

## LABORATORY WORK BOOK

Name of the Student RACHERLA SANTHOSH								Roll Number								
		T-B Semester	and the second s				3	9		ı A			G	3		
Nam	e of the	Course Faculty MY• N •	Raghai	va Ra	0				.Facu							
	1			( variopi <sub>2</sub>	(S AWARDED											
	Exercise Number	EXERCISE NAME	Aim/	Algorithm /	Source Code Calculations and Graphs		_	Program Execution Results and Error Analysis		-	Viva - , Voce		Total			
			Preparation	Performance in the Lab			1									
			4				4		4			4	1	20		
1	8.1	The Disk Access Dilemma	4		4		ب	1	VE	S PO	1	4	2	0		
2	8.2	The SSTF DISK Scheduling Challenge	'		01 =	0	ad i	49	a.	de	1	in				
3	8.3	Future Tech Corporation	Cons	005	008	50	. 000	1				,01				
4	8.4	THE C-SCAN DISK SCHEDULING ODYSSEY	10.0	o o	- cou											
5	8.5	The c-scan Disk SHEDULING QUEST.	4114	grown	Short	- 0	OL X		-			•				
6		. ,		0= 5	evene		- 10	9	1.	103	de.					
7		10	Postu	_001	indi =	1/18	ties	-	- 1	1310						
8			ortion,	P. loss	10°2 =	cell	No.	-	40	-15	bito	3				
9				: 0	capolas		1/6	1	an'a	YE	1	•				
10			610			10			mars	Volt						
111		Trables of														
12	2	Jinsmi	enten	2 th 1	12 (110)		- 0									

Banthah

Signature of the Student

Signature of the Faculty

8. Dink Scheduling.

8.1

The Disk Access Dilemma:-

AIM: - Write a Program for the Lisk access Lilemma which operates under the First - Gome, First Served (FGFS).

PROGRAM :-

initial - position = 150

requests = [200, 50, 800, 300, 100]

def fefs\_disk\_ øscheduling (initial\_position, requests).

The Total Control and

total \_ head \_ movement = 0

current\_position = initial\_ position

Order- of tracks = [initial - Position]

for request in requests:

movement = abs (=werest - position - request)

total-head\_movement += movement

Sworent - Position = Yequest

order- of tracks, append (request)

retion total\_head\_movement, order\_of\_tracks

total\_movement, track\_order = fcfs\_disk\_ scheduling

("initial\_position, requests)

Point ("Total head movement: "total \_ movement)

Point ("Order of tracks visited: "track\_ order)

Output:

Total head movement: 1150

Order of tracks visited: [150, 200, 50, 800, 300, 100]

The SSTF Disk Scheduling Challenge:

AIM: - Write a Program on the goal of SSTF is to reduce the total speck time by always selecting the request closest to the current heart Position. This approach minimizes the movement of the Lisk head between requests.

8.2

```
They total head movements or the
PROGRAM: -
initial - position = 750
requests = [1200, 500, 900, 1500, 300]
def Setf_disk_ Doheduling (initial_position, requests):
  total - head - movement = 0
  = initial_position
  Order of tracks = [initial - position]
 Yemaining_ Yearnests = Yearnests, copy ()
 while remaining - requests:
 closest - request = min (remaining - requests, key =
    Samble x: abs (current - position - x))
  movement = abs (surrent - position - closest - request)
  total - head - movement += movement
 Current_ Position = closest / Yequest
 order - of - tracks, appoind ( Dosest - request)
 remaining_ requests. remove (closest_request)
return total - head - movement, order - of tracks
```

total movement, track order = Dotf - disk - Doheduling (initial position, Yequests) Print ("Total head movement:", total movement) Pount ("Order of tracks Visited:", track-Order). OUTPUT :-Total head movement: 1950 Order of tracks visited: 6750, 900, 1200, 1500, 500,300] +utive Tech Corporation :-AIM: - Write a program on Future Tech Corporation using elevator algorithm. PROGRAM : initial position = 2500 requests = [2800, 1500, 3500, 4000, 1000]

Link \_ Dize = 500

def Dam - disk \_ Dicheduling (initial \_ Position requests, Lisk\_ Size):

```
requests - above = Dorted ( [track for track in requests if
        track > initial_position])
 request below = Dorted [ [track for track in requests if
                 track ~ "nitial . Position]
                     Yeverse = true)
 total_head_ movement = 0
 Order- of - tracks = []
 <ur>current - Position = initial - position
 for track in requests_above:
   movement = abs (= worent - position - track)
   total-head-movement += movement
  = wwent - Position = track
  order- of - tracks. append (track):
return total-head-movement, order fof-tracks
total-movement, trock-order & Scan-Lisk.
  Scheduling (Initial Posttion, Veguests, Lisk-Size)
Pount (" Total head novement:" total_ novement)
Pount (" Order of tracks Visited:"
```

8.4

[Initial position] + track - order) OUTPUT : \_ total head movement: 5498 Order of tracks visited: [2500, 2800, 3500; 4000, 4999, 1500, 1000) The G-SGAN Disk Scheduling Odyssey ? -ATM: - Write a priogram for the C-SCAN Disk Scheduling Odyssey. PROGRAM: return total - head - movement. "initial - position = 4000 requests = [4200, 1000, 6000, 7500, 2000] dok - Size / 10000 Point ("Total hand Movem det = scan = disk\_ scheduling (initial

Position, requests, dlsk\_bize):

Yeavests\_above = Boited ([track for track in
Yeavests of track >= 9 nitial\_ Position])

Veguets- below - Sorted ( L track for track in Veguests if track < initial\_position]) total - head - movement = 0 Order - of \_tracks = [] = wrent \_ position = initial \_ position for track in requests - above: movement = abs (surrent\_ Position \_ track) total \_ head \_ movement += movement: = wovents - position = track order- of\_ tracks. append (track) return total-head-movement, order- of-tracks total\_movement, track-order = = - Dean\_disk\_ Dicheduling (initial\_position, requests, Jisk\_size) Pount ("Total head Movement: ", total\_ movement) Point ("Order of tracks Visited:" [initial\_ Position] + track\_ Order] the second of the second of

OUTPUT :-

Total head movement: 13999

Order of tracks Visited: [4000, 4200, 6000,

7500, 9999, 0, 1000, 20007

The C-SCAN Dish Scheduling Puest at Tech Fusion Labs :-

AIM: - Moute a Program for Techfusion Labs on the C-SCAN Disk Scheduling Juest.

PROGRAM: -

initial = Position = 3500

Yequests = [3800, 600, 7000, 1500, 2500)

Just \_ dize = 8000

def = Dean\_disk - Deheduling (initial\_ Position, reasuests, dlsk - Dize):

8.5

requests - above = Borted ( [ track for buck un requests if track >= initial - Position J.) requests - bolow = Dorted ([track for track in raquests if track < initial\_position]) total - head - movement = 0 order-of-tracks = [] = werent \_ Position = initial \_ Position for track in requests \_ above: movement = abs ( current - position - track) total\_head\_movement += movement = wount \_ Position = track order - of \_ tracks. append (track) return total-head\_movement, oroler-of-tracks total\_movement, tracky/Order = <\_ Scan\_ Lisk - Scheduling (initial - position, requests, disk s Dize) Leaghing remission of the collect

Pount ("Total head movement:", total\_movement)

Pount ("Order of tracks visited:", [initial\_position]

+ track\_ order).

OUTPUT : -

Total head movement: 14998

Order of tracks visited: [3500, 3800, 6000, 7000, 7999, 0, 600, 1500, 2500]

8/11/8/