence in: sum);
    for (i = 0; i < n; i++)
        printf ("1.d", sseq[i]);
}

void scan (int arr[], int src, int n)
{
    int i, sum = 0, j, sseq[20], index;
    quicksort (arr, 0, n-1);
    for (index = 0; index < n; index++)
    {
        if (arr[index] == src)
            break;
    }
}

```

```

i = index + 1;
j = 0;
while (i < n)
{
    sseq[i] = abs(arr[i] - arr[i-1]);
    i++;
    j++;
}
sseq[j+1] = abs(dishes - arr[i-1]);
i = 0;
sseq[i+j+1] = abs(dishes);
while (i < index)
{
    sseq[i+j+1] = abs(arr[i] - arr[j-1]);
    i++;
}
for (i = 0; i < (n+2); i++)
    sum += sseq[i];
printf ("SCAN in Total seek sequence: %d in seek sequence: %d", sum);
for (i = 0; i < n+2; i++)
    printf ("1.d", sseq[i]);
}

void scan (int arr[], int src, int n)
{
    int i, sum = 0, j, sseq[20];
    quicksort (arr, 0, n-1);
    for (index = 0; index < n; index++)
    {
        if (arr[index] == src)
            break;
    }
}

```

```

i = index-1;
j = 0;
while (i >= 0)
{
    sseq[j] = abs(arr[i] - arr[i+1]);
    i--;
    j++;
}
i = index-1;
j = 0;
while (i >= 0)
{
    sseq[j] = abs(arr[i] + arr[i+1]);
    i--;
    j++;
}
for (i = 0; i < n; i++)
    sum += sseq[i];
printf("SCAN \n Total seek sequence is %d", sum);
}
}
void main()
{
    int src, arr[20], i, n, copy[20];
    printf("Enter no. of disks");
    scanf("%d", &n);
    printf("Enter values");
    for (i = 0; i < n; i++)
        scanf("%d", &arr[i]);
    printf("Enter source");
    scanf("%d", &source);
}

```

```

printf("Enter no. of disks");
scanf("%d", &n);
for (i = 0; i < n; i++)
    copy[i] = arr[i];
arr[0] = source;
copy[n] = arr;
steps(copy, source, n);
scan(copy, source, n);
}

```

Output

Enter no. of disks : 5
Enter values : 10 25 30 45 12
Enter source : 19
Enter number : 50

FCFS
Sequence 77
19 15 5 15 33

CSCAN
Sequence 118
31 50 22 13 5 18

SCAN
Sequence 61
26 15 5 13 2

OUTPUT :

```
Enter numebr of disks: 5
Enter 5 values: 10 25 30 45 12
Enter source position: 19
Enter number disks: 50

FCFS
Total seek sequence: 77
Seek Sequence:
9 15 5 15 33

C-SCAN
Total seek sequence: 118
Seek Sequence:
31 50 2 2 13 5 15

SCAN
Total seek sequence: 61
Seek Sequence:
26 15 5 13 2
```