OS Lab

1. Matrix Operations:

(1)	Mother operations: Breghom to hultiply, Add, Subtract, Transpok,
	Symmetry and diagonal sum.
	# include (stoloh).
	Void make Multiple Cotion (int motistivo), int moto [36w), int mult [7[100].
	; not horefalt, in Coleons first, introviscions, lost Colum selond);
	hord meller Addition (int ma 157 (100), int mas (7 (100), ind odd (3 (100), int how, int column
	void motios subtraction (wt mot 107 100), int mod (7(100), int sub (3(100), int row, int column)
	Void hother substitution (in Int (1007, in thou, int toum);
	Void mother Symmetry (in mot [][100], int how, int column),
10000000	Void Indres Diagond Sum (int mot[7 500], int trau, int column):
	ind mountil
	in hat [[10][10], hat a (100][100], hall [100][100], odd (100] (100] (100] (100), thousand of (100)
	thons 2 [100] [Now fist, Column first, how second, column Selond;
	plot (" Tulo 4. number of hough Column of fish butters. wil;
	sconf (" 1. d 4 d", Prought, delanifiset),
	point ["unti plements of hollin 1").
	for 11:0; 12 rowfst.; i++)
	la 1 j = 0; j clolem fort i j H)
	560rf / 4.24, & moti (17857).
	phonel (" Buts the number of Good La Column in method ("1),
	sanf 1" 1.21.24, I howselow, & column selond),
	print (vonte the abovert of motries 24).
	por (rat 1 = 0; 1 < n, 14)
	Br (intisorien; jtt)
	Sanf (" 4.3", & moto (i? (j?)) inapt
	mothix Multiplication (most, ind), mille, colon first, howevers, Columbocards
	plointf (" Repet of not rin miltipl' Cotion In");
	for (on : =0, icn; (++) f
	for (int 5:0; if on 1 1th)
	print ("4.d", mell (;)(j?).
	phull ("Ina),
	3
	protein such

	The state of the s
	matrix Addition (mot!, but 2, and, hourfist, colum fiss).
	print (" Robert of Adolition 1/1/2)
	for lont 1:0:1 <n;1+1)< td=""></n;1+1)<>
100	for / in/i = 0: i (n; (++)
λ.	ost phint (1' 1.24, and (1677),
to y	print (41 n4),
	3
	La Sec Sec. Acres Nov. 1 1 1 1 2 2
	mothin substraction (mot 1, not 2, sub, howfist, when first)
	print 1" Reled of Subtraction: Int),
1	for (int 1 = 0; icn; i+1) {
-3	for (intj=v;jen;j+1)&
	phont 11 4.24, SUB (.7(4);
	phont (4 mas),
_	1
-	mothis Thousand (mst), thouse, how First, lakem first).
	mothis Trompoh (met 2, Hons 2, how Better Welen fot Selve)
71	front (" Throng pole of hobers!: \n")
	for lent ? =0; 1 cn; 1+1) !
	BA (int y 0, j(n, j ti) pront (" 1. 2", bron (:7[]).
	of phote ("W.d. tombette
	2
	phrall (" Troughot of bothing: well)
	for finhing, 12n, 141) {
	for Controvienini
555	
	phontf (" 1.34, -Nongo [.7, j])
	print (" 1, 2, -1/2015) [.7, g]).

	Void motriplishelp of Cint mot Groon, ant phone (35100) indrew
	In (olum) (
	An () 1 20 1 ic how 1 +1)
	12 (int 3 =0) 1 < Column; 1 /t)
	Hypris Cy7Ci7 = bret Ci7Cj7
)
	poid motrix Symmetry (and mot [] Two], inthou, m Colom)
11.	fr (inti so, ic how, "tt)
C1 .	los (int 1:0 : 1 = column: i++) 1
	for (int joo; je column; j++) 1 if (mot (i7 Cj3 1 - h) Cj2(i) !
	flagso;
	1 Hole;
	3 ?
	if (floy 2 = 0)
	briok.
	7
	of (floy > 1)
	Wintf (" Signatual"),
	clu
	plostf (" not symmetris "),
	3
	Noid motio Digond Sing (and mod (7 1007, int trow, int column)
	forther int phonopolsam-o, non-principal sum =0,
/	for (intion; icrow, it)
	for (intjou; je colum, j'H)
	2/ (; ±=;)
	phon wpd Siem + 5 met (i) (A)
	(iti = 9001-1)
	2 Prin apol Sum + 5 mt (:) (4):
	phint (" Sum of phon C. pol ab eggod of d'in, phone pol Sum)

11.1	output:
	into no of hous and columns of 15th holys
	3 3
	into the element of mont VID'
	1 3 2
- Ann	45 or 6
1 8	8 4 3
	Is No no of how & colum of so Cond hutting
	3 3
	onto elements of foton hortring:
	2 12 11
	5 6 16
	7 4 1
	Repult of motrix odelilar:
	3 5 13
	Γο 1122
	1 8 4
	Repelt of notre buttle for colien !
	31 39 61
	117 (94 (81
	r7 132 /rr
	40 15
	motrix (is not symmetric motrix 2 is not symmetric
	Mollips 2 (1) ver syrice to
	Thought of hating!
	nur8
	3 7 4
- 11	ð 63
	Cum of Non plantipel diogend is 16
	in of Non planapol dioyond is 11

INF	MITY	
Date /	1	1

	Date / /
	molling Symmetry (hot I Iron First, Column First),
	hote's Symmetry (not), how First, Column First);
	me Wille good fam (mot), rowfult, column fill.
	India Diagol Sum (hot), Row Ge Cond, Column G. Co. 1).
	2
	slam o;
	2
	Void mallex Multiplication (int modil 3 lbo7, int mod (7(100), int hold (100),
	in twistilet, int Columnt ut, and how scoul, int colum scool)?
	ent 1, j, k,
	for (1:0; ic swellet, ++i) &
~	for (j > 0; j < (Elina fest; j 1+)
	hull (.7(1)=0,
	for (100; ic sought; ttil 8
	for 1900, if a Colon follow, ith
	for (11:0; Ke Kolem F. St., KIt)
	mult (i7 (j) t= mot (Ci7(u) * hold (K)(j?);
•	3
7	
	Void hothis bld tion (in hot (7 hos), in hoto (760), in odd (7600)
	in how, In Column)
	12 1,1
	for (100; 1 chows, 1++)
	for (520, j'color; j'tt)
	pold (1) (1) = hot 1 (07 (17+ hots G2 (12)
	3
	深 之
	Void Intersultraction (and had 103 (woT, int has 103 (lost in sub (7 lios)
	introve, in column 1
	Br Contino, ichow; itt
. 54	for (ont ; 50; j' < Column; j+1/ sub () (1) = not (() () 7 - not 2 () 7 () ;
	sub Co) (1) = holl Ci)(11- holders (1),

Output:

```
Enter the number of rows and columns of first matrix: 3 3
Enter the elements of first matrix:
1 3 2
45 5 6
8 4 3
Enter the number of rows and columns of second matrix: 3 3
Enter the elements of second matrix:
2 12 11
5 6 16
7 4 1
Result of matrix multiplication:
31 38 61
157 594 581
57 132 155
Result of matrix addition:
3 15 13
50 11 22
15 8 4
Result of matrix subtraction:
-1 -9 -9
40 -1 -10
1 0 2
Transpose of first matrix:
1 45 8
3 5 4
263
Transpose of second matrix:
2 5 7
12 6 4
11 16 1
The matrix is not symmetrical.
The matrix is not symmetrical.
Sum of principal diagonal: 9
Sum of non-principal diagonal: 15
Sum of principal diagonal: 9
Sum of non-principal diagonal: 24
PS D:\Programs\OSlab>
```

2. Write a C program to simulate the following non-pre-emptive CPU scheduling algorithm to find turnaround time and waiting time. i. FCFS ii. SJF (pre-emptive & non-pre-emptive)

	2061
	let non-preomptus CPV Shedeling
	FCES olgorithm to find hernoround time and westing time
-1	See Se
9)	# include < stdio.h)
	Void weit-time (int n, int basst time [], int weiting-time[]); Void togradeund-times (int n, int basst time [], int weiting-time[]
	wind transferred times (int n int barst time I, int waiting time I)
	and the same actions and the same and the sa
	Void orgtime (int n, int proc [], int burst time []).
	vee algunt (in)
	int moin () (
	int prec [7 = {1,3,3};
	int built time E7 = {12, 3, 42;
	intn;
	plint (" onter the number of processes (");
	Story (" 1. 1", ln).
	proof ("Enter the processes In");
	Storf t"
	for (inti=0;i <n;i+1) &<="" th=""></n;i+1)>
	56mf(" 1.d", & proc (:7);
	print (" onto the burst times In");
,	folin 1:0; i< n; 1+1
	sconf 10 1 dt, l burst time[:7);
1	ovgtime (n, ploc, bust time).
	return 0;
	3
	Void wort time (int n. it set is
	Void wort time (int n, int built time [7, int weiling tous]
	inti;
	for (i=0; i <n; i++)<="" td=""></n;>
	whitno time Fize
	Sum + = buyltin (:]:
	{ = 5004-6m (:];

	writing time 1:7.
	worting time [i7 = sum;
	Vord Thompsound to call do 11 8 ft to 57 8 to to to 5
	int lumoround_times (int n, int burst time [7, int worting_time[7]
	for (int i=0; i <n; i++)="" td="" {<=""></n;>
	twomphound - Time [i] = b whit time [i] + waiting-time [i];
	3
	3
	management of the second of th
	Void org time (int n, int proc [], int burst time (])
1	ent worting-time Ing.
1	int turn fround time [n];
1	floot ovg-woil, ovg-tat,
1	wort time (n 1, get to with to).
1	turn downd times (n, bull time, working time, tromstand times
1	in sum wat = 0;
1	int sum tot = 0;
	for (int i=0; i < n; i+1){
1	Sum - woit += woiting_time[i];
1	Sum- tol + = two noround-time Cit;
1	3
1	aug_tot = (floot) sum_wort ln;
1	org woit = (floot) sum-worth;
	frint (" brokes (+ Burst time It Worting time to Turnolound time (")
1	for (int i 50; ien; i+1) &
*	phints ("1. dltlb 1-dltlt 1. J tt & ", procco7, buretime
1	walking time (17, two notwed (ino (-7);
1	plint (" In Average woiling time it: tof", ang-time!");
#	phind (" In Average trans-dound line 11:15 ", ovg-tat),
#	4
	Till the state of

Out the number of products 3 3 Enter the process 1 2 3 12 Process Burst time Worting the Turn abound time 1 4 0 4 7 7 19 Average working time is: 3.666667 Average term pround time is: 10.00					-//
Anter the processes Frotes the bearst time 1	output:	2.81	net Volumbar	4	
Anter the processes Frotes the bearst time 1	Onto the	umped of pro	esses		
Enter the prolegges Enter the beingt time 1	4 3			the second stand	
Enter the besit time 17 3 12. Process Burst time Indesting time Turn around time 1 4 0 4 2 3 4 7 3 12 7 19 Avarage wasting time is: 3.666667		processes	ili e de la compania	13	
Enter the biest time 12 3 12. Protes Burst time Indesting time Turn abound time 1 4 0 4 2 3 4 7 3 12 7 19 Avaroge worting time is: 3.666667			1.5		
Profess Burst time Woiting time Turn or ound time 1			ali A. S. S.		
Proless Burst time Working tiles Turn oround time 1 4 0 4 2 3 4 7 3 12 7 19 Avarage woiting time is: 3.666667			g = 0		
2 3 4 7 3 12 7 19 Avarogs woiting time is: 3.666667	4 3	72.			
2 3 4 7 3 12 7 19 Avarogs woiting time is: 3.666667	Proless	Burst time	Worting time	Turn oro	and time
Avaroge woiting time is: 3.666667	1,	4	0	4	
Avaroge woiting time is: 3.666667	2	3	4	7	1
Avaroge woiting time is: 3.666667	3	12	7	19	
Average tem shound time is: 10.00		2.	1 × 20 1	4()	1 13
Average term dround time is: 10.00		- 4			0
Average telm drown time is: 10.00	Avarone	white to ?	2 ////2		
Mostage nom placere lime is: 10.00	Auston	the sale of	S. 66666 f		
	nveloge	com scoure (a)	ne 15: 10.00		
			- Ja . a . c		
		:1:	- M. S. L. L	1 4	
			-12-1		
		1	1.1		
			and the same	e April	
the state of the s		1	Landa Tan		
		1 5 1 1	Alexander of the rest		-51

	to the int bull time (3 int.	/ >
	Void tumoround-times (int n int bull time (3, int wo	n) lin
	int turn ground-time [])	1
_	for (int 100, icn; 141) =	-
7-11	for (int is time til = burghtime (i) + Uxorting-time!	13:
	2	
	void sort (ind n, int proc (3, in part time (7) (and toh	p .
	for (intiso; icn; itt) <	
	Los (in) i=1; i(n; i++)	
	if (busyt-line [i] > haret time [i])	
	temp = burst. line (i);	
	Swylling [] = Swylling [];	
	burst.ton (175 temp:	
	protomp = proc(j);	ř
	pro((i) = pro((:7)	
	phot Ci7 = total;	
	2	
	?	
	3	
	2	
	void orgline (int n, int proc (7, in burst-time (7)	
	But woiting time (n);	
	int turn dround - time [n]	
	floor ong-woit, ovy-tot;	
	won't-time (n, bulst time; won't my-time).	
	dun orande timed (n, burytime, writing time, tunde	
	int Sum- wort-0;	-
	Int sum-tot=0;	_
	for (int 1=0; icn, i++)	_
	Sum Bort + 3 wenting - tim (. 7.	_
	for (:nt:=0; : <n; :+1)<="" td=""><td>_</td></n;>	_
	Sum_tot from + = toumplourd_time C.].	_

-				
	009-61	5 (floot) Su	m-tol/n/	
-			m- woll/n;	
				my t Jum Franktim ("),
	for ()	nt 100; Kn; 1		
		froll (ut.	114.9147.3147	of miprot (:), burt time (
			hu (3), tomoround-to	
	plunts (" overage time 1	sorting time : # f", o	vy-with
	frint/	overege how	oround-time: 1. f 4	ovy-tot);
	3			100 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
	net put:			
	Enter nu	of protesses		H H
	3			
	unto the	processis		4.4
6.24	1 2	3		
	inter the	burst times		
	4	3 12		
	Proless	Burd time	Wer'ting time	Turn dround time
	2	3	ó	3
	1	4	3	7
	3	10.	7	19.
	/			1
	Andron	on the time	: 3.333	
	Aug to	and to	my; 9,666	
1	. Tologe M	om swing he	1, 000	The state of the s
1	13	upload	one of	
22/9/		Sigle	(men)	
		1 hen	u d'a	1
	- 2			
io sa II				

Output:

```
Menu:

    First-Come, First-Served (FCFS)

2. Shortest Job First (SJF)
Enter your choice: 1
Enter the number of processes: 3
Enter burst time for each process:
P[1]: 4
P[2]: 3
P[3]: 12
Process Burst Time
                          Waiting Time
                                           Turnaround Time
P[1]
P[2]
P[3]
                                   0
                                                    19
                 12
                                   7
Average Waiting Time = 3.666667
Average Turnaround Time = 10.000000

    First-Come, First-Served (FCFS)

2. Shortest Job First (SJF)
Enter your choice: 2
Enter number of processes: 3
Enter burst time for each process:
P[1]: 4
P[2]: 3
P[3]: 12
                          Waiting Time
                                            Turnaround Time
Process Burst Time
                                   0
P[2]
P[1]
                 4
P[3]
                 12
                                                    19
Average Waiting Time = 3.333333
Average Turnaround Time = 9.666667
```

3. Write a C program to simulate the following CPU scheduling algorithm to find turnaround time and waiting time. A) Priority (pre-emptive & non-pre-emptive) B) Round Robin (Experiment with different quantum sizes for RR algorithm)

	Do.	INFINITY Date
	Lob 2.	
28-06-23	E 60 C	
	Write a Chroghom to & hulate the following on Sheduling algorithm to find two maround time on a) Priority	CPU
٥.	White a Chrogram to much I begin shound time on	d writing to
	Sheduling objection to fine war	Jim
15	a) Priorty	
	6) Round Robin	
so Va	The state of the s	
8	# include < stdinks	
	# include <stdbb,h)< th=""><th>- 10 P. 20</th></stdbb,h)<>	- 10 P. 20
	void suop (int a , int b); {	
	void praisty-algorith ();	The state of the s
	Void fround robin ();	
	V 0. 0 2 3 618 , ,	
	int moin () (
2	int choile;	
	while (1) {	and the same of
	phint (" onter I for phiduty Algorithm Into for	Round Robin Wilt
	3 for ovi (Nil).	
	Store ("+ d" Achoric);	
	Switch (chork) (
	Cosk 1: phrolity olystithm ().	
	brick;	
	Cosed; Bound Probin ().	
	black;	
	(ch3: epit(o);	
	defruit!	
	pronty ("Invold choice (""),	
	7	10 19 19
	}	
	Setwin O;	
	}	
	Void phodoly- objection ()	
	int niifi	
	Void phidity- olydrothen () { int n, i, i: fried ("Winter the number of photoless:");	
		1000



=	Date / /
	Scenf ("4. 19, 8n),
1 1	int burst lim (n?, phoch, ty [n], proless-id[n],
da V	The state of the s
3 - 1-1	for (i=0, icn, i++) {
1994 - 4	phint (" only the burst time and phiolity for protess 11;" 1+
	phint (" only the burst time and phiolity for proless 13;" 1+1 Sconf (" 1 & 4d", & burst time [i], pho obity [i];
	protes _ id [:7:i+1;
	2
	7
	for (i=0, icn-1; itt) {
	for (j = [+1; j < n; j t) {
	if (photity [1] > photity (1))
	5 wop (& pr. dis (17, 1pr du) (17);
	Swop (L butstlim C. 7, & burst line ();
	Swopl & prounid (; 7, & protest id (; 1);
	2
	7
	int working time [n7, twom drawned-time [n7, total-working lim =0,
	told fulm down time = 0;
	for (int 1: 0; ien ; i+1){
	working time (i) = total working time;
	told-waiting-time + = burst time[:]:
	4
	for (;nt; =0; ;cn;;++){
	ton dround Itim (: 75 worling-time (5) + Least-time (1);
	total- turnshound time + 5 turn abound - time [il;
/	print (" Process 10) & Burst tim + worting tim 1+ Turnoround Tim \10)
	for a =0, 1 < n, 1+0 &
	plints (" 4.d (t) t 1.d (t
	for a =0, ien, i+0 & plints (" +d (t) t +d (t)
	tury pround lime (: ?):

	Date / /
	1) Was hold water to
	ph of (" Aveloge working time of In Iflow) fold working time fol.
	printy 1" Auring born about true of the floor some summound line
	st I (" Told worting time! tod w, letter tom ling time!
	prost (" Average working land time 4 (m", (flood) lotal ternoround time prost (" Told working time " to d' ", teled - working time). John of (" Told telenoround lime 42 m", told talm dround time)
	2
	void hound-hobin () (
	let a alcontinutine in 11
(a. 1)	powell ("Honto the number of probe (601 : 14").
	Sonf ("1,29 (n),
	307
	and built tom (n?, Surrowney ting (n), worting time (n),
	turnoround line (27;
	The state of the s
	for (int i=0; icn; i+1){
	print ("6 nter berst, time for probes 1. 1:", i+1).
	scorf ("1. 2", Rbuist time (it):
	remorning - time [i] = built - time [i];
	3
	prints (" Enter time quanteur!"),
	Scanf (" 1 d". Lywontien,).
	/ int time =0, done =0;
	while (done 1 = n) {
	for (on: :0; icn; itt)
	if (Semoiny tin (:) Solf
1	If larning to [:] > q16ortun) ?
1	the t= quentury;
	Jamoing-lime - quantan;
t.	elses
	They + = Semaincy time [i]:
	i will I !



	Date / / /
worting time (i) = time - bust-like (i)	or harman
Generaling - line (5/50)	
olone 1+	
3	11
2	. 17
2	1
7	
and total west to = a. Total tuberdand to	Lung 50,
flood total working time = 0, total turnoround to	
for (i=o; i <n; <="" i+1)="" td=""><td>.7,</td></n;>	.7,
total-working-time + = working-time [total-termoround-ton + = larn drand	t [:1:
wild turnordurd ton to turn decine	ans cry
(
prot (" Protes ID to Burst 7:m to Worling Time	, (c jameean)
for (;=0; i <n; &<="" i+)="" td=""><td>1 1 1 .91 + . 5.7:</td></n;>	1 1 1 .91 + . 5.7:
print (" 1. 1 th t. 1. 1 th t. 1. 1 th t. 1. 2 th t. 2.	- C.7).
woiting time (it, turnglound-to	my (1),
print ("Apoline won try line it of With	11 1 H to faultin ha)
fit set (" Average him drexued time! 4. f (") (fe	(00) FOR UMOROUGE LINE / W/
	₄ + 1
print 1" Total worling time : 1.0 w, told-world	eng am 1,1
phints (" Total worling time: V.d' wishold-world phints (" Total humoround time: V.d (" to	to sum oldered - time !!
, 0	
3	
Troling of phiority algorithm	
307	
Inhed Proless pholoty burst line.	
Po 3 12	
P ₁ 2 5	4 (/ 81)
P2 1 4	from Gutt Charl
Sollie ollody to phistily. Woi	try time tim oround trong
P3 1 4 1	9
P1 3 5 4	21
Po 3 12. 1	« /

	· Ave/9	
1 1 1	Grantt Chal:	
	P2 P Pa	
	0 4 1	
	Troling for round robin Meduling:	
	Inbut proles burst time Worting time	turndrom
	Input proles bus wing	- 9
	ρ. 3	9
	ρ 5	13
	12	
	tim Quantum/shel = 2.	
(b	time buentum/sled - d.	1 7 4 3
1 12 1	Contl chat:	
	Po P, Po P, Po	Pa
- 1	0 3 4 6 8 9 11	(3)
		1 1 1 1 1 1
100	s. 11. 1:	11 51
	out put:	1 18
¥	Cto 1 B3 bb det de so H	
	Enter I for producty dayor than	
	2 for Round Robin 3 for o'ryt	- 1
	> for clift	
	tates the number of probables: 3	
	Extra built true and bladt last	Prince
	Enter bush time and photoly for process 1: 12 3	
	Enter built time one best to the	13.50
	Entr buist time one proprity for process; 41 Process ID Buist the Dunt	1 1 1/100
4,	Proces ID Burst time Provity Walting time	Tu VnaVoral Tita
	2 5 2 1	<u> 4</u>
	1 10 3 9	91
- 4	P. Committee of the com	01
(Average turn orward time: 11.333 333	1 50



				CO INFINITY		
	Total wat	t: / 2		Date / /		
	Total worting	une: 15				
	To Economy On	a wine 134				
	onter 1 for ph	or to olgor the				
	2 for Round Robin					
	3 for 6 h I T					
	2					
	Enter number of	prolibel 3				
	Enter buly tim for	r probes 1)	4			
	balls bull ton for	,				
	Ents time quanteen	1.2	N. L. J.	The VIII		
	Pro (ess ID	Burst time	Worting time	Turnaronal time		
	1	<u> </u>		9		
*	3	5	7	12.		
	Average wreiting time	d true : 1.60	16667	1		
	70 tol Usiting time: 17.000000					
	Total turn doing time: 27.000000					
/	100 400					
	Enter 1 for brias	olgor //m				
	2 for how	nd John				
13	3 fot 6 bz					
1 2	3					
23/6/						
						

Output:

```
Enter 1 for Priority Algorithm
        2 for Round Robin
        3 for exit:
 1
Enter the number of processes: 3
Enter burst time and priority for process 1: 12 3
Enter burst time and priority for process 2: 5 2
Enter burst time and priority for process 3: 4 1
                Burst Time Priority
                                                 Waiting Time
                                                                  Turnaround Time
Process ID
                4
                                 1
                                                  0
                                                                  4
2
                5
                                 2
                                                                  9
                                                  4
                12
                                                  9
                                                                  21
Average waiting time: 4.333333
Average turnaround time: 11.333333
Total Waiting time: 13
Total Turnaround time: 34
Enter 1 for Priority Algorithm
        2 for Round Robin
        3 for exit:
 2
Enter the number of processes: 3
Enter burst time for process 1: 4
Enter burst time for process 2: 3
Enter burst time for process 3: 5
Enter time quantum: 2
Process ID
                Burst Time
                                 Waiting Time
                                                 Turnaround Time
1
                4
                                 4
                                                  8
2
                                 6
                                                  9
                                 7
                                                  12
Average waiting time: 5.666667
Average turnaround time: 9.666667
Total Waiting time: 17.000000
Total Turnaround time: 29.000000
Enter 1 for Priority Algorithm
        2 for Round Robin
        3 for exit:
```