```
DATE 08.15 24
             Lab-1
10 Wille a c program to simulate the
   following non- pre-emptive CPU scheduling
   algorithm to find TAT & WT.
   -> FCFS
   #include <stdio. h)
   # define MAX 10
   void fets (int n. int at(), bt()) &
    int CE[MAX];
    int dat [MAX];
   int wt [MAX]:
   int total . wt = 0;
                           - water many
   Pot total tat : 0:
   Part corrent. time = 0;
   for (int 120; in; i++) f
     Ct [i] = -1;
   for (int int 1:0: icn; i++) }
     3f (corrent time < at [i]) }.
      2 correct - time = at [i];
      Ct[i] = current time + bt[i];
     Current - time = (+[?];
   for (int ?20; ixn; ?++)?
     tat [i] = c+ [i] - a+ [i]:
    total tal + = tat[i];
  for (int 1:0; in; i++) &
     wt[i] = tat[i] - bt [i];
    total_we t = wt[:];
```

```
printf ("In Process | t Arrival Time | t
       Completion Time It Turn Around Time It writing
          Fine of Ji with wing
   for (int ico; ikn; i++) }
     printf ("%dlt %d/t %d/t/F%d/+/+%d/t/t
            % od \n", 3+1, at[i], bt[i]
             Ct[i], tat[i], wr[i]);
withing you or other with street there's series a street
   printf ("In Average waiting Home: %.26",
          (float) total-wtln);
   printf ("In Average turn wound time: % 2 f
           (float) total - tal (n);
   int main () to the land where
   int niiste built because the
   print ("Enter number of processes:");
   Scanf ("%d" &n);
   ink at [n] be [n];
   printf ("Enter the asurival time: "In");
   scoof ("ord", fattilli
  point f ("Enter the burst time : \n");
   for (i=0; i2n; i++) {
    Scoof ("%d", & bt(i7)i
   fassin, at, bt); is and to the
    Putura 0;
```

16 7 so A to bear the

Output:
The state of the s
Fater the number of processes: 4
Fatar the agrical time:
0 1 5 6
Enter the burst time:
Process Arrival Burst Completion Turnaround Whiting
Process Arrival Burst comprehen Time Time Time Time
2 1 2 4 3 1 3 1 3 5 3 5 8 3 0
4 6 4 12 6 2
Average waiting time: 0.75
Average turnoround time: 3.50
Tivelage Tomoborno Time
STF - Non-preemptive
And the second
include < stdio-h>
include <s< td=""></s<>
define MAX 10
void SiFNonPreemptive (int n, int at (), by be()
int ct[max];
int tatemax];
int wt [MAX];
int It [MAX]
int total-wt 6 = 0:
int total - tat = 0;
int completed = 0;
int current time = 0;
fint shortest - job = 0;
int min-bt = 9999;
int is_completed [MAX] = {0};

```
for (int 100; inn; in+) }
  YE [i] = be [i] :
while (completed en) !
 for (int iso; ico; i++)/+ . " "
   ? (at (i) < = corrent time df .
        rt[i] < min-bl- 24 ! is-comple
            - 4ed (17) }
 Shortest-job = i;
min-bt = xt CiJ;
             the section as
Tt [shortest = job] or ; it was the the
?f (it [shortest - job] = = 0) }
  Completed ++;
  min_ br = 9999;
  is completed [shortest job ] 21:
  (+[shortest-job] = (virent-time + 1:
  tat [shortest-job] = Cf [shortest-job] -
                  at [shortest-job]:
  total tat it = rtat [shortest. job];
  wt[shortest-job] = tat[shortest-job] -
                   bl. [shortest job];
  ?f (wt [shortest-job] < 0) }
   bot Cshortest - job ] = 0; 11
    total wit t = wr [shortest job];
  4 correct time + + is
printf ("In Processit Arrival Time It Burst Time
    It Completion Time It Turn wound Time It
      utiling Time In);
for (int 1:0; icn; 1+4) }
 printf("%d/+ %d/+/+ %d/+/+ %d/+/+ %d/+/+ %d/n")
```

at [i), bt (i), ct (i), tat(i), wt(i)); printf ("In Average waiting time: %. 2 6" (float) total wr/n); printf ("In Average turnazound time: %. 16" (float) total tat/n); int main () 1 int n, i: printf ("Enter the no of processes:"); scant ("ord" fo); int at [n], bt [n]; printf ("Enter the arrival time : In"); for (int iss; ich) it +) } Scanf ("%d", Lot [i]); printf ("Enter the burnt time ! \n"); for (100) gen! i++)}. Scanf ("skd", f-bk[i]):... Sj F. Non Picemptive (n; at, bb); return of the state of the state of Output! Enter the no of processes: 4 Enters or o / enter the arrival time: Enter the burst time:

PAGE NO: 6

				DATE:	
Process	Arrival	Burst		Tunoround	Waiting
1000	Time	Time	Time	7 me	Time
1	0	6	CA 9 100	1691 4 H	3
2	0	8	2441	24 - 1	16
3	100	4.77	16 16	3:16 Blog	9
u	6	3		13 10	0
			- STYAM	100 403	
Average	Waiti	ng tin	no! A an	3 400 45	
Average		onand	time: 13	J 50 45	
0			Av . 300	Inter For	
			70 form	1070 July	
			and basis	more than	
			et with the		
			Designan		
					-
1			(17 sx c)	5 140	

100 24 20 20

and the state of

```
DATE (5 | 5 | 2 9
 void sif- pre-emptive (int n, int at (), int be())
 not wt [MAX];
 int is-completed [MAX] = {o};
 for (int i=0; i<n; i++) i
 int shortest-job = - 1;
for (int 1:0; ikn; i++) 1
  if (at [i) <= current time &f xt[i] <
                 min_bl- 44 rt[i] >0) {
      Shortest - job = i' ;
     min-bt = rt (i);
if (shortest - job = = -1) }
```

· Continue; 1+ [Shortut-job] -- ; if (+ [shortest - job] = = 0) {

current _ time + + ;

-> STF (Preomptive).

define MAXIO

not CT[MAX];

Pat tat [MAX];

int at [MAX]:

int total_wt : 0;

int total -tat : 0;

int completed : 0;

int current - time = 0;

* [[] = b+ [i];

while (completed In) {

int min bt = 9999;

include < stdio. b)

Completed + +; ct [chartest job] = corrent time + 1; tat (shortest job) = ct (shortest - job) at (shortest - job); total. tat + = tat [shortest - job] i wt (shortest - job) = tat (shortest - job) bt [shortest - job]: Pf (wt [shortest-job] < 0) wt [shortest-job] = 0: total - w+ + = w+ Cshortest - job]; PS - completed [shootest-job]=1: current - time + +; printf ("InProcess) + Arrival time + Bunt time + Completion Time It Turnaround Time It Waiting Time In"); for (int 100; ikn; i++) t printf ("%d/+ %d/+ /+ %d/+/+ %d/+/+ %d It It "ad la" it, at (i) bt (i) (c+Ci) tat(i), w+(i)); print ("In Average weiting time: %.2F",
(float) total-w+/0); printf ("In Average turnoround time: %.2F" (float) total - tat (n); int main () { int n, is print ("Enter the number of processes:"); Sanf ("%d", fn); int aten), brenj: printf ("fater the arrival time: \n"); for (1:0; ikn; i++) }

Scoof (" " od", & bt-[;]); Printfl"Enter the burst time 'ho"); for (1=0; ikn; 1++) { Scanf ("od" f. ble [i]); St-preemptive (n, at, bt); return of output: Enter the number of processes: 5 Enter the arrival time: ·2) 1 H: 0 2 1/1 1/1 1/1 Enter the buist time! 5 1 :6 .3 Processes Arrival Burst Completion Turnaround Waiting Time: Time Time Time Time 1 1 1 2 1 1 1 31 20 100 1000 5 1000 11 10 3 4 1 4 6 16 , 10 5 2 1 3 3 5000 A V 3500 Average Waiting time: 3.40 Average Turnaround time: 6.60

17

1.277 3

```
Write a C program to simulate the
  following cou scheduling algorithm to
  -> Priority (pre-emphire)
  -> Round Robin
Pound Robin was of the sale of the
  # include «Staio h)
  # define MAX 10
  void round robin (int n, int b+C), int
                     quantum) {
  int wt [max] = log;
  int tat [MAX] : toy;
  int remaining-bl- [MAX];
  int total - wif = 0, total - tel = 0;
  int time = 0;
  for (int 100; icn; i++) 1 {
   remaining - bt- (i) = bt- (i);
  while (1) }
   int done : 1;
   for (int i=0; izn; i++)?
     if Cremaining - bt (i) > quartum) of
       done : pi
       2f (remaining-bl- (i) >quantum) {
         time + = quantum:
         remaining _ bt[i] -= quantumi
       3 clse {
         time to remaining bt (i);
         wt (i) = time - bt [i];
         remaining - bt [i] = 0; 9
```

```
3f (done == 1) break;
  for (int i=o; icn; i++) }
   tat Ci] = bt Ci] + wt Ci];
   total-wf += wx [i];
   total-wt += tat (i);
  printf ("In Process It Burst Time It Waiting
    Time It Turnaround Time In"Ji
  for (int i=0; izn; i++){.
  +at [] = bt [] + wt [];
  total -wt + = wt [i];
   total - tatt = tat[i];
 prints ("InProcesses t Burst Time It waiting
   time I trumpsound time (a:):
 for (int 1:0; ixn = 1++) {...
    printf("%d|+%d|+|+%d|+1+%d|n"
       it i bt[i] wt[i], tat[i]);
printf ( In Average waiting time : " & a f"
              (float) tat-wt/n);
printf ("In Average turnaround time: % af
  (float) total_tat(n);
int main () !
int o quantum:
prints ("Enter the number of procuses:");
scanf ("%d" +n)?
int : bt:[MAx]:
printf ("Enter Burst Time for each process:
    (n');
for (int 1=0; izn; i++) }
```

PAGE NO: 12 DATE:

printf ("process %d"; ;+1); scanf ("end", & bt(ij); printf (Enter the six of time blice Scant ("ord", 2 quantum); ground-brobin (n. bt, quantum); greturn 0; output: Enter the number of processes: 3 Enter Bust Time for each process: Process 1: 24 Process a: 3 Process 3: 3 Enter the size of time Sice (quantum): 3 Process Buryl- Time Wasting Time Turnoval Time Average Daiting time! 5 Average turnationed time: 15

```
PAGE NO 15
-> Priority (Pre-emptive)
# include < stdio.h>
void sont (Pot prids), int pl], int all
          int bt[] int b(], int n)/
 int min= p[o], temp=0;
for (int ?=0; 120; 1++) {
   min= p[i];
   for (int j=$ = j < n; j + +) {
   it (p[i] < min) }
      temp = at[i]i
       at (i) = at (j);
       temp = bt(j)i
       bt[i] = bt[i];
       bt (i) : temps
        temp = b[j]i
       b[j] b[i];
        bC] = temp;
        temp = p[j]i
        pli] = pli];
        p[i] = temp i
         temp = proc. id [i];
        proc - ?d [i] = proc - ?d [j];
        proceed [j] =/temp;
( ) nion biov
int n (=0;
printf ("Inter number of processes:");
scent ("%d" 4n);
int proceident, at ent brent, etent, tatel
```

PAGE NO: 14.

: [a]q , [a]te , [a]d , [a]m , [a]tw double org trat = 0.0 ttat = 0.0. for (int 1=0; i<n; i++) { proc_ (d(i) = (+1; 9 prints ("Enter Priorities: \n"); for (int 1=04 120; 1++)/ Scoot ("062", 4p(17); prints ("Enter arrival times: In); for (int 7:0; i2n; i++)
scan(("%d" fat (i)); il'al runt loud star I toured For (30+ 120; 120; 1+4) Scanf 10%d & bt (i)); b(i) = b(i); m [i] = -1; Y((i) = -); Sood (proc. ?d , p. at, bt, b, n); Pot count =0, prio = 0, priority = p[o]: while (count <n) of for (int 125; icn: 1++)[? Elat [7] <= C & P[7] 7= priority && b[i] 70' 24 m[i] [=01) } x=?; priority = p(i); 26 (PCX) 20) 1 if (at(x) = ==1)

```
PAGE NO: 15
     91+ [x] = (-a+[x];
    b(x) --;
    (++1
  if (b(x)==0) }
   Count +++;
   C+[x] = C;
    m[x] = 1;
    while (x>=1 & & b(x] ==0)
    ¿ privrity = p[--x];
  ? ( ( count == n)
   DIPOIS:
 for (int 1=0; icn; i++) )
 ((1) to - ((1) = ((1) tat
 for (int 1:03 izn; 1++)
 w+ [i] = tout [i] - bt [i];
 Print [" PIDL+ Priority L+ AT L+ BT L+ CT) + FAT/+
                           WILL BILL ");
 for (int i= 03 ikng i++)
 brintel. bary 1 + 291+ 1+291+1+291+291
           + %d/+ &d In", proc_id[i] pti] at[i]
     bt[i], ct[i], tat[i], wt[i], x+[i]);
 for (int 1 =0; icn; 1++) {
   + tal + = tot[i]
   two + = wt[i];
avg_tat = ttut / (double) ni
avg - wt = twt/ (double) n;
printf ("In Average turnasand Time: 981 fms) "
                             Ovg-tat );
prints ("In Average waiting Time: 061 fmila", augustie
```

PAGE NO: 16

output:

Enter number of processes: 4

Enter priorities:

10 20 30 40

Enter carrival times:

0 1 2 4

Enter burst times:

5 4 2 1

PTD Priority AT BT CT TAT WT RT

P1 10 0 5 12 12 7 0

P2 20 1 4 8 7 3 0

P3 30 2 2 4 2 0 0

P4 40 4 1 5 0 0

Average waiting time: 2.50 ms

```
-> SA Priority (Non Preemptive)
# include < stdio-h>
void soit (int proceded ) int pc), int atg
            int bl (), int n) }
int min = p[o], remp = 0;
for (int 1:03 1201 it 4) }
min = p[i]:
for (int j=i: i<n; j++) }
  if (ptj) < min) !
    temp = at [i];
     attij = attiji
      temp = bt (j);
bt (j) = bt (j);
      bt (i) = temp;
      temp = p[j]:
       وزآع و دراع
       pCi] = temp;
       temp = proc id(i);
        procid(i) = procid(j)i
       2 procidej = temp;
youd main() }
int 0, (=0;
printf ("Enter number of processes:")
Scanf ("60" 8n);
int proc-iden), at Enj, bt Enj, ctCnj, tatEnj,
         wt Cn), mcn), gitch], pcn);
double avg tat = 0.0 + tot = 0.0 avg wt=0.0,
```

```
fox ( int 1 = 0; (20; 1++) {
proceid[i] = i+1;
meijess
print( Enter priorities In );
for (int 100) ((n: 14+))
 Scant ("%d", 4p[i]);
 print ( Enter currival times 10);
 for line 1=0; 120; 1++) $
 Scanf ("bd fat [i]);
 prints ("Enter bout times In");
 for (int 1:0) icn; it+) {
    scant ( "d", 4 ht[i]);
     m() = -1;
    Y Ei Jani
Sort (proc ?d, p, at, bir, n);
int count to, proto, priority- pro);
int 200 (=0;
while (count (n) {
 for (in) = == ; i<n; i++) }
    if latei) <= ( lf p(i) 7= priority de
                        m(i)1=1)8
   priority - p Cili
 if (+(x)===1)
 xt(x) = c - at(x);
  ? (at(x) L= L)
   ([x]+d = +)
  else
     (+=a+(x)- (+b+(x))
```

```
count + 4 ;
  CATXJ = C3
  m [x]=1;
  while (x >=1 - && m(--x]=1)}
     priority = p(x)
    birek; 5
   X++;
   if (count == in)
     break ;
for (int i==; icn; i++) 1
  tat(i) = (+(i) - at(i))
for (int 1 = 0; icn; i++)
   wtCi] = tatCi] - bl·Ci];
Printf ("PIDI+ prior)+ATI+BT/+CT/+TAT/+WT/PT/
for (int 100; (<n; 1++)
Print F(" P88 1+ 281+1+881+1881+ 881+ 881 + 881 + 881
            14 %dla", proc_id(i), p(i), at(i)
           bt(i), (t(i), tat(i), wt(i), it(i));
for (int i=0) icn i++) {
    + dat + = tat[i]
    twot + = wt(i);
  avg-tat = + tat/ (double) n',
avg- wt = twt /(double) n',
  prints ("In Average turnaround time: " & I fins In"
                            ovg_tat);
print ("In Average Laiting time: % 1 fmslo"
                      ovg wt);
```

PAGE NO 19

PAGE NO: 20

march love 13 leads

output:

Enter number of Processes: 41 Enter Priorities: 10 120 30 40 Enter : arrival times: 0 1 2 4 Forter burst times: 5 4 2 01 11 Non-preemptive Scheduling: PID POTAT BT CT TAT WT RT 1 10 0 5 5 5 0 0 2 20 1 4 12 811 47 7 3 30 2 2 8 66 74 4 4 40 4 1 6 82 1 Average trinwround time: 6.000000 ms

Average Daiting time: 3.000000 ms

16 5 19 45

of the little Calles Called 1 to

to Clastering toil machiner toy

20 - first states

The state of the second state of the

and the second of the second of the

DATE: 02 (-6) 214

```
3 Multilevel Queue
  # Poclude (Statio h)
  void willist processed, into; int bill
          iot atci, int weels
  WIEDTO :
  for (int i=1; ien; i++)1
   WHE i] = bt[i-1] + wt[i-1] - at[i-1];
  if (w) (i) (i) )

WH(i) = 0; y
 void TAT (int processes [], int n, int b+[]
       int wich, int tated) f
 for (?n) :=0; ?(n; i++) !
  tat(:) = bt(i) + wt(i); }
 void roundRobin (int processes [), int n, int beci,
                     int at() int quantum) ?
 int wten, taten, cten, total wt=0,
                total - tat = 0;
 int remaining by [n];
 int completed = 03
 int time so;
for (int 1:0) 1:01 1 +4) 1
 remaining bt (i) = bt[i];
while Completed < n) {
for (int 7:03 ikn : 1++) {
?f Cremaining - bt-[i] >0 ++ at[i] <= +ime) {
if (remaining - bt[i] <= quantum) }
  time + = remaining b+ (i) i
  remaining - bt [i] = 0;
   (+[i] = time;
```

```
PAGE NO 2 1
```

```
completed ++;
                3 else /
                    Hime + = quantum;
                 remaining bt [i] = quantum; y
          3
      I witness to the western of Apollows
      WT (process o bt at wt);
    TAT (processes, n, bt wt, tat);
     printle ("Processes Burst Time Arrival Time
      Waiting Time Turn Around Time Completion Time In ");
     for (int 1=0; icn; i++) }
     priot ( P % dlt 1+ %d(+ 1+ %d(+ 1+ %d(+ 1+ %d(+ 1+ %d)+ 1+ %d)+ 1+ %d(+ 1+ %d)
                       In procuses (i) bt(i) at(i) wt(i) tat(i) (10)
    total - w+ += w+[i];
    total fat + = tat (i); }
   printf ("Avg WT = %Flo" (Flock) total wit 10);
   printf ("Avg TAT = % Fln" (flowt) total - tat /n);
  void fife (int processor) that a int btc) intato) }
   int witing, tation), (+(n), total with total tatio;
   WT (processes in bt, at, wt);
 TAT (proceny of bt, wt, tat);
 printf ( Processes Bust Time Arrival Time Waiting Time
                    TwenAround Time Completion Timela);
 for (int 1:0; i(n; i++)}
   C+Ci) = a+Ci) + b+Ci);
  Printf("p%d1+1+%d1+1+%d1+1+%d1+1+%d1+1+%d1+1+%16"
             processes (i), b+(i), a+(i), w+(i), ta+(i), (+(i))
  total wh + = wt (i);
  total tat t= tat (i); 3
printf ('Avy WT (fifi) = %Fln', (floot) total-wt/n);
printf ("Avg TAT(fifs): % (float) total tat/n);
```

	Pot main () 1	(ii)
	1 1 2 3 9 3 1	
	int of size of (processes) / size or (processes Los);	
	int bt[]= 10,5,8,12,183:	
	int at ()= {0, 1, 2, 3, 43	
	int munching = 9)	
	round Robin (processes, n, bt, at, quantum);	
	tits (processes in bt, at).	
	retuin 0;	
	3 James Later Land 1922 1 2 February	
1	botout:	
207	De la	CT
	Processes Bust Time Arrivaltime Waitinglime TAT	39
- 110	10 15	23
	Pa 5 9 2 14 22	33
	P4 12 3. 20 3.2	45
	P5 15 4 29 44	50
A1	ug WT (Round Robin) = 14.6000	
+ + +	vg TAT (Round Robin) = 24 - 6.000	
7 - P	MINES BE ALL WIT THE CIT	
F	10 10 1	
P	1 10 15 6	
P.	3 20 32 15 12 3 20 44 19	
0	3 20 32 15	
1000		
P.		
A	9 WT (F(FS) = 14.60000.	
AV	9 TAT (FCFS) = 24.60000	

PHOR NO 214 CARE OS [04] 24

```
Rate monatonis
    # include < stdio h>
    # include < Stalib h)
     # include < math h>
     void soit (int pocl), int b(), int pt(), int n) [
     int temp to:
    for (int 10) ich; 14+) }
     for lint j :: j : j : j + ) !

rf (ptCj] < pt (i)) {

temp = pt (i);

pt (i) = pt (j);
       pt (j): dempi
        b(j) b(i);
         b(1): temp) 11 + (7) 11
        semp processes i gont
        proc (i) : proc(j): and if
      proc (j) + templ; 3
      300 11 500 972 met 15777 134 300
     3 " ( met 1 0 000 1 000 1 000 1 m)
    int god (idt a fint b) { " I have
    int rate and a comma
    While (670) }
      Y= a 66; (41) 'n' 11 1414
      a s b; " (1) to 1 (7) of ( alternal ) " a med
 11 - 11 b= 11 3 in 3 months of 12 13 death
   return 6 ? - Plant - 31 8 of 1 Plant
2 mily " with ? ( core - 2 mil) see
    int (comul (int p(), inta).
   int lim = p(o);
    for link i=1; icn; i++) {
```

```
lem: lem* p[i]/ged (1em; p[i]);3
   void main D1 100 min bov
   printf ("Enter no of processes:");
  s(anf ("%); +n);
  int procent, bend, ptend, remend;
  printf ("Enter CPU Bunk time: In");
  for (int i=0; ien; (++) f
  printf ("Enter time period: In");
  for (int is a icos, i+++)
  Scoof (""ad" fpt[i]);
  for (Int 1=0; icn; i+t)
  proc (i) = i+1;
  Soit (proc , b , pt , n) . . .
  int 1: lemol (pt, b)
  print ("lem = "sale"; 1);
  printf ("In Rate Monotonic Schodoling");
 printf ("PIDI+ Bunkl+ Periodla");
 for (int 1=0; icn; [++).
  printf ("%d|+ 11%d 14 17%d 1+ 16%d 1 1 1 18do")
       proc(i), b(i), pt(i));
 double sum = 0.0;
for (int i=0; i(n; i++){ + =0 }
 Sum + = (double) b(i) / p+(i) }
double ths = n* (pow (2.0, (1:0/ni)) -1.0);
printf (") 10 % If < = % IF => % Sta" , sum,
          Ths, (sum <= Ths)? "true" = "fulic" );
exit (0);
printf ("Scheduling accord for old mulala"; 1);
```

Port time = 0, previo, x=0; while (time < +1) }
int fine; for (int i= 0; i(n; i++) f if (time % pt (i) = :0) rem(i) = b(i); if (rem[i] 70) } if (prev != proc (i)) } printf ("idmy onwords: process %d runninglo", prev = proc(i); rem[i] -- ; for; Co Except of a strait break; with the state of the st ix co; 6 min t showing and There & 2 mil ! showing Love y in the second of the second if (d f) f d man i securos trad 16 (x 121) { 1000 2 december 2000 print (" " on weads : (pu is idle la", time); x=1; y and beside wall time ++;

Output:	
Enter the number of Processes: 3	
Enter the CPU bust times !!	
3 2 2	
Enter the time period!	
20 5 10 : 1 - 17	
LCM = 20	
Rate Monotone Scheduling:	
PID Bint Period	
2.1.2	
3 2 10 10	
1 3 20	
0.750000 < = 0.779763 => tove	
Scheduling occurs for 20 ms	
Oms Onwards: Process & grunning	
2ms onwards! Process 3 grunning	
4 ms onwards! Process 1 grunning	
5 ms onwards: Process 2 grunning	
7 ms onwards : Process 1 nunning	
8 ms onwords : CPV is idle	
oms onwards! Process 2 gwoning	

PAGE NO: 3 8

```
2) Evolvest Deadline fitt
# include < stdio. h)
# include «Stalib.h»
# include < math. h>
void soil- (int proct), int d(), int b()
       3 ( n +n; , (3 +q +n;
int temp = 0;
for Lint i=0; i(n; i++) 1.
for (int j : i jen; j++) &
it (90) < 90(1)) }
 demp = dej]:
  d (i) = demp;
  temp: pt[i];
  pt (i) = pt (j);
  pt (j) temp:
   temp = bCj)
   Hemp: proc (1):
   proc(j): proc(j);

proc(j): temp; 3
     The to die was the
  3 70 19 man 31 -10
Tradedic of the suited of 3 34-10
int ged (int a, int b) &
 int vy their wast to be delivered
 while (b) o) & server server
 a 26; 3 2000 add 155" } 7 hive
 b=rzoy man mis for
```

```
PAGE NO 25 2
  int demul (int poli) int o) f.
   ico) q = mil +ni
   for (int i=1; ix n; 17++).
   Lem: (lem * p(i))/ged (kim; p[i]);}
     return lem ; 3 - 11 ...
 void main Of:
  printf ("Enter no of processes:"):::
 Scarf ("060" fn);
 int procent, bend, ptend dend, rementi
 printf ("Enter (PU Bunt: Himm In'l):
 rem (i): b(i);
 printf ("Ealer the deadlines: In"); -
 for (int 1:0; ikn; i++)
 scanf ("%d", fd(f));
printf ("Enter time period!\n"); +
for (int 1:0; i(n; i+1):::

scanf ["?d", & pt [:]):...
for (int i== ; irni, i++)
proc (i) = i+1; 'and :: "
sort (proc d b, pt, n);
Polt L: lemul (pt, n);
printf ("In Earliest Peadine Scheduling: In");
Prints (" PID It Book It Beadline | + Periodin");
for (int 1=0; 7(n; i++) 1 211
bring E ( " org 14/4 org 14/4 org/4/4 org/4/4 org/4/4 org/4/4 org/4/4 org/4/4 org/4/4 org/4/4 org/4/4 org/4/4
process, beis, dess, pteiss;
printf ("Scheduling occurs for god milala", 1)
int time=0, piev=0, X=0;
int next Deadlines (n);
```

```
for (int 100; icn; i++) 1.
  next Deadlines (i) = d(i);
  rem [i]: b[i]; 3
while (+ime < 1) }
 for lint ito; icn; i++) f
   if (time " > p+(i) == 0 of fine 1= 0) {
   next Deadlines (i): time + d(i);
   rem [i] : b(i) i y
int minDeadline = 1+1;
int taskTofkecute = - 1;
for lint ita; ich; it+) {
if (rem (i) > a lef next Deadlines (i) < min Deadlines
Emin Deadline = next Deadlines (i)
 task To Execute = 11:3
if (task To Execute 1 = -1) }
 printf (""dms: Task "6d is running. In" time
             procetask ToExecute]);
  rem[+ask To Execute] --; 3
else ?
  4 printf (" %dms : (PU is idle. In" Hime);
time + + i
output :
Enter the number of processes: 3
Enter the CPU burst time:
0 1 2
Enter the deadlines!
Enter the timeperiods'
```

3 4 6

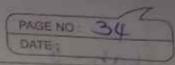
2 0 1 200214

	Earliest Deadline Scheduling:
	PID Busttime Deadline Period. 3 2 4 6
	3 2 4 6
	2 1 5/1 5 10/5 4 15/10/2
	1 1 1 0 3 3 3 3
	Sheduling occurs for 12 ms
1	Oms! Task 3 ?s grunning
	ims: Task & is monning
1	1 ms: Task 3 is nonning
1	3 ms: CPU 38 Polle
l	4 my: Task 2 ?8 munning 1
	5 ms: CPU ?s idle
H	6 ms Task 3 is grunning
I	7 ms: Task 3 is monning
	8 ms: CPU is idle
	9 ms: CPU is idle
	ioms: cou is title !
	11,ms cpu, is tidle
	The state of the s
	The state of the s
	1 12/3
	alietti il villa col col dinina.

Will Proportional Scheduling. # include (Stdio h) # include <stalib h> # include < time. h> # define MAX-TASKS 10 # define MAX_TICKETS 100 # define TIME UNIT DURATION MS 100 struct Task 8 int Hidis Land & Charles of int tickets i void schedule (struct. Task tasks (), int num tasks int + time_span_ms) } int total tickets : 0: for (int 120; 14n; i++) &. total - tickets + = tasks[i]. tickets; Sant 2 134 (\$10 lans Stand (time (NULL)); int correct time to ! int completed tasks:0; printf ["Proce to Scheduling ! In"); while (completed - tasks . < num - tasks) { int winning - ticket = rand () % total - tickets: int comulative tickets = 0: for (int i=0; iknum_tasks: it+) & i Cumulative tickets + = tasks[i]. tickets; if (winning ticket < womulative ticket) { printf ("Time %d- %d: Task %d is running In", worrent time, current time +1, tresks[i], tid); coverent time + +; break; y y completed tasks ++;

* time_span_ms = convent_time * TIME_UNIT DURATTON MI: int main () f Struct Task tasks [MAX-TASKS]; tot num_tasks! int time span msi printf ("Enter no of taske:"); scanf ("ord" of num tasks); if (num-tasks <= 0 11 num-tasks) { printf ("Invalid no of tasks. Plean enter valid number : \0", MAX_TASKS); return 1: - 14,100 July 1 1/2 4 printf ("Enter no of tickets: \n"); for (int ?= 0; ic num tasks ; i++) tasks[i], tid = i+1; printf ("Task %d tickets:", taskir(i). tid); scanflood", Stockets); to talk battle don't to the printf ("InRunning tasks: In"); schedule (tasks num. tasks, of time_span_m); print & C' la Time span of Gard + Chart : "d milliscrands In", time_span_ms); return of the same (total mikelines = delil schooles) to Value V 18 68 NOT : 68 - 68 Lini 1 .) +4 102

to the think the will be the track with fasting



output: Enter number of tasks: 3. Enter number of tickets: Talk 1 tiket: 10 Task - 3 Hicket: 30 -11. Running tasks: 1 1 days 30 Process Scheduling! Time 0-1: Task 3: is running 2 is running Time 1+2: Task Time 2-3: Task 2: is running Time span of Gantt chart 300 milliserands 7 45 13 12 10 (10 100) Colony of (10) 7160 the state was stated to the state of the 11. 1 ASTRUE (10 -1 whoma) bd. (100 xshoms) 48 : (" P. 11.0 E ...) 22.0 "] 22.05.1 tion or man as the continual

```
Lab - 5
1 Write a C program to Simulate producer.
  Consumer problem using semaphores.
 # include < stdio. h>
  # include <stalib.h>
  int mutex = 1. full = 0, empty = 5, X=0;
  int main () f
  iot o:
  void producer();
  void consumer ():
  int wait (int);
 not signal (int):
 printf ("In 1. Producerlas. Consumerlas. Exit");
 while (1) 8
  print f ("Infinter your chaice:");
  scanf (" "d" fn);
  switch (n) {
  case 1:
     if ((mutex ==1) ++ (empty 1=0))
     producer ();
     9250
     printf ("Buffer is full!!");
   break;
 cose 2:
    if ((mutex ==1) &f (full != 0))
    consumer ();
    else
    printf ("Buffer is empty ! 1");
    break ;
Case 3:
  exit(a);
```

3 return (a); 4

int wait (int s) & return (-- s); } ent signal (int s) f return (++s); void producer() f mutex = wait (mutex); full = Signal (full); empty = wait (empty); printf ("In Producer produces the item %d" x); mutex = Signal (mutex); yord consumer () f mutex = wast (unutex); full = wast (full); empty = signal (empty); printf ["In Consumer consumes . Hem %d", x); mutex = signal (mutex) : , but blockers at the title and touch output: 1. Produces a. Consumer 3. Exit Enter your choice: 2 Buffer is empty!! Enter your Choice:1 produces the item 1 Enter your Choice: 1 peroducer produces the item 2

PAGE NO. 37

Enter your choice: 1 - + il -Produces the itemis Enter your choice! I
Produces produces the idem 4 Enter your choice: 1 Produces produces the item 5 is the Enter your choice ! I was the state of Buffer is full !! Enter your choices a Consumer consumer item 5 Enter your choice 182 11 11 Consumer consumer item 4 Enter your choice : 2 Consumer consumes Ptem 3 Enter your choice 12 w in the consumer consumes litem a Enter your Charce 2 Consumer consumes item 1 Enter your choice ! 2 Buffer is empty! Enter your choice: 3 Program execution completed.

mula d

remine) 6

4179 3

PAGE ND 38 Lab -7 @ write a c program to simulate Bankers algorithm for the purpose of deadlock avoidance # include < stdio h> int main() { print f ("Enter the number of processes:"); Stant ("%d", &n); printf ("Enter the number of resources: "); Scanf ("" &m); int ovailable (m); Printf ("Enter the available resourcesie"); for (int =0; icn: i++) { for (int j=0; j<m; j++) { Scanf ("%d", &maximum[i][j]); int allocation (n) [m]; printf ("Enter allocated resources for each proces: In"); for (int ito; ito; i++) } for (int jess jem; j++) f Scanf ("%d", fallocation (i) (j)); y int need [n] [m]; for (int i=0; icn; i++) / for (int j=0; jem; j++) in need (i) [j] = maximum (i)[j] - allocation (i)[j); 117 Filmstonon + + Filmed for (int i=0; ikn: i++) { printf("1 p%d: 1" 1+1); for (int j=0; jem; j++) { printf (" "at", allocation [i][j]);

```
for (int j=0; jm; j++)t
                          3 printf ("id" maximum [i][j]);
                        for (int j=0; jcm; j++)?

printf("%d", need [i][j]);

printf ("I"); }

int work (m);
                         for (int 1=0; 12m; 1++){
                   work (i) = available (i);
                        int finish (n);
                       for (int 1:0; (2n: 1++) }
                        finish [ i] = 0; y
                        int sofesequence [n];
                        int Count =0;
                        int sofe = 1;
                        while (count en) ( ) is the other to
                      ant found = 0; is totally wind I it is
                       for (int i=0; i<n; i++) }
                         ? F (finish (i) = = o) / ...
                          int if it is the sales of the sales
                            for (j=0; j(m; j++)}
if (ned (i)(j) > nork (j)) {

break; y}

if (j = m) {

for (j = 0; j < m; j + +) {

Example 1.5 | f < m; j + +) {

Example 2.5 | f < m; j + +) {

Example 3.5 | f < m; j + +) {

Example 3.5 | f < m; j + +) {

Example 3.5 | f < m; j + +) {

Example 3.5 | f < m; j + +) {

Example 3.5 | f < m; j + +) {

Example 3.5 | f < m; j + +) {

Example 3.5 | f < m; j + +) {

Example 3.5 | f < m; j + +) {

Example 3.5 | f < m; j + +) {

Example 3.5 | f < m; j + +) {

Example 3.5 | f < m; j + +) {

Example 3.5 | f < m; j + +) {

Example 3.5 | f < m; j + +) {

Example 3.5 | f < m; j + +) {

Example 3.5 | f < m; j + +) {

Example 3.5 | f < m; j + +) {

Example 3.5 | f < m; j + +) {

Example 3.5 | f < m; j + +) {

Example 3.5 | f < m; j + +) {

Example 3.5 | f < m; j + +) {

Example 3.5 | f < m; j + +) {

Example 3.5 | f < m; j + +) {

Example 3.5 | f < m; j + +) {

Example 3.5 | f < m; j + +) {

Example 3.5 | f < m; j + +) {

Example 3.5 | f < m; j + +) {

Example 3.5 | f < m; j + +) {

Example 3.5 | f < m; j + +) {

Example 3.5 | f < m; j + +) {

Example 3.5 | f < m; j + +) {

Example 3.5 | f < m; j + +) {

Example 3.5 | f < m; j + +) {

Example 3.5 | f < m; j + +) {

Example 3.5 | f < m; j + +) {

Example 3.5 | f < m; j + +) {

Example 3.5 | f < m; j + +) {

Example 3.5 | f < m; j + +} {

Example 3.5 | f < m; j + +} {

Example 3.5 | f < m; j + +} {

Example 3.5 | f < m; j + +} {

Example 3.5 | f < m; j + +} {

Example 3.5 | f < m; j + +} {

Example 3.5 | f < m; j + +} {

Example 3.5 | f < m; j + +} {

Example 3.5 | f < m; j + +} {

Example 3.5 | f < m; j + +} {

Example 3.5 | f < m; j + +} {

Example 3.5 | f < m; j + +} {

Example 3.5 | f < m; j + +} {

Example 3.5 | f < m; j + +} {

Example 3.5 | f < m; j + +} {

Example 3.5 | f < m; j + +} {

Example 3.5 | f < m; j + +} {

Example 3.5 | f < m; j + +} {

Example 3.5 | f < m; j + +} {

Example 3.5 | f < m; j + +} {

Example 3.5 | f < m; j + +} {

Example 3.5 | f < m; j + +} {

Example 3.5 | f < m; j + +} {

Example 3.5 | f < m; j + +} {

Example 3.5 | f < m; j + +} {

Example 3
                          work[j] + = allocation[i][j];
                          Safesequence [count ++]= ?: 19 19 19 19 19
                        found = 13 4 / - 1
                                     :(COURT miscolles "0 or 7440).
```

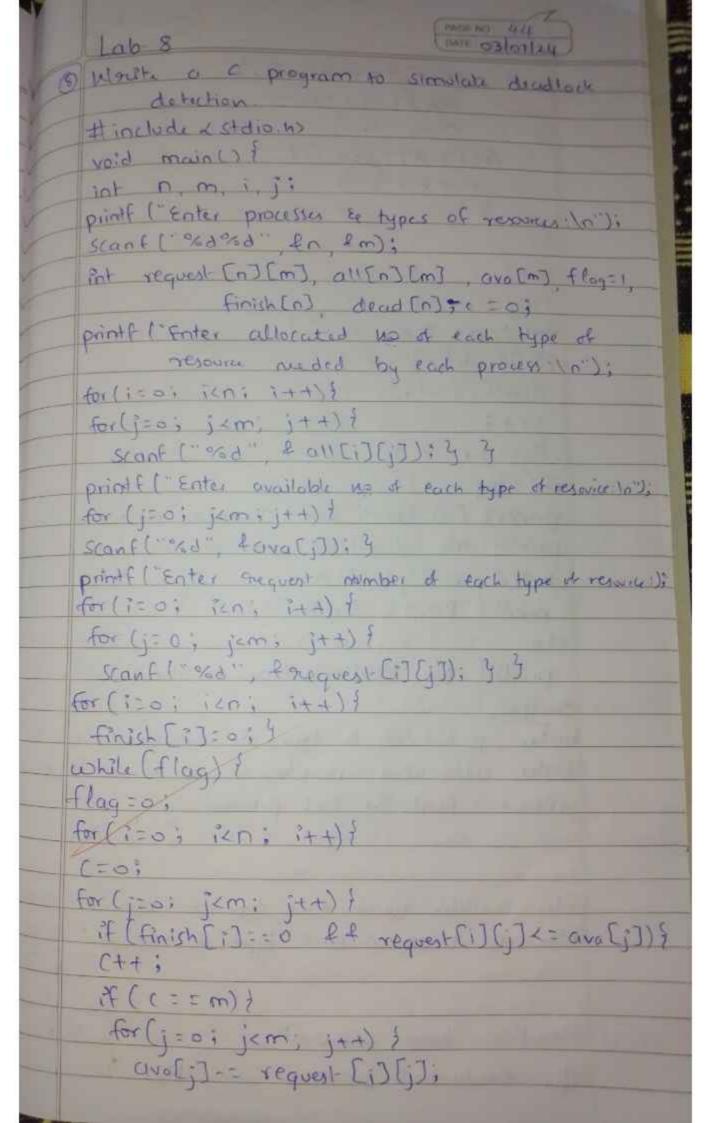
t brosto

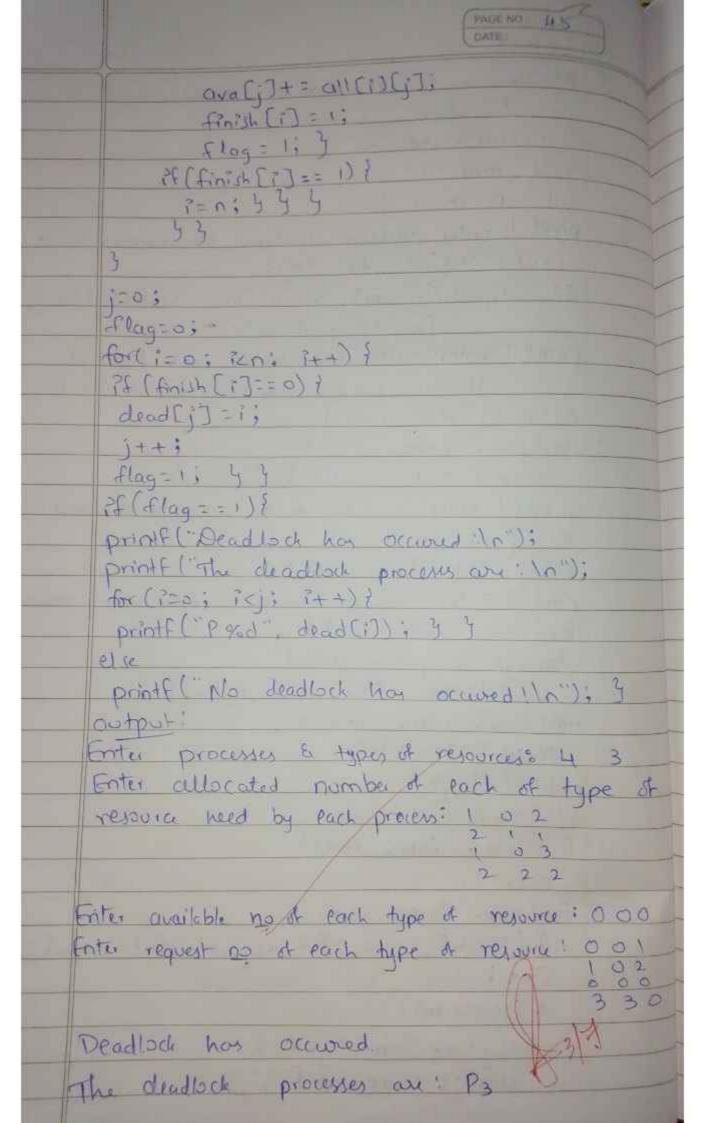
```
190 HO 40
  if ( ! found) {
   Sofe = 0;
  break & y y
  if (safe) {
  printf ("The system is in sofe state. In");
  printf ("Safety sequence: ");
  for lint 1:0; 32n; i++)f
   printf ("Pord", safe sequence to [i]+1); }
  printf ("In"); }
  elsef
  printf ("The system is in unkafe state &
       might head to diadlock. In");
  3 return of 3
  Output!
-> Enter the number of processes: 5
 Enter the number of resources: 3
 Enter the ovailable resources: 3 32
 Fitter maximum resourcy for each processes: 753;
    2 2
    0 2
    2 2
                19/4
 4 3 3
 Enter the allocated resources for each processes: 0 10
 3 0 2
 2 1 1
 0 0 2
                    Max Need
 Processes Allocation
                        5 3 74 3
 Pi 0
                     3 2 2 0 2 0
     3 0 2
 PL
                     902600
          3 0 2
 Ps
                    2 2 2 0 11
 Pu
          0 0 2 4 3 3 4 3 1
 Pt
 Safety sequence! P2 P3 P4 P5 P,
```

@ Write a C program to simulate the concept of Dining-Philosopher's problem. # Paclude < statio in> # define MAX_PHILOSOPHERS 5 MM 21,119 void allow one to eat (int hungry (), int n) ! int isWaiting [MAX-PHILOPHERS]; for (int 11=0; 140; (4+) /1 -9") 740.0 Esthatting Cil = 13 9 1 1 (17 2) 24 19 for (int 1=0; 120; 3++) } prints ("Pred is granted to eat in", hungry [1]): is Wating Changry [i] = 0 it is for lint j=0; jen; j++) { who ! 36 (35 Mailing Chungny (7)) } " Market print (P % d is investing and bungay (i)); 5 I for Crat k = 0; k < n; k++) f it is is Marting (R) = Grand at all I is Waiting (Kungry (17) = 0; Files void allow-two-to-eat- (int bungry I, int a) { IF (n < 2 11 OF MAX_PHILOSOPHERS) F printf ("Invalid no of philosophers. In); of greatures of a formation and when For (int 1:0; 1cn-1: 1++) } for (int j=i+1; j<n; j++) ! printf ("posd and posd are garried togetha" hunger (i), hungry (j)); For (Pot k = 01 k < 0; k = 1) 4 1 A 2011 1000 print [P %d is waiting (hungry (k)); } 3 4 8 8 10 4 9 9 Catifu Sequence 1 63 Pg PG PG PG Pg

DATE Pot main () } int total philosophers; hungry count; int bungry - positions [MAX PHILOSOPHERS]; printf (Doter total no d philosophen); Scanff. %d" & total philosophens; 3F (total - Philosophers > MAX PHTLOSOPHER 11 total philosophers < 2) { prints ("Invald number of philosophen in"); netwo 1; 3 print [(" How many are hungy!"); scanf ("sd", & hungay - count); of Chunging count < 1 Illinging count > total philosophed printf ("Invalid number of hungry philasophen. In"); 94 hin 1: 4 for (int i=0; is hungry count; i++) } printf (Enter philosopher and position: ", i+1); scanf (- %) & hungy positions (i)); ? [(hungry - positions (i) < 011 hungry - positions (i) >= 1) 1 (chad - philosophers) { printf [Invalid philosopher position In); return 1; 4 int ch; while (1) & printf ("In 1. One can out at a time la); printff" In 2. Two Can lat at a timelo " printfl" In 3. Exit In Di printfl Enter your choice: "); Scanf ("" hd", & cha); Switch (ch) { Case 1: cellow_one_to_eat (hungry-positions, hungry-rount); break;

Case 2: allow two to cat (hungry positions, hungry-bunt); break ; Cas- 3' exit(a); default: 2 printf! Invalid Choice to "); return o: output: Enter total no of philosophens: 5 How many are hungry: & Enter philosopher 1 position: 1 Enter philosophin 2 position 4 1. One can eat at a time 2 Two can eat at a time 3 Fx 11 Enter your Choice !! PI is granted to eat-P4 is granted to eat 1 One can eat at a time 2 Two Can lat ata time 3 Fxit Enter your choice 2 PI and Py oble granted to Pat 1. One can lat at a time 2. Two can lat at a time Enter your Chair: 3





Lab-9 a program to simulate the following continues memory cultocation techniques a) Work - 6+ 10 Best - fil-C) FIRL- AL # Poclude Estations Struct Block 1 int black no: int black size; int is free; struct file? int file-noi int file-size. wood first Fit I struct Block blocks), int no blocks, struct File files [] : int n. files) ! printf ("File no: 1 + File size: 1+ Block size: 1+ Fragmint (n"); for (int ?= 0; ikn files; i++) } for (int ; = 0; j<n-blocks: j++) + if Chlocks [j]. is-free &4 blocks [j]. blocks size >= file size) { blocks[j]. is free = 0i printf("%d1+1+8d1+1+8d1+1+8d1+1+8d1+1+8d10", files[i]. file no files(i) file size blocks(j). block-no, blocks(j) block Size - film[i) file-Size); break; 3 void worst Fit (struct Block blocks[], int n. blocks Struct File flust], int no files) !

DATE OF ATT

Printf (" File no ilt File size It Block - no 14 Block size: (+ Fragmentin'); for link to; ren- files; it+) ? int worst fit black =- 1; Pol max fragment =-1; for link j=0; j<n-blocks; j++)} of (blacks [j]. 3. free & blacks [j]. blacks size > Files [1]. File_size) { i'nt fragment = block (j) - block - size - fiter (i). file - Size; if (fragment > max-fragment) } max fragment = fragment; work fit - block = 13 9 24 (wast-fit-block 1=-1) } blocks (worst-fit-block). is free = 0; printf ("8d 1+1+8d 1+1+8d1+1+8d1n", files (i) file no files [i] file-5138 blocks [wort fit block] block-no blocks[worst-fit-block]. block-size max-fragment); I void best-fit (struct Block blocks (), int n-blocks, struct File files[] int n_files] ! printf ("File no: It File size: It Block no: It Block size 14 Frag ment (n'); for (int ?= 0; ikn files i++) } int best- At block 7 1; int min fragment = 10000; for (int j=0; jen-blocks; j++) } if (blocks[]] is free && block[] block-size 7= filestil. files (13) int fragment = block [] block - size 7 = filey(1). File-5135)}

```
DATE NO 4 8
   of int (fragment < min_fragment) }
     min fragment = fragment :
    best fit black = 1; 3
 TF ( best - 6+ - black 1 = -1) }
 blocks [ best- fit - block], is free =0;
 print( " " d)+ 1+ " cd)+ 1+ 8 d)+ 1+ = d( ) . Ale no.
 files [i] file-size, blocks[best-fit-block] block no,
 blocks [ best - fit - block]. block _ Size , min fragment); }
 int main()
 int n-blocks n-film;
 printf ("Enter no of blocky");
Scanf ("45d", &n-blocks);
printf ("Enter up of file");
Scanf [ "Sd" &n files):
struct Block blocks [n blocks]:
 for (int foo; isn_blacks; i++)!
 block [i]. block no = i+1i
printf ("Enter size of book "d:", iti);
 scanf ("%d" &blocks (i] block - size);
blacks[i]. is free - 1; }
stouck file files [n files];
for lint iso; icn-files; i++) {
files[1] file no= it1;
printf ("Enter size of file %d:", i+1);
scanflied 4files (i). file_size); }
firstfit (blacks, n-blacks, files, n-files);
printf ("In");
for (int i=0; icn-blocks; i++) {
blocks [i]. ?s free = 1; 3
```

part infallace Rogram - 10 White a C program to simulate tage ruplacement algorithms COFIED b) LRU C) Optimal # include < stdio m # include klimits. h> # shelvde < stalib.4> void fifelint pages[], int n, int capacity)! int frame [capacity], sidex = 0, page_faults = 0; for (Bot ?= 0: is Capacity: ?++) ? frame (+1]:-1; for (the 700 isp: 1++)1 int found = 0; for (int j=0; j (apacity; j++)! if (frame (j) = = pages (i)) ! found = 1: break ; 3 of (!found) { frame [index] = pages[i]; index = (index +1) 2 capacity: page faults + + ; } printf ("FIFO page faults: "Edla", page-faults); void trulint pages [], int n, Bot capacity)! frame (capacity), counter (coepacity), time = 0. page - faults = 0; for (int 1:0: iccapacity; it+)}

-1 = fame[i]=-1;

PAGE NO 45 wont fit (blacks, nublacks, files, nu files); printf ("10"); for (in) ?= 0; icn_blocks; i++) { blocks [i] is free = 1; best Fit (blacks, noblacks, files, nofiles); guturn 0; output: Enter us of blocks: 3 Enter no of files: 2 Enter size of black 1:5 Enter Six of block a: 2 Enter SIX of black 3: 7 Enter size of file 1:1 Enter Six of file 2:4 Memory Management Scheme- First fit File_no Block_no Block_size Fragment Memory Management Scheme: Worst fit File Size Block no Block size: Fragment File-no 1 3 7 Memory Management scheme: Best fit File no Filesize Blackeno Blacksize Fragment

counter (i)=0; for (int i=0; i<n; i++)/ Port found = 0; for lint jeo; je capacity i jet) 1 of [frame[j] = pages[i])? Counter[] + Home ++; break; 5 of (IAound) } int min: INT_MAX, min_index =-1; for (int j=0; j (capacity; j++)! if Crounker (j) < min) { min = Counter [j]; min_index=ji3 freme[min_index] = pagel[i]i Counter [min - index] = time + + ; page faults ++; } printf ("LRV Page faults: "EdIn", page faults); void aptimal (int pages (), int n, int capacity) { int frame [capacity], page faults = 0; for (int i=0; i< copacity; i++) frame [i] = -1; for (int 1=0; icn; i++)3 int found = 0; for (int j= 0; jx capacity; j++)) it (frame [j] = = pages (i))} found = 1; break: 4 4

```
PACE NO 52
   of (I found) }
    int faither = it 1;
    int index = -1;
   for (int j: 0; jecapacity; j++) !
     int k;
     for (k= ++1; k < n; k++) !
    it (frame [j] == pages (k))
      break; 4
     3F (k > foothest) }
     forthest = k;
    index = j; } }
  of (index = = -1) }
   for link j= 0; je capacity; j++) }
   3f (frame (1)==-1) f
    index = j :
   break: } }
 frame [index] = pages [i];
page faults + +; }
 printf ("Optimal page faults " godla", page faults);
 int main () }
int n. Capacity;
prints ("Enter the number of Pages");
Stant ("864", fn);
int *pages = (int*) malla (n* size of (int));
printf ("Enter the pages");
for bint 1=0; icn; i++)1
Scanfl" ofd" &pages [i] !
printf ("Enter the time capacity");
scanf ("%d" + capacity);
printf ("InPages:");
```

