**Problem:**

Write a C program to solve the following problem.

Suppose that a disk has 5,000 cylinders,numerbered 0 to 4999.The drive is currently serving a request at cylinder 143,and the previous request was at cylinder 125.The queue of pending request,in FIFO.

order,isc

86,1470,913,1774,948,1509,,1022,1750,130

Starting from the current head position, what is the total distance (in cylinders)that the disk arm moves to satisfy all the pending requests for each of the SCAN disk-schedulling algorithm\*/

**Explanation:**

1.I have made 3 function: cal(),enqueue(),display().

2.After the completion of the program,it will make First-in,First-out order order problem as given in the program.

3.User will get to know the total os disk and moves.

# Code Snippet

#include<stdio.h>

#include<stdlib.h> #define size 5000 int queue[size], front= -1,rear= -1; void enqueue(int);

void display(); int main()

{ int i,n; int value; printf("no of elements to insert:"); scanf("%d",&n); for(i=0;i<n;i++)

{

scanf("%d",&value); enqueue(value);

}

display();

calc();

}

void enqueue(int value)

{

if(rear==size-1)

{

printf("queue--full");

}

else

{

if(front==-1) front=0; rear++; queue[rear]=value;

}

}

void display()

{ int i;

if(rear==-1)

{

printf("queue--empty");

}

else

{

for(i=front;i<=rear;i++)

{

printf("%d\t",queue[i]);

}

}

}

void calc() {

int total\_distance=0,d,i; int current\_pos,prev\_pos; printf("\n"); printf("enter prev pos:"); scanf("%d",&prev\_pos); getchar(); printf("\n"); printf("enter current pos"); scanf("%d",&current\_pos); getchar(); d=current\_pos-queue[front]; if(d<0) total\_distance=-d; else total\_distance=d; for(i=front;i<rear;i++)

{

d=queue[i]-queue[i+1];

if(d<0)

d=-d; total\_distance=total\_distance + d;

}

printf("\a"); printf("distance=%d",total\_distance);

}

**Problem -2:**

Q.2 If a teacher is being served at the food mess and during the period when he is being served, another teacher comes, then that teacher would get the service (food) next. This process might continue leading to increase in waiting time of students to get food. Ensure in your program that the waiting time of students is minimized

# Explanation

1.To see the waiting and burst time.

2.To reduce waiting time.

**Code Snippet**

#include(stdio.h) int main()

{

int arrival\_time[10], burst\_time[10], temp[10]; int i, smallest, count = 0, time, limit; double wait\_time = 0, turnaround\_time = 0, end; float average\_waiting\_time, average\_turnaround\_time; printf("\nEnter the Total Number of Processes:\t"); scanf("%d", &limit); printf("\nEnter Details of %d Processes\n", limit); for(i = 0; i < limit; i++)

{

printf("\nEnter Arrival Time:\t"); scanf("%d", &arrival\_time[i]); printf("Enter Burst Time:\t"); scanf("%d", &burst\_time[i]); temp[i] = burst\_time[i];

}

burst\_time[9] = 9999; for(time = 0; count != limit; time++)

{

smallest = 9; for(i = 0; i < limit; i++)

{

if(arrival\_time[i] <= time && burst\_time[i] < burst\_time[smallest] && burst\_time[i] > 0)

{ smallest = i;

} }

return 0;