

## LABORATORY WORK BOOK

Name of the Student: N' Kavi Chanculla							Roll Number							
Cla	ss	C5D-B Semester.	helli	sement	<u> </u>					, 6	7.	В	3	
Cou	urse Coc	le: ACSDIO Course	e Name :0	is labor	itory	3	39		1	-	-			
		Course Faculty M5.0				******		Fa	culty ID	) <u></u> I	ARE	109	JI.	
		mber. 2			ຼ ລູ	2.4		•	ate :	xel!	091:	24	nanj-ti	
	Exercise Number	EXERCISE NAME			KS AWARDED									
S. No.			Aim/ Preparation	Algorithm / Procedure Performance in the Lab			ce Code	ions Results and Error					Total	
							ulations Graphs			TOF.		2	10.55	
			4	4			4		4		4	WED-COM	20	
1	<b>a</b> ∙1	Managing system & User pareses	Ti de la c	-47	ius o	94	j sa r		- 1					
			0	. 2	2		4		4	-	4	+	7 (	
2	9.9	managing pobscheduling				into t	<b>.</b>		i in p	- Maria	1.			
		in a computing certific	i ey	2	.2_		4	4.7	4	-	4	14	ب	
3	5.3	Managing pant Jobs	,	9	9		Ls -		6		4	17	20	
4	9.4	Task scheduling ina	.4				1				i	No.		
		multi-use system	G	9	2 -1	1-2	1.	ħ.	4		t	1	21	
5	<b>3-</b> 5	Job scheduling in a	الرواق الا	1、技服的		200		lo si	 		1	1-		
		computing duster	4	2	2		4		4		4	1	1	
6		198			sm)	= <del>}-</del>	tja alf	1		-	,	NAME OF TAXABLE PARTY.		
7		The state of the s	-50	1		¥ 75	ji h.	والمعار	đ z		11.5	- Charles		
y. 1			531 7	Jan San San San San San San San San San S		_				_		1	4	
8	4.5	4.50	- 2			1	.CNi	COLT.			11			
9		Caroli (Francisco)	1. x 3r _ #	JE E	2 245/11	J. F	-	. 3.			L L	d.		
10		And the state of t	Filter.	J. DI WILLIA	10711	al ru					is ph	- Company		
11					11-12-13	1-1	Y * 7	- 3	3.1*		177	T		
			9 37	Salt F										
12						<b>H</b> . I		إينا	 	-				
													_	

N · Ravi chandrika Signature of the Student

Signature of the Faculty

## START WRITING FROM HERE

2.1

Alm: You are managing a computer system that runs various types of processes, categorized into system processes and uses processes. System processes critical for functioning of 0s and are given higher policialty compared to uses processes.

 $\mathfrak{D} \cdot \mathfrak{I}$ 

## Code:

from queue impart Queue

aid calculate-total-time ( system-processes, uses-processes):

process-queue = Queue()

too polocess, time in system-polocesses. Flems():

priocess - queue · put ((priocess, time))

for process, time in user-processes. items():

priocess-queue.put ((priocess, time))

total-time = 0

while not proces-queue-empty ():

process, time = porocess-queue.getc)

total-time +=time

return total-time

System-priocesses = { Priocess A': 5, 'Poiocess B': 3, 'Priocess C':1}

Wes-priocesses = { Priocess O': 4, 'Priocess E': 2, 'Priocess F': 6'}

total-time = calculate-total-time (System-priocesses, USCS-processes)

print(t' The total time required to complete all priocesses is:

{ total-time? Units"}.

output: The total time oregared to complete all processes is:

2.3 Aim: You are managing a computing centre that priocesses jobs submitted by various departments of an organization. Jobs dre categorized into two possessity levels: high possessity and the low · philosity. Code: tion qu'eue Empart Queue dy calculate-total-time (high-paiarity-jobs, law-parasity-jobs): job-queue = Queuec) for job, time in high-policity-jobs-items(): · Job-queue. put ((job, time)) tou job, time in low-policosity-jobs-items(): job-queue put ((job, time)) total-time = 9 ityru: susud-dot too stidu while not job-queue empty() Job, time = job-queue.get() total-time += time Idea William high-poilouity-jobs = ? Job R : 8, Job B': 5, Job C': 103 low-periodity-jobs= & Job P': 6, 'Job E':3, 'Job F': 7} total-time = calculate-total-time ( high-priority-jobs, low-priority-jobs print[f" The total time vicquired to complete all jobs is: Stotal-time & units Output: The total time required to complete all Jobs W: 39 units.

2.3 Aim: You are managing a shared printing environment where the posint jobs differ from depeatment to department. Posint jobs one categorized into high priority & Low priority.

Code:

from queue Proposit Queue

dy calculate-total-time ( high-policialty-jobs, low-policialty-jobs):

job-queue = Queuec) for job, time in high-princity-jobs. Items():

job, queue. put (Cjob, time))

tou job, time in Low-pariority-jobs. itemses: job-queue-put((job, time))

total-time = 0

while not job-Queue. empty():

job, time = job-queue.qu()

total-time += time

return tokal-time

high priority-jobs = & Job A'+15, 'Job B': 10, 'Job c': 20%

low-priority-jobs = & Job D':5, Job E':8, Job F':3}

· fotal-time = calculate total-time Chigh-paioxity-Jobs, low\_priority Jobs)

Polint (1" The total terme required to complete all jobs is: & total-time? units")

The state of the pro-Output: The total time required to complete all print jobs is:

61 Units.

**ROLL NUMBER:** 2.4 Aim: You are managing a multi-use system where tasks from different were need to be exhectuted for paracessing. Tacks are categorized into two priority levels: high priority and Low pelicolity. Code: from queue Empart Queue dy calculate-total-time (high-policially-jobs, low-policially-jobs): task-queue = Queuec) for task, time in high-priority-lasks. items(): task-queue. put (Ctask, time)) for task, time in Law-parasity-tasks. Hemes): task-queue. put ((task, time)) total-time=0 യ - 1<sup>ൽ ക</sup>ംഗിയും പാഗി ന് യ<del>േൽ</del> ഉദ്വി. രാർ while not task-queue. cmpty(); -, time = task-quelle. getc) total-time += time Low-priority-jobs= & Task D': 6, Task E': 4, Task F': 10} total-time = calculate-total-time. Chigh-priority-gobs, law-priority print It the time required to complete au sobs is: étotal-tim and it is the second of the second Output: The total time required to complete all jobs is: - 1 ( ) 1 - Williams = 1 ( ) 1 - 4 ( ) 1 William So units.

The American warden or combigued crimes and broggers love submitted by exportaments of a reason shoulder of folia its 3.4 exagentized that a light parently massen parently of the down Dagarille. Courtes Am queue Simpaul Burne dy calculate - total -trace ( high- priority-jobs, meetion, locals Joh-Quepe Bueuces to job, time in high-policonity-jobe. Tempero: Job-queur . put ((Job, (ime)) for Job, time in mectium-policetity-jobs. items(); 306-queue. put ((Jobstime)) : c semals " adol - Uniconal - was of smit, doly rest job-queue-put ((job, time)). total-time = 0 While not gob-queue. empty (); -, time = job-queue, getcs total-time + = time , return total-time High-parooraty-Jobs = { '30b A' : 30, 'Job B' : 15:1506 c' : 253 dedium-paravity-jobs: { 'Jobo': 10', 'Jobe': 12, 'Job F': 8?, law-poilagety-jobs= 2'Job G1': 5, 'JobH': 4, 'Job P': 62 total-time = calculate -total-time Chigh-pariority-gobs, median-priority-gobs, law-priority-gobs)

## ROLL NUMBER :

possot (4" The total 4 Fine sequired to complete all the tosler.

i. I statel - (7mg/mil time)

Output: The total time agained to complete an the tasks is: