VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.



LAB REPORT on

OPERATING SYSTEMS

Submitted by

SUSHMITHA KL(1WA23CS003)

in partial fulfillment for the award of the degree of BACHELOR OF ENGINEERING in COMPUTER SCIENCE AND ENGINEERING



B.M.S. COLLEGE OF ENGINEERING

(Autonomous Institution under VTU)
BENGALURU-560019
Feb-2025 to June-2025

B. M. S. College of Engineering,

Bull Temple Road, Bangalore 560019 (Affiliated To Visvesvaraya Technological University, Belgaum) Department of Computer Science and Engineering



CERTIFICATE

This is to certify that the Lab work entitled "OPERATING SYSTEMS – 23CS4PCOPS" carried out by **SUSHMITHA KL** (**1WA23CS003**), who is Bonafide student of B. M. S. College of Engineering. It is in partial fulfilment for the award of Bachelor of Engineering in Computer Science and Engineering of the Visvesvaraya Technological University, Belgaum during the year Feb 2025-June 2025. The Lab report has been approved as it satisfies the academic requirements in respect of OPERATING SYSTEMS - (23CS4PCOPS) work prescribed for the said degree.

Dr. Seema PatilAssistant Professor
Department of CSE
BMSCE, Bengaluru

Dr. Kavitha SoodaProfessor and Head
Department of CSE
BMSCE, Bengaluru

Index Sheet

Sl. No.	Experiment Title	Page No.
1.	Write a C program to simulate the following non-pre-emptive CPU scheduling algorithm to find turnaround time and waiting time. →FCFS → SJF (pre-emptive & Non-preemptive)	1-19
2.	Write a C program to simulate the following CPU scheduling algorithm to find turnaround time and waiting time. → Priority (pre-emptive & Non-pre-emptive) →Round Robin (Experiment with different quantum sizes for RR algorithm)	20-37
3.	Write a C program to simulate multi-level queue scheduling algorithm considering the following scenario. All the processes in the system are divided into two categories – system processes and user processes. System processes are to be given higher priority than user processes. Use FCFS scheduling for the processes in each queue.	38-45
4.	Write a C program to simulate Real-Time CPU Scheduling algorithms: a) Rate- Monotonic b) Earliest-deadline First	46-58
5.	Write a C program to simulate producer-consumer problem using semaphores	59-63
6.	Write a C program to simulate the concept of Dining Philosophers problem.	64-70
7.	Write a C program to simulate Bankers algorithm for the purpose of deadlock avoidance.	71-74
8.	Write a C program to simulate deadlock detection	75-81
9.	Write a C program to simulate the following contiguous memory allocation techniques a) Worst-fit b) Best-fit c) First-fit	82-88

10.	Write a C program to simulate page replacement algorithms a) FIFO b) LRU c) Optimal	89-100
-----	---	--------

Course Outcomes

C01	Apply the different concepts and functionalities of Operating System	
C02	Analyse various Operating system strategies and techniques	
C03	Demonstrate the different functionalities of Operating System.	
C04	Conduct practical experiments to implement the functionalities of Operating system.	

S. No.	Date	Title	Page No.	Sign / Remarks
1.	6 3 26	FCFS)	2001	1
1.	613/25	STF 9-	03	63
2 .	13 3 25	Priority Round Robin		1
3.	3 4 25	Multilevel 10	14	13/5
		Rate Monotonic. 10	19	0,019
		Earliest Deadline J	22	200
4.	16/4/26	Producer bensumery	26	^
	Table?	Dining Philosopher	29	3/4
5.	1-14/25	Deadlock Detection 7	33	1
		Deadlock dvoidance	37	15/5
6.	815125	Memory Allocation -10	41	Orla
4.	16/6/26	FIFO, Optimal, LRU to	45	Utole
				paget
		notity		
		1		

1)Write a C program to simulate the following non-pre-emptive CPU scheduling algorithm to find turnaround time and waiting time.

*FCFS:

```
#include<stdio.h>
 void CT(int processes[], int n, int bt[], int at[], int ct[]) {
   ct[0] = at[0] + bt[0];
   for (int i = 1; i < n; i++) {
      ct[i] = (ct[i-1] > at[i] ? ct[i-1] : at[i]) + bt[i];
    }
 }
 void TAT(int processes[], int n, int bt[], int at[], int ct[], int tat[]) {
   for (int i = 0; i < n; i++) {
      tat[i] = ct[i] - at[i];
    }
 }
 void WT(int processes[], int n, int bt[], int at[], int tat[], int wt[]) {
   for (int i = 0; i < n; i++) {
      wt[i] = tat[i] - bt[i];
}
 void AVG(int processes[], int n, int bt[], int at[]) {
   int wt[n], tat[n], ct[n];
```

```
CT(processes, n, bt, at, ct);
  TAT(processes, n, bt, at, ct, tat);
  WT(processes, n, bt, at, tat, wt);
  int total_wt = 0, total_tat = 0;
  for (int i = 0; i < n; i++) {
     total_wt += wt[i];
     total_tat += tat[i];
  }
  printf("\navg bt= %.2f", (float)total_wt / n);
  printf("\navg tat = %.2f", (float)total_tat / n);
}
int main() {
  int n;
  printf("no. of processes: ");
  scanf("%d", &n);
  int processes[n], bt[n], at[n];
     printf("enter bt and tat: n"); for (int i = 0; i < n; i++) { printf("bt %d: ", i+1);
     scanf("%d", &bt[i]);
     printf("tat %d: ", i + 1);
     scanf("%d", &at[i]);
     processes[i] = i + 1;
   }
```

```
AVG(processes, n, bt, at);
return 0;
}
```

```
no. of processes: 4
enter bt and tat:
bt 1: 7
tat 1: 0
bt 2: 3
tat 2: 0
bt 3: 4
tat 3: 0
bt 4: 6
tat 4: 0
avg bt= 7.75
avg tat = 12.75
```



FCFS o muses has dear contract, and that cop-#indude < stdoo hs # enclude estalib hy int main (){ int size; print ("Enter no. of proces "); scanf ("1.d". 4x(xe); Dend the many prom : Lit forty int arrival (size): ent burst [sixe]; int TAT [SIZE]; INTERNAL PROPERTY int mail [sixe]; may be book ent completion [sixe]; prints ("Enter arrival time & burst time of process"); 101 (int 1=0: 12 size: 144) reans ("1.d1.d", farrivalled, found [i]). \$ a manage in the waters int 8um = 0; ent sumbwel = 0; int sumcomp=0: for (int t=0; izsize: i++)/ sum = sum + buset [1]; P completion [i] = sum; Rumbwut = Sumbwut + burst [i]; sum comp = + completion[i]

	Classmate Date Page
	2 151
	and tatsum = 0;
	11-1 1 = 0' i2 s(2c; (++)){
	tatsum = tatsum + TAT(1);
	5 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
	int summait =0;
	for (wint z=0; 12512e; c++) {
	wait [[] = completion [i] - busili];
	summail = summail + wait [i]:
	Suit Police - Suit American de la company de
	float avgtat = tatsum / (float) size;
	float ougcomp = sumcomp/(float) size;
	goat arguait = summait / (gloat) size.
Bosin	print ("avgtat = 1.f avgcomp = 1.f
	ovgwait = 1.4. avgtat. avgwait, avgwait
	there are the same of the same
TEST HOW	OUTPUT A MARKA A . & L B V T JAPAN
	Enter no of process: 4
	Enter avorival time & burst time of
	process 0
	H of marmus dans
	O Mess with and some
	4 That street a must be must
	o mora fill and vigina
	6 James & Jacobson 2 - Fronting
	3 Labouts (max) = dorozowa t

	Date Page 03	
avgtat = 13.250000	Stanting Inge	
avg comp = 18.25 000,		
avgwait = 8.5000		
	E INSPERSE	
Gantt chart	Letak Japan	
PI P2 P3		
0 7 11 1	6 19	
A service shot	January Land	

*SJF(Non-preemptive):

```
#include <stdio.h>
#include <stdlib.h>
struct Process {
  int id;
  int bt;
  int at;
  int ct;
  int tat;
  int wt;
};
int compareArrivalTime(const void *a, const void *b) {
  return ((struct Process*)a)->at - ((struct Process*)b)->at;
}
void calculateTimes(struct Process processes[], int n) {
  int time = 0;
  int completed = 0;
  while (completed < n) {
     int shortest = -1;
     int min_burst = 1000000;
     for (int i = 0; i < n; i++) {
       if (processes[i].at <= time && processes[i].ct == 0) {
          if (processes[i].bt < min_burst) {</pre>
             min_burst = processes[i].bt;
             shortest = i;
        }
     if (shortest == -1) {
       time++;
```

```
} else {
       processes[shortest].ct = time + processes[shortest].bt;
       processes[shortest].tat = processes[shortest].ct - processes[shortest].at;
       processes[shortest].wt = processes[shortest].tat - processes[shortest].bt;
       time = processes[shortest].ct;
       completed++;
     }
  }
}
void calculateAvg(struct Process processes[], int n) {
  int total_wt = 0, total_tat = 0;
  for (int i = 0; i < n; i++) {
     total_wt += processes[i].wt;
     total_tat += processes[i].tat;
  }
  printf("\navg wt = %.2f", (float)total_wt / n);
  printf("\navg tat = %.2f", (float)total_tat / n);
}
int main() {
  int n;
  printf("no. of processes: ");
  scanf("%d", &n);
  struct Process processes[n];
  printf("enter bt and at: \n");
  for (int i = 0; i < n; i++) {
     processes[i].id = i + 1;
     printf("bt %d: ", i + 1);
     scanf("%d", &processes[i].bt);
     printf("at %d: ", i + 1);
```

```
scanf("%d", &processes[i].at);
   processes[i].ct = 0;
 }
 qsort(processes, n, sizeof(struct Process), compareArrivalTime);
 calculateTimes(processes, n);
 calculateAvg(processes, n);
 return 0;
 }
no. of processes: 4
enter bt and at:
bt 2: 3
at 2: 8
bt 3: 4
at 3: 3
bt 4: 6
at 4: 5
```

avg wt = 4.00 avg tat = 9.00%

	CT had a CR array
	#include <stdio. h=""></stdio.>
_	# include < limits. h>
	struct Process (
	int id. AT. BT. CT. TAT. WT. completed;
_	J; SEE SOUT INSPIRE
	void nort Bydonival (struct Process pl).
	for (int i=a; i <n-1; i+){<="" td=""></n-1;>
	for (int j=1+1; j=n; j++)4
	of CPIIT AT > pcj J. AT) }
	struct Proces temp = p[i];
	(ptij = ptj)
	pcjJ = temp; 1 + + + + + + + + + + + + + + + + + +
	49 - TAT TEXABLULE LAS ETATERION
	7

three are	Com the O
rocal calculateSTENER Preemptive (Atruct Preem pl7, ant n){ ant remptited = 0 arrowst Armi = 0; fleat detain(7 = 0; actal TAT = 0; mille (completed < n){ and absolute = 1, minBT = INT_MAT, for lant (=0, ien, ien){ af (spli) remptited & & pli] Al <= arrowshizeme be gli) & 1 < min & 1}	printf ('In Inscend with TAT \a'); for last 1-0; (an seed) (b) The printf ('In Inscend 'In); for last 1-0; (an seed) (b) The printf ('Ind Ind I'd'); printf ('In deringe Wit 'In 21', letal with); printf ('In deringe TAT 'In 21'), letal with);
min 87 : p[i] 87; shartes - 6;	ant main()
if (shortest1) { zwentstime ++;	print ("Enter no of precession"; asan ("Id", 4-n);
p[shanlest] x CT = supremitime + p[shanlest] x CT = supremitime +	shuit Process plat: ten lint t-0; inn, (++)/ ptileid = (+1;
plaheniut] x TAT - plaheniut J CT - plaheniut] x WT - plaheniut J - TAT -	prints ("Enter Americal (AT) and Burst Zime (BT) for process 10" (+1); acan ("Id" led" special AT. special BT);
plahartest 7 × completed = 1; Sotal W1 plahartest 7: NT; Jolah 1871 - plahartest 7: TAT;	plick completed = 0:

*SJF (pre-emptive):

```
#include <stdio.h>
 #include imits.h>
 struct Process {
   int id;
   int bt;
   int at;
   int rt;
   int ct;
   int tat;
   int wt;
 };
 void calculateTimes(struct Process processes[], int n) {
   int completed = 0, time = 0, shortest = -1;
   int min_burst = INT_MAX;
   while (completed < n) {
      shortest = -1;
      min_burst = INT_MAX;
for (int i = 0; i < n; i++) {
         if (processes[i].at <= time && processes[i].rt > 0) {
           if (processes[i].rt < min_burst) {</pre>
              min_burst = processes[i].rt;
              shortest = i;
      }
      if (shortest == -1) {
         time++;
         continue;
      }
```

```
processes[shortest].rt--;
     time++;
     if (processes[shortest].rt == 0) {
       completed++;
       processes[shortest].ct = time;
       processes[shortest].tat = processes[shortest].ct - processes[shortest].at;
       processes[shortest].wt = processes[shortest].tat - processes[shortest].bt;
     }
  }
}
void calculateAvg(struct Process processes[], int n) {
  int total_wt = 0, total_tat = 0;
  for (int i = 0; i < n; i++) {
     total_wt += processes[i].wt;
     total_tat += processes[i].tat;
  }
  printf("\nAvg WT = %.2f", (float)total_wt / n);
  printf("\nAvg TAT = \%.2f", (float)total_tat / n);
}
int main() {
  int n;
  printf("No. of processes: ");
  scanf("%d", &n);
  struct Process processes[n];
  printf("Enter BT and AT: \n");
  for (int i = 0; i < n; i++) {
     processes[i].id = i + 1;
     printf("BT %d: ", i + 1);
     scanf("%d", &processes[i].bt);
     printf("AT %d: ", i + 1);
```

```
scanf("%d", &processes[i].at);
processes[i].rt = processes[i].bt;
}
calculateTimes(processes, n);
calculateAvg(processes, n);
return 0;
}
```

```
No. of processes: 4
Enter BT and AT:
BT 1: 8
AT 1: 0
BT 2: 4
AT 2: 1
BT 3: 9
AT 3: 2
BT 4: 5
AT 4: 3

Avg WT = 6.50
Avg TAT = 13.00%
```

Page O	Classification of the property
# include < stolio h> # include < stolio h> # include < limit h struct Process! int id, AT. BJ. CT. TAT. WT. RT. remaining-BT. completed;	minBT = p[i] remaining = BT; shortest = 6; if (shartest == -1)? remaining = BT = 1; beter 4. if (plshartest J. remaining = BT = 1) plshartest J. RT = remaining = P; plshartest J. RT = remaining = P;
reid calculate STF forcemptive (struct Process p[], int n] { int completed = 0, current Jime = 0; {leat stolative = 0, total TAT = 0; } for (int i = 0; icn; i+1) { p[i]. remaining = BT = p[i] BT; } nehile (completed < n) { int shortest = 1, minBT = INT - MAX; for (int i = 0; icn; c++) { if (!p[i]. completed & & p[i]. AT < = current Jime & & p[i]. remaining - BT < min BT) { min BT) {	p[shoutest] remaining_BT==0){ p[shoutest] x CT = current J.me; p[shoutest] x TAT = p[shoutest] CT- p[shoutest] x TAT = p[shoutest] . TAT- p[shoutest] x wr = p[shoutest] . TAT- p[shoutest] x completed = 1; dotativ(T += p[shoutest] . WT; total TAT+ = p[shoutest] . TAT: "completed ++: "

classmate	
Q Page Page	CA CA
printf ("In Process + WI + TAI In");	OUTFUT:
400 lant 1=0: 1<0; 6++)d	Enter no of pracuses: 4
perints (" I'd I'd I'd", plid. d. plid. WT	Enter AT & BT : 0 8
ρ[(J. ΤΑΤ);	Enter AT & BT : 1 4
4	enter AT & BT 29
prints ("In dverage WIT: 1.24", total WIIn)	Enter AT & BT 3 5
prints ("In duerage TAT: 1.24", lotal TAT In)	and Anadone in web teasts him at
S Control	yant chart
The state of the s	
int main()(P. P2 P4 P1 P2
int n;	0 1 5 10 14 26
print ("Enter no of processes");	Markey States
scan ("1d", 4n);	AVA WT 65
and the second s	AVG TAT: 13
struct perocen plnj;	The same and an all the same
for funt (=0; (cn; 6++))	Karaman Alan
p[1] • cd = t+1;	Nat /
prints ("Enter AT and BT for process id.	W. r.
- In Chaband, his a mitter directs of the (+1);	30
man ("td td" spli). AT. &plij. BT);	0
plida completed =0;	
3	Adam non har continu
- Parising and Marian As Inc.	his words a manufactor for some 1 1 x 1 to
calculate SJF Preemptive (p.n);	The plantage and a state of the
return 0	13-14-14-15-1-14-1-1-1-1-1
4	
	A STATE OF THE STA

2. Write a C program to simulate the following CPU scheduling algorithm to find turnaround time and waiting time.

 \rightarrow Priority (Non-pre-emptive):

```
#include <stdio.h>
#include <stdlib.h>
struct Process {
  int id, bt, at, priority, ct, tat, wt;
};
int comparePriority(const void *a, const void *b) {
  struct Process *p1 = (struct Process *)a;
  struct Process *p2 = (struct Process *)b;
  if (p1->at == p2->at)
     return p1->priority - p2->priority;
  return p1->at - p2->at;
}
void calculateTimes(struct Process processes[], int n) {
  int time = 0, completed = 0;
  while (completed < n) {
     int highest = -1, highestPriority = 1000000;
     for (int i = 0; i < n; i++) {
        if (processes[i].at \le time \&\& processes[i].ct == 0) {
          if (processes[i].priority < highestPriority) {</pre>
             highestPriority = processes[i].priority;
             highest = i;
        }
     if (highest == -1) {
```

```
time++;
     } else {
       processes[highest].ct = time + processes[highest].bt;
       processes[highest].tat = processes[highest].ct - processes[highest].at;
       processes[highest].wt = processes[highest].tat - processes[highest].bt;
       time = processes[highest].ct;
       completed++;
     }
  }
}
void calculateAvg(struct Process processes[], int n) {
  int total_wt = 0, total_tat = 0;
  for (int i = 0; i < n; i++) {
     total_wt += processes[i].wt;
     total_tat += processes[i].tat;
  printf("\nAvg WT = %.2f", (float)total_wt / n);
  printf("\nAvg TAT = \%.2f", (float)total_tat / n);
}
int main() {
  int n;
  printf("No. of processes: ");
  scanf("%d", &n);
  struct Process processes[n];
  for (int i = 0; i < n; i++) {
     processes[i].id = i + 1;
     printf("BT %d: ", i + 1);
     scanf("%d", &processes[i].bt);
     printf("AT %d: ", i + 1);
     scanf("%d", &processes[i].at);
     printf("Priority %d: ", i + 1);
     scanf("%d", &processes[i].priority);
     processes[i].ct = 0;
```

```
qsort(processes, n, sizeof(struct Process), comparePriority);
calculateTimes(processes, n);
calculateAvg(processes, n);
return 0;
}

No. of processes: 5
BT 1: 3
AT 1: 0
```

```
BT 1: 3
AT 1: 0
Priority 1: 5
BT 2: 2
AT 2: 2
Priority 2: 3
BT 3: 5
AT 3: 3
Priority 3: 2
BT 4: 4
AT 4: 4
Priority 4: 4
BT 5: 1
AT 5: 6
Priority 5: 1
Avg WT = 3.20
Avg TAT = 6.20
```

	Copy 11
Criarity Non pre emplace	ul (sharlest = = -1) L
Hindurch estation his	time ++;
The state of the s	centinue
struct Process	The state of the s
int pick arrival time, burst time	the first and the time +
completion time waiting time.	plshortest] completion-time = time +
turn around time;	plshoutest J. burst time;
9:	plshoulest I turnaround - time =
	prohoutest] completion - time - prohoutest
void periority Non Preemptive (struct	planertest I wailing time = planertest d.
enous pro, int not	turnacound-time -platortest 1 burst-tim
int completed = 0 time = 0, min	timportung - une - promotion retros
priority, shortist;	total - waiting + = plahortest 1 waiting
great total waiting = 0, total	time.
turn accound = 0	total-turnaround += pishertist].
	turnaveund time;
netule (completed < n) L	time += planertest J. bourt - time;
min-periority = 9999;	completed ++;
shortest = -1;	6 to 1 14 a 1 Table 0. 1 to 1 t
	And Best Dies at Decree Could
for (int i=0; (en; (4)) (prints (Non Pre-emptive Princily
uf (pli) arrival time < = time &&	scheduling (n);
prid powerity < min-priority 44	painty ("PIDIT AT IT BT IT PIT ET IT TATI
pris completion - time ==0)	WT \t");
1	for (int 100; cen; (+1);
min-paranity = ptc I privarity.	prenty (" tart 1 & rt / drt / drt 1 dre 1 dr 1.
shortest = c;	plit, pid, plit averval time, plit brust
, 9	time, plis. priority, plis completion.

	A.
ang	OUTPUT:
1.211	Enter no of processes 5 Enter Process ID. derival Jime. Event Jime. Privarity: 1 0 4 2 Enter Process ID. derival Jime. Event Jime. Privarity: 2 1 3 3 Enter Process ID. derival Jime. Burst Jime. Privarity: 3 2 1 4 Enter Process ID. derival Jime. Burst Jime. Privarity: 4 3 5 5 Enter Process ID. derival Jime. Burst Jime. Privarity: 4 3 5 5
Mada	Non Orcemptive Orcarity Scheduling PID AT BT P CT TAT WIT
me,	1 0 4 2 4 4 0 2 1 3 3 1 6 3 3 2 1 4 8 6 5
me.	4 3 5 5 13 10 5 5 4 2 5 15 11 9
	during June: 4.40 duringe Junnavound Jime: 7:40
7	1.17 Mary 201 1 and 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
	me.

,

→ Priority (pre-emptive):

```
#include <stdio.h>
#include <stdlib.h>
struct Process {
  int id, bt, at, priority, ct, tat, wt, rt;
};
int compareArrival(const void *a, const void *b) {
  struct Process *p1 = (struct Process *)a;
  struct Process *p2 = (struct Process *)b;
  return p1->at - p2->at;
}
void calculateTimes(struct Process processes[], int n) {
  int time = 0, completed = 0, min_priority, shortest;
  for (int i = 0; i < n; i++) processes[i].rt = processes[i].bt;
  while (completed < n) {
     shortest = -1;
     min_priority = 1000000;
     for (int i = 0; i < n; i++) {
       if (processes[i].at <= time && processes[i].rt > 0 && processes[i].priority <
min_priority) {
          min_priority = processes[i].priority;
          shortest = i;
        }
     }
```

```
if (shortest == -1) {
       time++;
     } else {
       processes[shortest].rt--;
       time++;
       if (processes[shortest].rt == 0) {
          processes[shortest].ct = time;
          processes[shortest].tat = processes[shortest].ct - processes[shortest].at;
          processes[shortest].wt = processes[shortest].tat - processes[shortest].bt;
          completed++;
     }
  }
void calculateAvg(struct Process processes[], int n) {
  int total_wt = 0, total_tat = 0;
  for (int i = 0; i < n; i++) {
     total_wt += processes[i].wt;
     total_tat += processes[i].tat;
  }
  printf("\nAvg WT = %.2f", (float)total_wt / n);
  printf("\nAvg TAT = %.2f", (float)total_tat / n);
}
int main() {
  int n;
  printf("No. of processes: ");
  scanf("%d", &n);
  struct Process processes[n];
  for (int i = 0; i < n; i++) {
     processes[i].id = i + 1;
     printf("BT %d: ", i + 1);
     scanf("%d", &processes[i].bt);
     printf("AT %d: ", i + 1);
     scanf("%d", &processes[i].at);
```

```
printf("Priority %d: ", i + 1);
    scanf("%d", &processes[i].priority);
}

qsort(processes, n, sizeof(struct Process), compareArrival);
calculateTimes(processes, n);
calculateAvg(processes,n);
return 0;
}
```

```
No. of processes: 7
BT 1: 8
AT 1: 0
Priority 1: 3
BT 2: 2
AT 2: 1
Priority 2: 4
BT 3: 4
AT 3: 3
Priority 3: 4
BT 4: 1
AT 4: 4
Priority 4: 5
BT 5: 6
AT 5: 5
Priority 5: 2
BT 6: 5
AT 6: 6
Priority 6: 6
BT 7: 1
AT 7: 7
Priority 7: 1
Avg WT = 9.86
Avg TAT = 13.71
```

Round Robin:

```
#include <stdio.h>
struct Process {
  int id, bt, at, rt, ct, tat, wt;
};
void roundRobin(struct Process processes[], int n, int quantum) {
  int time = 0, completed = 0;
  while (completed < n) {
     int done = 1;
     for (int i = 0; i < n; i++) {
       if (processes[i].rt > 0 && processes[i].at <= time) {
          done = 0;
          if (processes[i].rt > quantum) {
             time += quantum;
             processes[i].rt -= quantum;
          } else {
             time += processes[i].rt;
             processes[i].ct = time;
             processes[i].tat = processes[i].ct - processes[i].at;
             processes[i].wt = processes[i].tat - processes[i].bt;
             processes[i].rt = 0;
             completed++;
          }
        }
     if (done) time++;
  }
}
void calculateAvg(struct Process processes[], int n) {
  int total_wt = 0, total_tat = 0;
  for (int i = 0; i < n; i++) {
     total_wt += processes[i].wt;
     total_tat += processes[i].tat;
```

```
printf("\nAvg WT = %.2f", (float)total_wt / n);
  printf("\nAvg\ TAT = \%.2f", (float)total\_tat / n);
}
int main() {
  int n, quantum;
  printf("No. of processes: ");
  scanf("%d", &n);
  struct Process processes[n];
  for (int i = 0; i < n; i++) {
     processes[i].id = i + 1;
     printf("BT %d: ", i + 1);
     scanf("%d", &processes[i].bt);
     printf("AT %d: ", i + 1);
     scanf("%d", &processes[i].at);
     processes[i].rt = processes[i].bt;
  }
  printf("Time Quantum: ");
  scanf("%d", &quantum);
  roundRobin(processes, n, quantum);
  calculateAvg(processes, n);
  return 0;
}
```

```
No. of processes: 5
BT 1: 8
AT 1: 0
BT 2: 2
AT 2: 5
BT 3: 7
AT 3: 1
BT 4: 3
AT 4: 6
BT 5: 5
AT 5: 8
Time Quantum: 3

Avg WT = 10.40
Avg TAT = 15.40
```

```
of write a Copy of found - Roben
                                                                     of procesus; 5
solly it include a stolen hy
                                                                         Enter BT and AT:
     struct Proces (
4w kd, bt, at, ot, ct, tat, wt;
                                                                         BTLIS
                                                                         AT1:0
                                                                         BT2:2
     word PR(struct Process process[], but ny ant quantum)
                                                                         AT2 :5
        < mu term = 0, completed = 0;
                                                                         BT3:7
           while (completed en) +
                                                                         AT3:1
                   Post (m+ 9=0; Jun; 3++) (
                                                                         BT4:3
                                                                         AT4:6
                      & (broom [0.24 >0 84
                                                                         BTS:5
                         processed at c= throwy
                                                                        ATS: 8 Teme Quantum: 3
                 of Core of to Squantum ) {
                      Home += quantum;
                                                                        ANG TAT: 15 . 40
                      process [8] of - = quantum;
                     Yelsi 1
                   Hence += process(1) out;
                    process (B. et = tem; process (B. et = process (B. et = process (B. et = process (B. et = process (B. et))
                                                                                              22
                                                                                                      14
                                                                                     11
                                                                               2
                                                                                     23
                                                                                              22
                                                                                                      15
                     processed at =0;
                   y completed ++;
                                                                                     14
                                                                                              8
                                                                                     25
                                                                   15
                                                                                             17
              y (done) trme+;
word calangestrum procumprounts, int nx
                                                                    PAY
           for (At 9=0) sen; stallat=0;
                   total aut += process [8]. cut;
                  totaltat += procus(1). tat;
```

3. Write a C program to simulate multi-level queue scheduling algorithm considering the following scenario. All the processes in the system are divided into two categories – system processes and user processes. System processes are to be given higher priority than user processes. Use FCFS scheduling for the processes in each queue.

→Multilevel scheduling:

```
#include <stdio.h>
#define MAX_PROCESSES 10
typedef struct {
  int pid;
  int bt;
  int at:
  int queue;
} Process;
void sortByArrival(Process p[], int n) {
  Process temp;
  for (int i = 0; i < n - 1; i++) {
     for (int j = 0; j < n - i - 1; j++) {
        if (p[j].at > p[j + 1].at) {
          temp = p[j];
          p[j] = p[j + 1];
          p[j + 1] = temp;
     }
```

```
void roundRobin(Process p[], int n, int quantum, int wt[], int tat[], int rt[]) {
  int remaining_bt[MAX_PROCESSES];
  for (int i = 0; i < n; i++)
     remaining_bt[i] = p[i].bt;
  int t = 0, completed = 0;
  while (completed < n) {
     int executed = 0;
     for (int i = 0; i < n; i++) {
       if (remaining_bt[i] > 0) {
          if (rt[i] == -1) rt[i] = t;
          if (remaining_bt[i] > quantum) {
             t += quantum;
             remaining_bt[i] -= quantum;
          } else {
             t += remaining_bt[i];
             tat[i] = t - p[i].at;
             wt[i] = tat[i] - p[i].bt;
             remaining_bt[i] = 0;
             completed++;
          executed = 1;
        }
     if (!executed) t++;
}
void fcfs(Process p[], int n, int start_time, int wt[], int tat[], int rt[]) {
  int time = start_time;
  for (int i = 0; i < n; i++) {
     if (time < p[i].at) time = p[i].at;
     rt[i] = time - p[i].at;
     wt[i] = rt[i];
     tat[i] = wt[i] + p[i].bt;
     time += p[i].bt;
```

```
}
int main() {
  int n, quantum;
  Process p[MAX_PROCESSES], sys[MAX_PROCESSES],
usr[MAX_PROCESSES];
  int sys_count = 0, usr_count = 0;
  int wt[MAX_PROCESSES], tat[MAX_PROCESSES], rt[MAX_PROCESSES];
  printf("NO. of process: ");
  scanf("%d", &n);
  for (int i = 0; i < n; i++) {
    printf("Enter BT,AT and Q (1=sys, 2=user) for P%d: ", i + 1);
    p[i].pid = i + 1;
    scanf("%d %d %d", &p[i].bt, &p[i].at, &p[i].queue);
    if (p[i].queue == 1)
       sys[sys\_count++] = p[i];
    else
       usr[usr\_count++] = p[i];
    wt[i] = 0;
    tat[i] = 0;
    rt[i] = -1;
  }
  printf("QT for RR: ");
  scanf("%d", &quantum);
  sortByArrival(sys, sys_count);
  sortByArrival(usr, usr_count);
  roundRobin(sys, sys_count, quantum, wt, tat, rt);
  int last_sys_time = (sys\_count > 0)? tat[sys\_count - 1] + sys[sys\_count - 1].at : 0;
  fcfs(usr, usr_count, last_sys_time, &wt[sys_count], &tat[sys_count],
&rt[sys_count]);
```

```
printf("\nP\tQ\tWT\tTAT\tRT\n");
  for (int i = 0; i < n; i++)
     printf("P\%d\t\%d\t\%d\t\t\%d\t\t\%d\n", p[i].pid, p[i].queue, wt[i], tat[i], rt[i]);
  float avg_wt = 0, avg_tat = 0, avg_rt = 0;
  for (int i = 0; i < n; i++) {
     avg_wt += wt[i];
     avg_tat += tat[i];
     avg_rt += rt[i];
  }
  printf("\nAVG WT: %.2f", avg_wt / n);
  printf("\nAVG TAT: %.2f", avg_tat / n);
  printf("\nAVG RT: %.2f\n", avg_rt / n);
  return 0;
NO. of process: 4
Enter BT,AT and Q (1=sys, 2=user) for P1: 2 0 1
Enter BT,AT and Q (1=sys, 2=user) for P2: 1 0 2
Enter BT,AT and Q (1=sys, 2=user) for P3: 5 0 1
Enter BT,AT and Q (1=sys, 2=user) for P4: 3 0 2
OT for RR: 2
```

0

2

AVG WT: 4.25 AVG TAT: 7.00 AVG RT: 4.25

1

2

2

P1

Р3

WT

0

2

8

TAT

RT

2

7

8

11

CLASSMALE Date Page	Classmate Profe 15
multilevel	int rumaining-bt[MAX-PROCESSES]
#include sstdio h>	ten lint i=0; i=n; i++) numcining-belli] = plij-bt;
# define MAX- PROCESSES 10	int t=0, completed=0;
typides struct i	while (completed < n) &
int bt:	int inecuted = 0; for (int 1=0; in 1117770)
int queue:	ab (remaining-btlit >0) (ab (relit==-1) relit=t:
veid sortbydraival (faces pl.), intoli	of (sumaining-bt(i) > quantum){
Process temp; par (cnt i=0; i=n-1; i++)	t+ = quantum: numaining = bt[i] -= quantum; belse f
for (int j=0; j=n-1; j=+) (ub (p[j].at > p[j+1] at) (tatiij = t - prijat;
temp = p[j]; p[j] = p[j+1];	net[i] = tat[i] - p[i] bt; numaining - bt[i] = 0;
p[j+1] = temp;	completed ++;
4 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	enecuted = 1;
void round Robin (Process pl.), int. n.	if (!executed) +++;
int quantum, ent net[], int tail]	Zin T Jan Kar examente anamente da la
	to but a set with the A. V. A. a. A. C. Coppension of

CIASSMATE Date Page	Classific Copy 17
void fife (fracen pl), into, int start time, int netis, int tatis int at[])! int time = start time; int time = start time; for (int i=0; icn; i++); if (time < plis, at) time = plis, at rtlis = time - plis, at; rtlis = rtlis; tatlis = retlis + plis, bt;	&ptid.queue); uf (ptid.queue = = 1) syp [seys - count ++] = ptid; else un [un - count ++] = ptid; nettid = 0;
y time + - p[i].bt;	prints ("Enter time quantum"); prints ("Id", & quantum).
ent moun () { int n, quantum; luces p[MAx_PROCESSES], sys [MAx_PROCESSES], ux[MAx_PROCESSES] ant nys-count = 0; ux (count = 0)	sent By drawal (sys. sys. count); sent By drawal (use, sys. count);
PROCESSES J. nt[MAx-processes].	int last-sys_dime = (sys_count > 0)? tat [sys_count - 1] + sys [sys_count - 1]
print ("Enter no. of processes"): scanb ("1.d", 80): for (int i=0; (40; c++) {	fefs (ust run wint, last-sep-time. sont [sep count]. stat tsep-count], 4 mt [rep - count]);
print ("talin BT. AT and Queue (1= Cystim. 2= Wer) for pol-d:", (+1). plil pid = (+);	priorits ("In Process Lt Queue Lt WIT Lt TAT LE RT In"); for (int i=0; i <n; i+1)<="" td=""></n;>
reans ("Idid Id", split bt, split at.	paintl ("Pold to be d to be le bed to be d to be d to ".



Date Page
prij. pid. prij. quene, netlij. ant lij. nt lij);
float avg-net=0, avg-tat=0, avg-nt=0;
avg-net += net li]: avg-tat += tat [i];
avg-at += atli
9
print ("In Avg INIT: % 2f", avg-w+ In).
paints ("In Avg Rt: 1024". avg- stin):
OUTPUT:
Enter BT. AT and Queue, (1 = System.
2 = User) per P1: 2 0 1
P2 : (0 2
P3: 6 0 1
P4:302
Process Queue MIT TAT RT
P1 1. 1. 0 2 0
P2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
P3
Pu 2 8 11 8
Avg CHARROSS 10 W FREE LAND
NOT FRED THAT BUT A DEPART OF THE PARTY

4. Write a C program to simulate Real-Time CPU Scheduling algorithms:

a) Rate- Monotonic:

```
#include <stdio.h>
#include <stdlib.h>
#define MAX_PROCESSES 10
#define MAX_TIME 50 // Maximum simulation time
typedef struct {
  int pid;
  int burst;
  int period;
  int remaining_time;
  int next_arrival;
} Process;
void rate_monotonic_scheduling(Process p[], int n) {
  int time = 0, executed;
  printf("\nRate Monotonic Scheduling:\n");
  printf("PID\tBurst\tPeriod\n");
  for (int i = 0; i < n; i++)
    printf("%d\t%d\n", p[i].pid, p[i].burst, p[i].period);
  while (time < MAX_TIME) {
    executed = -1;
    for (int i = 0; i < n; i++) {
       if (p[i].next_arrival <= time && p[i].remaining_time > 0) {
         if (executed == -1 || p[i].period < p[executed].period)
            executed = i;
       }
    if (executed !=-1) {
```

```
printf("%dms : Task %d is running.\n", time, p[executed].pid);
       p[executed].remaining_time--;
       if (p[executed].remaining_time == 0) {
         p[executed].next_arrival += p[executed].period;
         p[executed].remaining_time = p[executed].burst; // Reset for periodic
execution
    time++;
}
int main() {
  int n;
  Process processes[MAX_PROCESSES];
  printf("Enter the number of processes: ");
  scanf("%d", &n);
  printf("Enter the CPU burst times:\n");
  for (int i = 0; i < n; i++) {
    processes[i].pid = i + 1;
    scanf("%d", &processes[i].burst);
    processes[i].remaining_time = processes[i].burst;
  }
  printf("Enter the time periods:\n");
  for (int i = 0; i < n; i++) {
    scanf("%d", &processes[i].period);
    processes[i].next_arrival = 0;
  }
  rate_monotonic_scheduling(processes, n);
  return 0;
```

```
Enter the number of processes: 3
Enter the CPU burst times:
3 6 8
Enter the time periods:
3 4 5

Rate Monotonic Scheduling:
PID Burst Period
1 3 3
2 6 4
3 8 5
```

	classmate Dote Page 19
	Rate Monotonic
	#include <stdio.h> #include <stdub.h></stdub.h></stdio.h>
100	typedel struct l
	int pid;
	int period; int nemaining time; int next-arrival;
	4 Process
	noid rate monotonic (Process pl). int n) { int time = 0, executed:
	paint ("In Rate Monotonic Scheduling): paint ("PIDIT Burit It Period In");
	paint ("Idit bed to did n", p[i]. pid, piil. burst. p[i]. puriod);
1 _{ex}	nihile (time < MAX_TIME)();
et al	Jen (int 16 = 0; 12 cm; 16++){
- ()	of (p[i]. next-avoiral <= time ff
	p[exicuted] == -1 p[i]. period < p[exicuted] J. period]

4	Classwate Date Page		Classmate Date 21
	in (executed = -1) {		print[("Enter time perioditi"); for (int c=0: ion c++)/ scan[("Id", & processes (id).guiod); processes (id next-arrival =0; }
	prints (1 d Jask it d is runnings time, plenuted J. ped):		nate-monotonic (processes, n);
	plexicuted J. next - arrival += plexicuted J. next - arrival += plexicuted J. penied, plexicuted J. remaining time.		return 0;
36.4.	pleximited) burst		OUTPUT: Enter no. of process: 3
- Alledon	time++;	175	Enter time period 3 4 5 LIM = 60
lang, ba	int n: Process Process [MAX_PROCESSES]	Janes.	11 3 3 2 6 4 3 9
3.2 57 az	prints ("Enler the CPU burst time In"). Jen (int 6=0; (xn; 1++)? procency (id procency (id burst); procency (id 4 procency (id burst); procency (id xmaining time =	1)4,27	41000 2 = 0 7797 = fake
Uni	process [i] burnt:		

b)Earliest-deadline:

```
#include <stdio.h>
int gcd(int a, int b) {
  while (b != 0) \{
     int temp = b;
     b = a \% b;
     a = temp;
  return a;
}
int lcm(int a, int b) {
  return (a * b) / gcd(a, b);
}
struct Process {
  int id, burst_time, deadline, period;
};
void earliest_deadline_first(struct Process p[], int n, int time_limit) {
  int time = 0;
  printf("Earliest Deadline Scheduling:\n");
  printf("PID\tBurst\tDeadline\tPeriod\n");
  for (int i = 0; i < n; i++) {
     printf("%d\t%d\t\t%d\t\t%d\n", p[i].id, p[i].burst_time, p[i].deadline,
p[i].period);
  }
  printf("\nScheduling occurs for %d ms\n", time_limit);
  while (time < time_limit) {
```

```
int earliest = -1;
     for (int i = 0; i < n; i++) {
        if (p[i].burst\_time > 0) {
          if (earliest == -1 \parallel p[i].deadline < p[earliest].deadline) {
             earliest = i;
        }
     }
     if (earliest == -1) break;
     printf("%dms: Task %d is running.\n", time, p[earliest].id);
     p[earliest].burst_time--;
     time++;
  }
}
int main() {
  int n;
  printf("Enter the number of processes: ");
  scanf("%d", &n);
  struct Process processes[n];
  printf("Enter the CPU burst times:\n");
  for (int i = 0; i < n; i++) {
     scanf("%d", &processes[i].burst_time);
     processes[i].id = i + 1;
  }
  printf("Enter the deadlines:\n");
  for (int i = 0; i < n; i++) {
     scanf("%d", &processes[i].deadline);
  }
  printf("Enter the time periods:\n");
  for (int i = 0; i < n; i++) {
     scanf("%d", &processes[i].period);
```

```
int hyperperiod = processes[0].period;
  for (int i = 1; i < n; i++) {
    hyperperiod = lcm(hyperperiod, processes[i].period);
   }
  printf("\nSystem will execute for hyperperiod (LCM of periods): %d ms\n",
hyperperiod);
  earliest_deadline_first(processes, n, hyperperiod);
  return 0;
}
Enter the number of processes: 3
Enter the CPU burst times:
2 3 4
Enter the deadlines:
123
Enter the time periods:
123
System will execute for hyperperiod (LCM of periods): 6 ms
Earliest Deadline Scheduling:
PID
         Burst
                  Deadline
                                    Period
         2
                           1
                                             1
2
                           2
                                             2
         3
         4
                           3
                                             3
Scheduling occurs for 6 ms
Oms: Task 1 is running.
1ms: Task 1 is running.
2ms: Task 2 is running.
3ms: Task 2 is running.
4ms: Task 2 is running.
5ms: Task 3 is running.
```

Classwatz Dote Page	classmate tute Page 23
tarlius deadline	painth ("In Scheduling occurs for 1.d m in", time-lamit):
stinctude «cidio k v	while (time < time_limit) { int earliest = -1;
int gcd (int a, int b) {	for (int 1=0; icn: 1+1) {
nehule (b!=0){ int temp = b;	up (p(1) burst time >0) d
b=a1b; a=temp;	earliest = i;
neturn a	<u> </u>
int lam lint a. int b) (4
return (a+b/gcd (a.b)).	point ("Idm Jask Idu aunning)
struct Process &	time, prearliest I id);
int id. burnt-time, deadline, persed;	prearlist J. burst time;
3: house of some of the sound	ation by the transport of the same of the
void earliest deadline first (struct process p(N), int n. int time limit)	int main () (
print ("fartist deadline in);	print ("enter no. of processes.");
printh ("PID It BIT DLIT PIN");	N. Namera
for (cnt i=0: i <n: ("1.d="" (++)="" 1.d="" \t="" \t,="" pli)<="" printl="" td="" {=""><td>printle ("Enter CPU BT: \n");</td></n:>	printle ("Enter CPU BT: \n");
pid. p[i]. BT, p(i]. dt. p[i]. period);	yer (int i=0; i <n; &="" ("i-d";="" [i]<="" i++){="" process="" scan="" td=""></n;>

CIASSAALE Date Page	classaule Date
procusu [1] x sol = i+1;	OUTPUT
14	Enter the no of processes 3
	Enter CPV schiduling B1 2 3 4
prints ("Enter the deadlines In)	Enter deadline 123
1 - 1 - 1 - 1 - D. LED. C+1)K	Enter time period 123
scant ("Id". & processes (IV. d1).	enur simi period
3	
1/11 P. St 10 - 1 - 1	system will encute for hyperperiod
prints ("Enter time period In");	PID RT OF P
for (int i=0; i <n, i+){<="" td=""><td>PID BT DL P</td></n,>	PID BT DL P
man ("Id" & processer [1] period);	
L According to	
	3 4 3 3
int hyper period = processes to J x period for lint (=1; i < n; i++)L	Schiduling occurs for 6 ms
hyperperiod = lon (hyper period.	0 ms Jask 1 running
processes (id. puriced);	1 m - u -
	2 ms Josk 2 tunning
print(("In System will execute in	3 ms - 11
hyperperiod (x(M of period); 1.dm)	n' 4 m
hyperpercod);	6 ms Jask 3 nunning
	TO HIS OWN & RUTTUL TO
earlist deadline just (procuses &	
hyperperiod)	- Ul
netwon of	121
	A Salta A L. Lander and Marie
y (Ge) conserve and finally	
	with the standard with the standard and
ENT CONTRACTOR OF THE PROPERTY	Elistana Ingga a stage
Still manager to the later of the same	The second secon

5. Write a C program to simulate producerconsumer problem using semaphores

producer-consumer:

```
#include <stdio.h>
#include <stdlib.h>
int mutex = 1;
int full = 0;
int empty=3;
int buffer_item = 0;
int wait(int s) {
  return (--s);
}
int signal(int s) {
  return (++s);
}
void producer(int id) {
  if ((mutex == 1) && (empty != 0)) {
     int value;
     scanf("%d", &value);
     printf("producer %d produced %d\n", id, value);
     mutex = wait(mutex);
     full = signal(full);
     empty = wait(empty);
     buffer_item = value;
     printf("buffer:%d\n", buffer_item);
     mutex = signal(mutex);
  } else {
     printf("buffer is full!\n");
```

```
void consumer(int id) {
  if ((mutex == 1) && (full != 0)) {
     mutex = wait(mutex);
    full = wait(full);
    empty = signal(empty);
    printf("consumer %d consumed %d\n", id + 1, buffer_item);
    printf("current buffer len: 0\n");
    mutex = signal(mutex);
  } else {
    printf("buffer is empty!\n");
}
int main() {
  int np, nc, cap;
  printf("enter no. of producers:");
  scanf("%d", &np);
  printf("enter no. of consumers:");
  scanf("%d", &nc);
  printf("enter buffer capacity:");
  scanf("%d", &cap);
  empty = cap;
  for (int i = 1; i \le np; i++) {
    printf("producer created: %d\n", i);
  }
  for (int i = 1; i \le nc; i++) {
    printf("consumer created: %d\n", i);
  }
  while (1) {
    producer(1);
     consumer(1);
```

```
}
return 0;
}
```

```
enter no. of producers:1
enter no. of consumers:1
enter buffer capacity:1
producer created: 1
consumer created: 1
16
producer 1 produced 16
buffer:16
consumer 2 consumed 16
current buffer len: 0
32
producer 1 produced 32
buffer:32
consumer 2 consumed 32
current buffer len: 0
4
producer 1 produced 4
buffer:4
consumer 2 consumed 4
current buffer len: 0
producer 1 produced 0
buffer:0
consumer 2 consumed 0
current buffer len: 0
```

Dele Pope	classrate
a So	Prop 41
brocheer benumer	printl (buffer is full (n);
#include «sidio. hu	4
#include estable ho	Management of Adaptives
int mutex = 1;	veid consumer lint soll
int full = 0;	mutex = nait (mutex);
int empty = 3,	full " wait (full);
int buffer item = 0;	empty = signal (empty).
int wait (int s) f	point ("consumer 1 of consumed 1 of)
networn (s);	perint ("current buffer len o'n"):
ant signal (ints)(mutex = signal (mutex);
return (++5)	prints ("buster in empty 11");
y and the same of	4
veid produces (ent ed)	3
ub ((mutex == 1) 44 (empty 1=0)){	ant main () t
scanf ("Id" fralue).	int np. nc. cap.
printl ("produces 1 d preduced 1 d's	prints ("Enter no of producers"):
muter - wait (mutex);	perint ("Id", enp)
full : signal (full);	stan ("old", 4nc);
empty : wait (empty):	printh ("Enter bubbis capacity");
print! ("buffer: 1.d \n", buffer itim)	scant ("t.d", ecap);
mutex = signal (mutex);	empty · cap;
y else (

	Date Page
	for (int i=1; i <= np; i++) { print ("producer mated: 4d n)
-	for (unt i=1; ic=nc; i++)l
-	nehule (1) { peroducer (1);
Tatla de 1	consumer (1)
· (*	neturn 0;
=	Enter no. of producers 1 Enter no. of consumers 1
	Enter bubber capacity: 1 Successfully created producer 1 Producer 1 produced 16
- (Buffer: 16 bonumer 2 consumed 16
	Producer 1 produced 4
	Consumer 2 consumed 4 Consumer buffer sen:0
	The section of the se

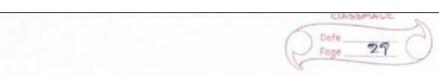
6. Write a C program to simulate the concept of Dining Philosophers problem.

```
#include <stdio.h>
#include <stdlib.h>
#include <pthread.h>
#include <semaphore.h>
#include <unistd.h>
#define MAX 5
sem_t mutex;
sem_t chopstick[MAX];
int totalPhilosophers;
int hungryCount;
int hungryPhilosophers[MAX];
int choice;
void *philosopher(void *arg) {
  int id = *(int *)arg;
  printf("P %d is waiting\n", id + 1);
  sem_wait(&mutex);
  sem_wait(&chopstick[id]);
  sem_wait(&chopstick[(id + 1) % totalPhilosophers]);
  printf("P %d is granted to eat\n", id + 1);
  sleep(1);
  printf("P %d has finished eating\n", id + 1);
  sem_post(&chopstick[id]);
  sem_post(&chopstick[(id + 1) % totalPhilosophers]);
  sem_post(&mutex);
  pthread_exit(NULL);
}
int main() {
  pthread_t thread[MAX];
  printf("Enter the total number of philosophers: ");
  scanf("%d", &totalPhilosophers);
```

```
printf("How many are hungry: ");
scanf("%d", &hungryCount);
for (i = 0; i < hungryCount; i++) {
  printf("Enter philosopher %d position (1 to %d): ", i + 1, totalPhilosophers);
  scanf("%d", &hungryPhilosophers[i]);
  hungryPhilosophers[i]--;
for (i = 0; i < totalPhilosophers; i++) {
  sem_init(&chopstick[i], 0, 1);
sem_init(&mutex, 0, 1);
while (1) {
  printf("\n1. One can eat at a time\n");
  printf("2. Two can eat at a time\n");
  printf("3. Exit\n");
  printf("Enter your choice: ");
  scanf("%d", &choice);
  if (choice == 3) {
     break;
  if (choice == 1) {
     printf("Allow one philosopher to eat at any time\n");
     for (i = 0; i < hungryCount; i++) {
       for (int j = 0; j < hungryCount; j++) {
          printf("P %d is waiting\n", hungryPhilosophers[j] + 1);
       int *id = malloc(sizeof(int));
       *id = hungryPhilosophers[i];
       pthread_create(&thread[i], NULL, philosopher, id);
       pthread_join(thread[i], NULL);
     }
  }
return 0;
```

```
How many are hungry: 3
Enter philosopher 1 position (1 to 5): 2
Enter philosopher 2 position (1 to 5): 4
Enter philosopher 3 position (1 to 5): 5

1. One can eat at a time
2. Two can eat at a time
3. Exit
Enter your choice: 1
Allow one philosopher to eat at any time
P 2 is waiting
P 4 is waiting
P 5 is waiting
P 2 is waiting
P 2 is waiting
P 4 is waiting
P 4 is waiting
P 5 is waiting
P 4 is waiting
P 4 is waiting
P 5 is waiting
P 5 is waiting
P 6 is waiting
P 7 is waiting
P 8 is waiting
P 9 is waiting
P 9 is waiting
P 10 is waiting
P 11 is waiting
P 22 is waiting
P 3 is waiting
P 4 is waiting
P 5 is waiting
P 5 is waiting
P 5 is granted to eat
P 5 has finished eating
P 5 is granted to eat
P 5 has finished eating
P 6 is waiting
P 7 is granted to eat
P 8 is waiting
P 9 is granted to eat
P 10 one can eat at a time
P 11 one can eat at a time
P 12 is waiting
P 13 is waiting
P 14 is waiting
P 15 is waiting
```



Poge 29
Dining Philosopher
and the state of t
Hindude < pthouad hs
Hinchide «stolio.h»
Landers and hard technical to be be a special and the
define N5
sem-t muten;
sem-t chopstick [MAX]
int totalehilosophers;
int hungey count;
ant hungryphilesophers [MAX];
ant choice;
void + philosopher (void + arg) {
int id = * (int *) ang;
print ("P ted is waiting in. id +1);
sem-wait (&mutex);
sem-wait (& chopaticks (id));
cem-wait (& chopsticks (id+1)%
total Philosophers);
paint ("P oled is asanted to eat" id+1).
perint ("P % u granted to eat" id+1).
print ("P1.d has finished eating in"
Mas = 1345/15 1 (id +1);
sem-post (& chopatick [id]);
sem-post (4 chopstick (id+1) %
dotal Philosophere J);
Authorities)
sem-post (& mutcz);
pthouad-exit (NULL);
11.2

	Date Page		classmate Dute Prop 31
	int main ()		ul (chorce = = 1) {
	pthouad-1 thouad [MAX]:		
	int i.		paint ("allow one philosopher do
	print ("Enter no of philosophers,		
	rean ("Id stotal Philosophis).		der (i=0; is hungay (ount: 1+1)
	printl ("How many are hungry").		for (int j=0: 12 kunguy(ount j+1)2
	scanb ("Id" & hungry count):		points ("P V. d is mailing in -
	for (i=0; ichungay (ount: i++) {		hingaylchilesophers Eg J + 1)
	printl ("Enter philosopher 1. 1 position		in third a matter Course of Coat 1):
	(1 to 1.01) "is 1. doldlahilisophus).		int id = malloc (size of (int)):
	reans ("Id" shungey skilosophen [i])		pthread - create (sthread (1). NULL.
	hungay Philosophers (if:		photosophir, id);
	3		pthouad.jain (thread(17. NULL):
	The Street No. 1 to V. Reg. I		y
	for (i=0: i < total Philosophen; i+1)/		4
	sem-init (& chapatick [17.0.1):		y
	y		actuur o ;
	sem-init (4 mutex, 0.1);		y
	notice (1)		
	paint ("I one can eat at a time")		OUTPUT:
211	point ("2.700 can eat at a time!		cone can eat at a time.
	prints ("Enter year choice:"):	1	2. Two can eat at a dime
	scanf ("Id", echoice);	W	5 Enit
1	1 (choice = = 3)d	50)	
	bruak;	[8]	Enter your choice: 1
	y		Allow one philosopher to eat at
	al (choice = = 3){		any time
	benak:		P3 " granted do eat.
	4		P3 u waiting

	Date Page
	Ph u waiting
ale o	Ph u granted to en
	ps u waiting
V	po u waiting po u granted to eat
73 14	ro is waiting
	La Catalanda aday and

7. Write a C program to simulate Bankers algorithm for the purpose of deadlock avoidance.

```
#include <stdio.h>
#include
<stdbool.h> int
main() {
  int n, m;
  printf("enter number of processes and resources:\n");
  scanf("%d %d", &n, &m);
  int alloc[n][m], max[n][m],
  avail[m]; printf("enter allocation
  matrix:\n"); for (int i = 0; i < n;
  i++)
     for (int j = 0; j < m; j++)
       scanf("%d",
        &alloc[i][j]);
  printf("eter max
  matrix:\n"); for (int i = 0; i
  < n; i++)
     for (int j = 0; j < m; j++)
       scanf("%d",
        &max[i][j]);
  printf("enter available
  matrix:\n"); for (int i = 0; i < m;
  i++)
     scanf("%d",
  &avail[i]); int
  need[n][m];
  for (int i = 0; i < n; i++)
     for (int j = 0; j < m;
     j++)
        need[i][j] = max[i][j] -
  alloc[i][j]; bool finish[n];
  for (int i = 0; i < n; i++)
     finish[i] = false;
  int
  safeSeq[n];
  int
  work[m];
  for (int i = 0; i < m; i++)
```

```
work[i] = avail[i];
   int count = 0;
   while (count <
   n) {
      bool found = false;
      for (int p = 0; p < n; p++)
        { if (!finish[p]) {
           int j;
           for (j = 0; j < m; j++)
              if (need[p][j] > work[j])
                break;
           if (j == m) {
              for (int k = 0; k < m; k++)
                work[k] += alloc[p][k];
              safeSeq[count++] = p;
              finish[p] = true;
              found = true;
         }
      if (!found) {
        printf("sys is not in a safe state.\n");
        return 0;
      }
   }
   printf("sys is in safe state.\nsafe sequence is:
   ");
   for (int i = 0; i < n; i++)
      printf("P%d%s", safeSeq[i], (i == n - 1)? "\n": "->
   ");
      return 0;
}
```

```
5 3
enter allocation matrix:
0 1 0
2 0 0
3 0 2
2 1 1
0 0 2
eter max matrix:
7 5 3
3 2 2
9 0 2
2 2 2
4 3 3
enter available matrix:
3 3 2
sys is in safe state.
safe sequence is: P1 -> P3 -> P4 -> P0 -> P2
```

	classmate
	Date Page 33
	Acadlock Actection
2	timely dead to the
-	thinglude < state of the
	#include sstabool. h.
	int main Of
	int n.m; "bord done done
	prints ("Enter no. of processes &
Ž	scanf ("1.d 1.d", &n. &m);
	as fille as hill for James Bay In the
	int allocation trJlmJ, man InJImJ,
	available (m);
	wool joins a join
	perint ("Enter allocation matrix");
	for lint i = 0; i(n: 1++)
	for (int j=0; f <m; j++)<="" td=""></m;>
	seans ("tel", & allocation tilly 1)
13. 3	Judiana - [[][[] King) Ja
	print ("Enter max matrix:");
	for Cent (=0; i <n; (++)<="" td=""></n;>
	for (int j=0; j=m; j++)
	sean ("Id", & man [iJ[j]);
	Coad award oad dans and
31111	prints ("Enter available matrix")
	for (int: 6=0; icm; 6++)
	seans ("1.d", & arrailable (17));
NA.	man I Chappen I de con la
	At topas

classes	
Date O	CASSMARE Date Age 35_
The second secon	
int need conjimi;	9
1- 1 (nt 1=0: 1 cn; 1+1)	9
100 (cm, (44)	1 0 0
and lifting maximum	at (!found) {
allecation (171)	becak;
	9
beel finishend;	4
ger lint (=0: (2n (+1)	bool deadloch : false;
finish (i) + falu:	for lint 1:0: 1(n: 5+4) }
ent count = 0;	al (! finishii) d
End Court	deadlock = true
while (count (n) !	briak;
bool found = false.	y
for (int 1=0: (in: 1+1) {	y
25 (! 4 cnish tel) d	I was a second of the second
int (:	al (deadlock)
101 (1-0; 12m 1++38	paintl (" Septem in deadlock state");
al (nud [i]) > available (j)	The state of the s
buak;	prints (septem not in cleadlock state)
9 44 4 4 14 14 14	
the and the test	neturn O;
1 (j = m)(7
for (int k=0, kim: kin)	001001:
availablitk J. : allocation [1] [k]	
ginih III - true:	Entu no. of processes & resources
found - time:	3
printl ("Brocen 1.d can finish",);	
count ++;	

8. Write a C program to simulate deadlock detection

```
#include <stdio.h>
#include
<stdbool.h> int
main() {
  int n, m;
  printf("enter num of processes and number of resources:\n");
  scanf("%d %d", &n, &m);
  int alloc[n][m], request[n][m], avail[m];
  printf("enter alloc matrix (%d x %d):\n", n,
  m); for (int i = 0; i < n; i++)
     for (int j = 0; j < m; j++)
        scanf("%d",
        &alloc[i][j]);
  printf("enter req matrix (%d x %d):\n", n,
  m); for (int i = 0; i < n; i++)
     for (int j = 0; j < m; j++)
        scanf("%d", &request[i][j]);
  printf("enter avail resource (%d values):\n",
  m); for (int i = 0; i < m; i++)
     scanf("%d",
  &avail[i]); int work[m];
  for (int i = 0; i < m; i++)
     work[i] = avail[i];
  bool finish[n];
  for (int i = 0; i < n; i++) {
     bool has Allocation =
     false; for (int i = 0; i < 0
     m; j++) {
       if (alloc[i][i]!=0) {
          hasAllocation = true;
          break;
        }
     finish[i] = hasAllocation ? false : true;
  while (true) {
     bool progress = false;
```

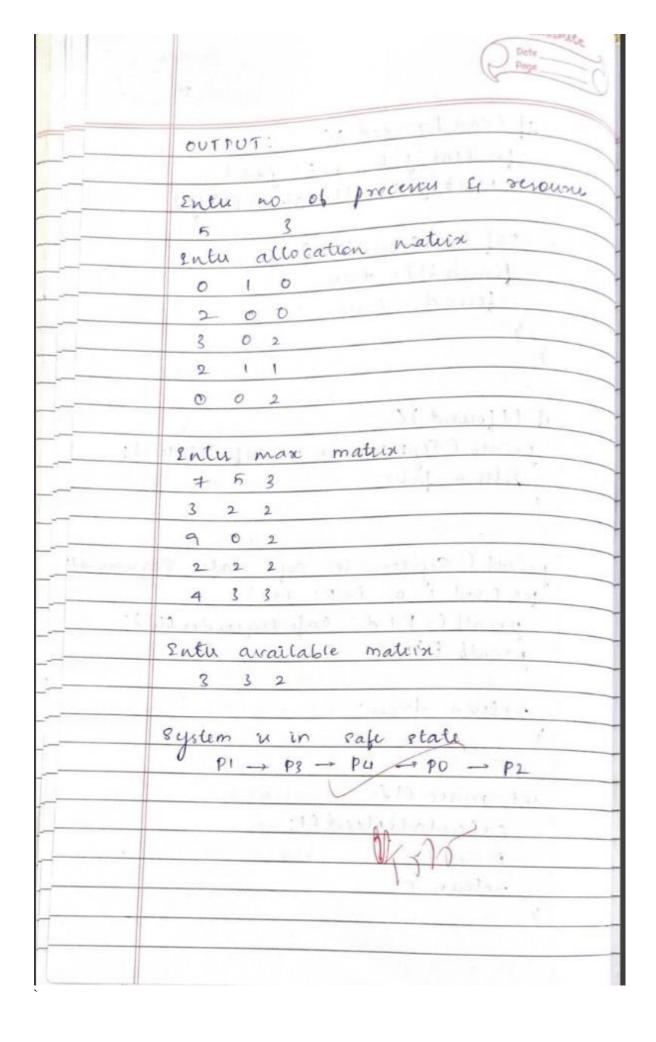
```
for (int i = 0; i < n;
     i++) { if (!finish[i])
        bool canGrant = true;
       for (int j = 0; j < m; j++) {
          if (request[i][j] > work[j]) {
             canGrant = false;
             break;
        }
        if (canGrant) {
          for (int j = 0; j < m; j++)
             work[i] += alloc[i][i];
          finish[i] =
          true;
          progress =
          true;
     }
   }
  if (!progress)
     break;
printf("\nDLD result:\n");
bool deadlock = false;
for (int i = 0; i < n;
  i++) { if (!finish[i])
     printf("process P%d is deadlocked\n", i);
     deadlock = true;
  } else {
     printf("process P%d is not deadlocked\n", i);
   }
if (!deadlock)
  printf("\nno deadlock detected in the system.\n");
  return 0;
  }
```

```
enter num of processes and number of resources:
enter alloc matrix (5 x 3):
010
2 0 0
3 0 3
2 1 1
002
enter req matrix (5 x 3):
000
2 0 2
0 0 1
100
002
enter avail resource (3 values):
000
DLD result:
process P0 is not deadlocked
process P1 is deadlocked
process P2 is deadlocked
process P3 is deadlocked
process P4 is deadlocked
```



	Page 37
	Dreadlock Avoidance
	12 (143 - 3 - 3 - 143 - 43 - 143 - 43 - 143 - 43 -
	ticnclude < statio. hs
_	Hinclude & stabool Ry
	# define PS
	# define R3
	int available CKJ={3,3,23;
	int max (PJSRJ = 6
	27. 5. 3 4,
	63, 2, 25, 0 Januar Jan
	19.0.24,
	£ 2. 2. 2 5. 3 5 5 5 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
	dals: syntimus e (il degr.
	y : / / / / / _ / _ / _ /
	La company de la granda de la planta de la collección de
	int allocation [P][P]={
	€ 0.61. 04 ALS 10=3 100 1.00
	62.0.04. (CD) ABOUT 1) I
	23, 0; 22, sea (100) land
	d 2. 1. 192 . 1 0 - 1 Jan 1 221
	20.0.24. [3] 53.01
	y; estable Birish Vicas
	Annual Annual
	int need [PJ [R J;

void calculate Need () ! for (int 5=0: 12P; 1+1) ! for (int 5=0: 12P; 1+1) ! for (int 5=0: 12P; 1+1) ! nucl (i) (j) = man (i) (j) - alteration (i) (j); safe for fround +1 found = 4 nu. bed is safe () (int north [8] = [1 false 5; int safe son [8]: int sound = 0; for (int 6=0: 62P; 6+1) ! print (System in safe (int 1=0: 12P; 1) ! bed found = false; for (int 6=0: 62P; 6+1) ! north () { false for for (int 1=0: 12P; 1) ! print (System in safe (int 1=0: 12P; 1) ! print (System in safe (int 1=0: 12P; 1) ! print (System in safe (int 1=0: 12P; 1+1) ! p	classmate Size \$9
for (int j=0. jek; j++)/ peol (int j=0. jek; j++)/ peol issafe()(int safe seq [P]; int safe seq [P]; for (int i=0; i=k; i++)/ peol issafe()(for (int i=0; i=k; i++)/ peol issafe(); for (int i=0; i=k; i++)/ peol issafe(); for (int i=0; i=k; i++)/ printl(int i=0; i=k; i++)/ printl(int i=0; i=k; i++)/ printl(int i=0; i=k; i++)/ printl(int i=k; i=k; i++)/ printl(int i=k; i=k; i+k)/ printl(int i=k) printl(i	
for (int j=0; j=k; j=k) nud (i)(j) = max (i)(j) - allocation(i)(j); safi seq fround + 2 finish (i) = true; found = drue, bool insafe (i)(int more kiff; bool finish (f) = i falu b; int safe seq [p]; ont count = 0; for (int i=0; i=R; c+1) more (i) = available (i); bed found = falu; for (int t=0; c=p; r+1)(if (i) finish (i))(printl ('Pid', safe pr	
alleration[1][]; safe sea frount + 2 finish [1] = true; found = draw, bool insafe () (int noork the J: bool finish [1] = italie b; int safe sea [1]: for count = 0: for (int i=0: i=R: c+1) roor [i] = available [i]: print[(System in so for (int i=0: i=R: c+1) bed found = false; for (int i=0: i=p; r++) f: x[(! finish [i]) f x[(! finish [i]) f y catture draw; for (int j=0: j=R: j++) f Af (need [i][]) roork [i]] f int main () f can Provide = false; break, lessafe();	C. J++/
safile flound 1.) y finish (1) = true; found = drue, bool issafe()(int nearly [8]: bool finish (18]: if also 5; int safe seq [8]: for count = 0; for (int 1-0: 1-8: 1+1) bool finish (18]: printl (System not to guillen false; printl (System in sa for (int 1-0: 1-8: 1+1) bool found = false; for (int 1-0: 12): (1+1) if (! finish [1]) { printl (P!d , safe printl (In); for (int 1-0: 12): (1+1) for (int 1-0: 12): (1+1) for (int 1-0: 12): (1+1) for (int 1-0: 12): (1-1) for (int 1-0: 12	accos (1) cfs,
finish (1) = true; found = drue, found = finish (F) = i false b; int safe seq [F]: for count = 0; for (int i=0: i=R: i+1) recribil = available (i); for (int i=0: i=R: i+1) head found = false; for (int i=0: i=P; i+1) for (int i=0: i=R: i+1) for (int i	1 : 1 :
bool issafe() { int work R]: bool finish P]: false j int safe seq [P]: int count = 0; for (int i=0: i=R: i+1) work i] = available i]; bool feund = false; for (int i=0: i=p; i+1) to (int i=0: i=p; i+1) for (int i=0: i=R: i+1) for (int i=R: i=R: i+1)	
bool issafe()(int noork [R]: bool finish [P]: ifalse; int safe sag [P]: ont count = 0; for (int i=0: c=R: c+1) bool fill: available [I]: printl (System in sa for (int i=0: i=p; i printl (P!d; safe printl (System in safe printl (P!d; safe printl (System in safe printl (P!d; safe printl (P!d; safe printl (System in safe printl (P!d; safe printl (System in safe printl (P!d; safe printl (P!d; safe printl (P!d; safe printl (System in safe printl (Sy	
int work [R]: bool finish [P]: [fabre 5; int safe seq [P]: ent count = 0; for (int i=0: c=R: c+t) bool fill: available [[]: point! ("System not in situal fill: point! ("System in sa fer (int i=0: i=P; i) point! ("Pid", safe point! ("In"), for (int i=0: i=P; r+t)? yo (int j=0: j=R j+t)? yo (int j=0: j=R j+t)? yo (int j=0: j=R j+t)? yo (int main ()? can Provide = fabre: break, ssafe();	
int work [R]: bool finish [P]: [fabre 5; int safe seq [P]: ent count = 0; for (int i=0: c=R: c+t) bool fill: available [[]: point! ("System not in situal fill: point! ("System in sa fer (int i=0: i=P; i) point! ("Pid", safe point! ("In"), for (int i=0: i=P; r+t)? yo (int j=0: j=R j+t)? yo (int j=0: j=R j+t)? yo (int j=0: j=R j+t)? yo (int main ()? can Provide = fabre: break, ssafe();	
bool finish [P] = [false 3] int safe seq [P]: cnt count = 0; for (int i=0; c=R: c+t) bool fill = available [I]: point! ("System in sa for (int i=0; i=p; i point! ("System in sa for (int i=0; i=p; i point! ("P!d", safe point! ("P!d", safe point! ("In"), for (int i=0; i=p; r+t)? x1 (! finish [I])? proced can proceed = true; for (int j=0; j=R; j+t)? A (need [I][]) > roock [J]]? can proceed = false: calculationed (1; break,	
int safe seq [P]: cnt count = 0; for (int i=0: i=R: c+1) rect [i] = available [i]: print[(System in sa for (int i=0: i=P: i print[(Pid; safe print[(Pid;	
cont count =0; for (int i=0: i=R: i+1) point (i) = available (i); point (i) = available (i); point (i) = available (i); point (i) = 0: i=P; i point (i) =	in cale state);
for (int i=0: i=R: c+1) real (i) = available (i): point (System in sa point (System in sa point (int i=0: i=P; i point (i	
real (1) = available (1); print (System in Sa pos (int 1=0, 1=p; t pos (int 1=0, 1=p; t pos (int 1=0, 1=p; t pos (int 1=0, 1=p; t+1) If (! finish (1)) pos (int 1=0; 1=p; t+1) pos (int 1=0; 1=p; t+1) for (int 1=0; 1=p; t+1) int main () can proved = fake; break; break; scalculationed (1;	
real (1) = available (1); print (System in Sa pos (int 1=0, 1=p; t pos (int 1=0, 1=p; t pos (int 1=0, 1=p; t pos (int 1=0, 1=p; t+1) If (! finish (1)) pos (int 1=0; 1=p; t+1) pos (int 1=0; 1=p; t+1) for (int 1=0; 1=p; t+1) int main () can proved = fake; break; break; scalculationed (1;	
nehile (count < P) { nehile (count < P) { point (P 1 d , Safe	ale state sequencis);
bed found = Jah; for (int t=0; i2p; i+1) \(\) Al (! finish [1]) \(\) bad can Proceed = true; for (int j=0; j=R: j++) \(\) Al (need [1][] > morek []] \(\) can Proceed = Jahe; break; break; lssak();	
bed found = fals; for (int t=0; i2p; t+1) \(\) **I (! finish [1]) \(\) bud can Proceed = true; for (int j=0; j=R : j++) \(\) **I (need [1][j] > novek [j]) \(\) (an Proceed = false; break; break; \$ 155a[c();	sequence Cid);
ta (int =0; i2p; r++)? *** *** *** *** *** *** *** *** *** *	
paci can Proceed + true; for (int j=0; j=8: j+t)? A (need (1)(j) > rooth (j))? (an Proceed = jake: break; calculate Need (1;	
A (need (1317) > novek (13) & int main () & (an Provid = falk; calculate Need (); break, is safe();	
A (need 13/1) > morekey) int main () L can proceed = false; break, calculate Need (1;	THE TAXABLE IN
can proceed = false; calculate Meed (1; break, issafe();	CALLETT ST.
break, issafe();	
y 5,210,11 bases has setum o	
7	



9.Write a C program to simulate the following contiguous memory allocation techniques

```
a) Worst-fit
b) Best-fit
c) First-fit
#include <stdio.h>
struct Block {
  int size;
  int
  is_free;
};
struct
  File {
  int
  size;
};
void firstFit(struct Block blocks[], int n_blocks, struct File files[], int n_files) {
  printf("\nfirst fit alloc\n");
  printf("file no\tfile size\tblock no\tblock size\tfrag\n");
  for (int i = 0; i < n_files; i++) {
     int allocated = 0;
     for (int j = 0; j < n_blocks; j++) {
        if (blocks[j].is_free && blocks[j].size >= files[i].size) {
           blocks[j].is_free = 0;
          printf("%d\t\t%d\t\t%d\t\t%d\t), i + 1, files[i].size, j + 1,
blocks[j].size, blocks[j].size - files[i].size);
           allocated = 1;
           break:
        }
     if (!allocated) {
        printf("file %d cannot be allocated\n", i + 1);
     }
   }
}
     if (worst_block != -1) {
```

```
blocks[worst_block].is_free = 0;
       printf("%d\t%d\t\t%d\t,i+1, files[i].size, worst\_block+1,
blocks[worst_block].size, max_fragment);
     } else {
       printf("file %d cannot be allocated\n", i + 1);
  }
}
int main() {
  int n_blocks, n_files;
  printf("enter the number of blocks:
  "); scanf("%d", &n_blocks);
  printf("enter the number of files:
  "); scanf("%d", &n_files);
  struct Block
  blocks[n_blocks]; struct File
  files[n_files];
  for (int i = 0; i < n_blocks; i++) {
     printf("enter the size of block %d: ", i +
     1); scanf("%d", &blocks[i].size);
     blocks[i].is_free = 1;
  }
  for (int i = 0; i < n_files; i++) {
     printf("enter the size of file %d: ", i +
     1); scanf("%d", &files[i].size);
  }
  firstFit(blocks, n_blocks, files, n_files);
  for (int i = 0; i < n_blocks; i++) blocks[i].is_free = 1;
  bestFit(blocks, n_blocks, files, n_files);
  for (int i = 0; i < n_blocks; i++) blocks[i].is_free = 1;
  worstFit(blocks, n blocks, files, n files);
  return 0;
```

}			

```
enter the number of blocks: 5
enter the number of files: 4
enter the size of block 1: 100 enter the size of block 2: 500
enter the size of block 3: 200
enter the size of block 4: 300 enter the size of block 5: 600
enter the size of file 1: 212
enter the size of file 2: 417 enter the size of file 3: 112
enter the size of file 4: 420
first fit alloc
file no file size
                                                                  frag
                            block no
                                               block size
        212
                                                                   288
                                               500
         417
                                               600
                                                                   183
3 112 3
file 4 cannot be allocated
                                               200
                                                                  88
est fit alloc
file no file size
                                               block size
                            block no
                                                                  frag
        212
                                               300
                                                                  88
         417
                                               500
                                                                  83
                                                200
                                                                  88
         420
                                               600
                                                                   180
worst fit alloc
                            block no
                                                                  frag
file no file size
                                               block size
                                                                  388
         212
                                               600
         417
                                                500
                                                                   83
         112
                                                300
                                                                   188
                            4
file 4 cannot be allocated
```

100000000000000000000000000000000000000	Classmate Date Page 41
	LAB 6 :
	and the law should still be a first
	# snclude (statio. hs
	th Extendia Commist) he is
	struct Block &
1	int size;
	ent allocated;
	STREETS OF FIRST DAY STORES
1	A distribution of the state of the form
	stouch file f
	int rize;
	int block-no;
	y; -
	Mark Charles Annual Christian Committee
	roid reset Blocks Cotout Block block 17.
	e come the ment of
	for (int c=0; i=n; c+1)(
	blocks [id. allo cated = 0;
	y
	4 and and a sumper filtred there
	word first fit (struct Block block 1).
1	int n-blocks, struct File feltell.
	int n-10001
	print ("In + Memory Management Scheme - Fint Fet");
	printh ("File - no: 14 File - size 1 & Block-no:
	plints ("Fill-No. 14 + 111 - 112)
	At Block-street:"); and hard and
	A marie when the most of any of

	Sec.
classmate	classrate Dirth
Charles C	15 my 48
for (int 1=0, 12n-files; 1++)}	af (block 1) allo cated 43
	blocks (g) size >= film [1] size) (
- N - D - D - D - D - D - D - D - D - D	il (best &dx == -11) blocks 13). 1726 c
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	block that adal cixe Il
block (1. sin >= files (1. eize) /	best odx = j;
block 113. Sieck - no = j+1;	4
block (3) allocated - 1;	4
print (1) de 1, de	1 (best 3dx1=-1){
print (1012. 91, blech (3.012)	block [best adad. allo cated =1;
i+1. fdu 117. 112. jan 200 41.6(2)	fills lid black - no a bestada + 1;
bacak;	printl(1.d 1+ 1.d 1+ 1.d 1+ 1.d 1+ ",
g and the	en block (best 2d x J. secc):
- '	selved
1 (filly rid, block - no = = -1)/	psint ("1-d1+1-d1+1-d1+1-d1+", 1+1,
1 1 d d - 1 d t - 1 d	\$du 111.51x1);
1+1, rilationery	9
4 3/33 3 313 301 31	4
9 3010 10 11 1014	S STA
	printf ("In Worst +d");
roid best Fit Cotruct Block blocks	ent my CMAXJ; factint 1=0; 1 <n;< td=""></n;<>
ent nublocks, struct File John 19	n block, s++);
int n-folul	
nine ("Memory Management : Scheme.	witid: blocklid;
5 . 5 10 1	allocate (Nf. nblock, process, n
resall ("File-no 14 File-rize);	procen);
peints ("File-no 1+ File-rize");	action 0;
for (11) (=0: i=n-files; 1+4){	9
int butled x = -1;	
Jos (rat j=0; j=n-blocks: ++){	
45 114 1 1 1 1 1 1 1 1 1 1	

111		classmate
		Dafe
	7.5	A ST
	N.	1 (1 (0 (1 (1))))
	OUTPUT:	n - 6 (h adam) 4
1	GXPLDTMD - S - S	SE PROMET IN THE
5 (5)		
36.0	Proces no de desid	6 .
	1 212	5 1
	2 417	2
	3 112	Net allokated
	4 426	Land Today of the
	the Anthonoration of the	Ld (3) relat
_	Best Fit:	A STATE OF THE STA
	+11/2 +/ +/ +/ 21/25 1-)	1 1 2 1 4 - 3
	2-(337), (3417)	3 11 1
	3 (12	
(1+)	4 4 426	The same of the same
-	CHETTALITUME	,
-	Worst Fit:	5
	1 212	2
-	3 112	
		read and through
	THE VEHICLE AND THE PARTY OF TH	Not allo cated
	and the Catal	×358 H
		old - Digot
	THE PROPERTY OF THE CENT	A LIBERT OF
1.00	maxture of deleter	310
		a white
		14 1857-191
	Mark College Fill	

10. Write a C program to simulate page replacement algorithms of fifo, LRU and optimal

```
#include <stdio.h>
#include <stdbool.h>
#define SIZE 7
#define FRAMES 3
int findLRU(int time[], int n) {
  int min = time[0], pos = 0;
  for (int i = 1; i < n; i++) {
     if (time[i] < min) {
       min = time[i];
       pos = i;
     }
  return pos;
int findOptimal(int pages[], int frames[], int n, int index) {
  int farthest = index, result = -1;
  for (int i = 0; i < FRAMES; i++) {
     int j;
     for (j = index; j < n; j++) {
       if (frames[i] == pages[j]) {
          if (j > farthest) {
             farthest = j;
             result = i;
          break;
     if (j == n) return i;
  return (result == -1) ? 0 : result;
}
void fifo(int pages[]) {
  int frames[FRAMES], front = 0, count = 0, faults = 0;
  printf("\nFIFO: ");
  for (int i = 0; i < SIZE; i++) {
     bool hit = false;
```

```
for (int j = 0; j < count; j++)
       if (frames[j] == pages[i]) hit = true;
     if (!hit) {
       if (count < FRAMES)
          frames[count++] = pages[i];
       else {
          frames[front] = pages[i];
          front = (front + 1) % FRAMES;
       faults++;
  }
  printf("Page Faults = %d\n", faults);
void lru(int pages[]) {
  int frames[FRAMES], time[FRAMES], count = 0, faults = 0;
  printf("LRU:");
  for (int i = 0; i < SIZE; i++) {
     bool hit = false;
     for (int j = 0; j < count; j++) {
       if (frames[i] == pages[i]) {
          hit = true;
          time[j] = i;
          break;
     }
     if (!hit) {
       if (count < FRAMES) {
          frames[count] = pages[i];
          time[count] = i;
          count++;
       } else {
          int pos = findLRU(time, FRAMES);
          frames[pos] = pages[i];
          time[pos] = i;
       faults++;
     }
  printf("Page Faults = %d\n", faults);
}
void optimal(int pages[]) {
  int frames[FRAMES], count = 0, faults = 0;
  printf("Optimal: ");
```

,

```
for (int i = 0; i < SIZE; i++) {
    bool hit = false;
    for (int j = 0; j < count; j++) {
      if (frames[j] == pages[i]) {
         hit = true;
         break;
       }
    if (!hit) {
      if (count < FRAMES)
         frames[count++] = pages[i];
         int pos = findOptimal(pages, frames, SIZE, i + 1);
         frames[pos] = pages[i];
      faults++;
  printf("Page Faults = %d\n", faults);
}
int main() {
  int pages[SIZE] = \{1, 2, 3, 2, 1, 4, 5\};
  fifo(pages);
  lru(pages);
  optimal(pages);
  return 0;
FIFO: Page Faults = 5
LRU : Page Faults = 5
Optimal: Page Faults = 5
```



	FIFO. LRU and Optimal:
-	
-	stindude < stdio. Rs
	int search lind key, int frame [], int f) &
	for (int t=0; izf; 144)
	if (frame [i] == key);
1	seturn it
	retun -1;
	A september of the same of the
-	ent find LRU lint time []. int f) {
	int min = timeloJ. pos = 01
	for (int t=1: 12f: (++))
	if (time (i) < min) min = time (i).
	· Walling a grace for the application por a c;
	. La cteun pos;
	St.
	the second contribution of the contribution of
	ind predict lint pages 13. int frame 13.
	ent no ent tdx, ent f) L
	int ses = -1, far = idx;
	for lint 1=0: icf: c++ W
	int j:
	ta (= 10 x; 12n; 1+1)
	il (1> far) far = j. res = i;
	break;
	y the install

	Euro Enge	Classaste Prop. 43
	if (jezn) returni;	const chas " names (3 = 1" "FIFO" "LRU".
	100.000	"optimal" 9:
	neturn (201 == -1)? 0 : 201;	prints ("1.3 Page faults: 1.21".
	9	names (types). faulls;
1 40		40 1 2 1 2 1 2 1
	and simulate cent pages 12. ent n. ent f. ent type 22 int frame (f). time (f). count = 0.	A common la sur arriva
	int s. int type of	int main () (
	int frame If J. time If J. count = D.	int n. f. algo
	faulu =0	prints ("Enly no of pages"):
	for lint t= 0: 121: 1+1) frames(1)=+	rang ("Id", an)
	, ,	ent page ind;
	failint t=0: cin; t++ X	
	int idx = seaich (pager [1], frame)	preat ("Entu page reference string"):
	4 (rdx == -1) (jac (int i=0: izn; i+1)
_	ent sep = 0:	ent page (n);
	if (type == 1) rep = fauth 1. f;	the page that
		paint (" Enter no of framer");
_	'elxel (type == 2) rep = (fault 24)?	Sant (18 34)
	faults;	print ("I FIFOIN 2 LRUIN 3 optimal in
	find LRU (time, f);	2 nter choice 1:
_	elseif (type== 3) sep = (faults ex)?	scant ("1.d". Lalgo);
	faulti: predict (pages frames . n. (41):	ib (algo >=) 4d algo e= 3
	frame treps is pages (1);	simulalle (pages, n. f. algo):
	if (type == 2) time (rep) = tount;	che che
	Jaulu++	paint ("Invaled choice"):
1	yelve of thype == 212	setuno;
	time (rdx) = count;	9
	9	
1	count ++;	
1	5	
1		

