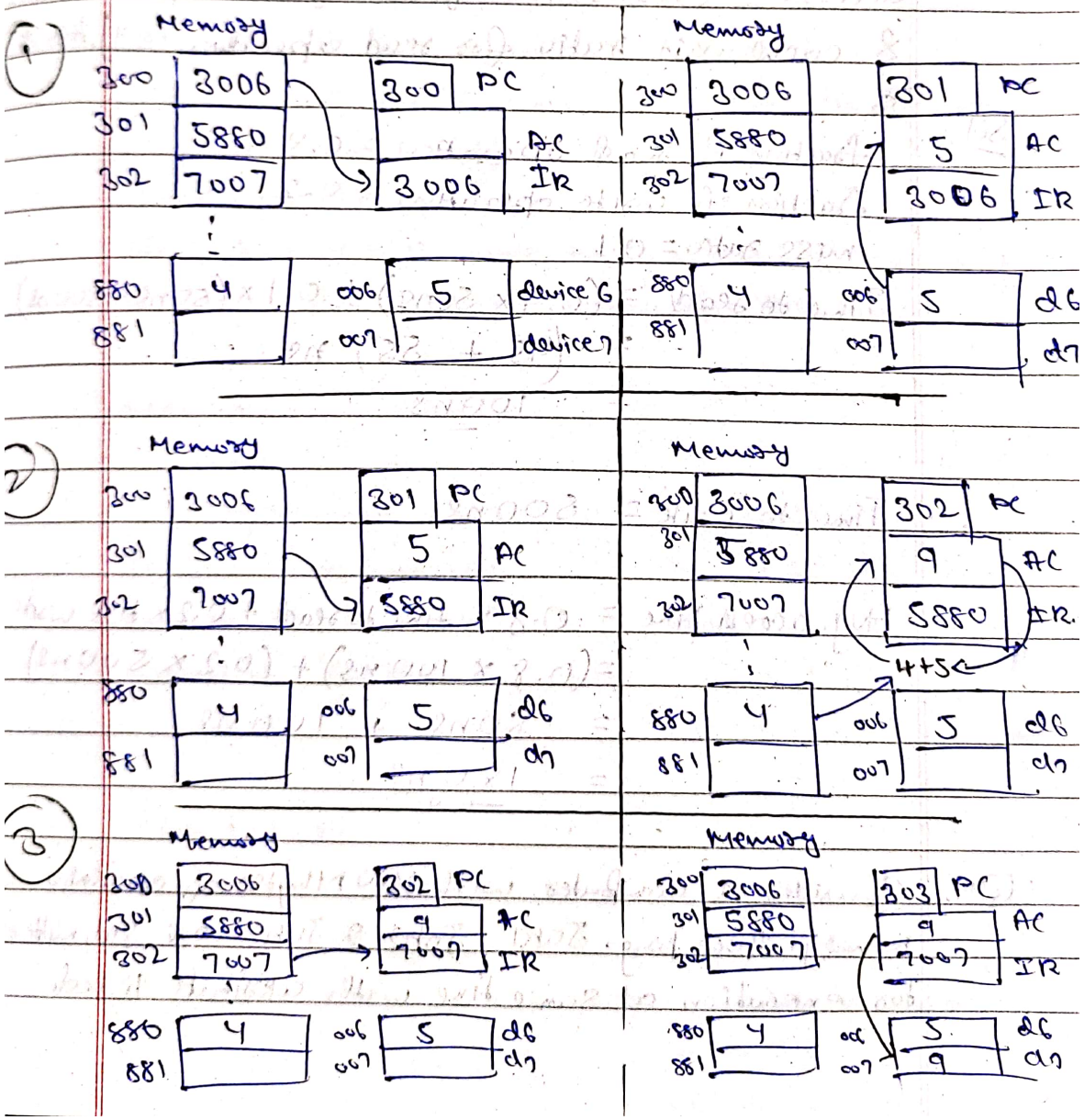
Page No.
Date: / /20

Assignment-1

1. Suppose hypothetical processor has 2 I/O instr:
   0011 = Load AC from I/O
   0111 = Store AC to I/O
   In these case, 12 bit address identifies a external device. Show prog. execution
   (a) Load AC from device 6
   (b) Add content of memory location 880
   (c) Store AC to device 7
   Assume next value retrieved from device 6 is 5 & location 880 contains a value 4.



Page No.
Date: / /20

Q2 Consider a hypothetical 64-bit microprocessor having 64-bit ins. comprised of 2 fields. The first 4 bytes contain opcode & rest 4 contain operand address. What is maxi directly addressable memory capacity?

Sol. 4 bytes = 32 bits
So, 64-32 = 32 bits contain operand address
So,
Maximum Memory capacity = 2^{32} bytes.
= 4GB

Q3. Col. the system's avg. access time. The key data includes cache access time (50ns), main memory access time (500ns), the fraction of read operation (80%) & cache hit ratio for read operation (0.9).

Sol.
fraction of read operation = 0.8
fraction of write operation = 0.2.
miss ratio = 0.1
Time to read = (0.9 x 50ns) + 0.1 x (50ns + 500ns)
= (45 + 55) ns.
= 100ns

Time to write = 500ns.

Avg. access time = 0.8 x time to read + 0.2 x time write
= (0.8 x 100ns) + (0.2 x 500ns)
= 80ns + 100ns
= 180ns

Q4. Consider a computer with two Mbytes of available memory. Three prog. Job1, Job2 & Job3 are submitted for execution at same time with attribute listed.

Page No.
Date: / /20

	Job1	Job2	Job3
Type of Job	Heavy Compute (90%)	Heavy I/O (10%)	Heavy I/O (10%)
Duration	10 min	20 min	15 min.
Memory req.	100M	150M	125M

Executed in sequence Job1, Job2 & Job3. Find out CPU utilization, memory utilization & throughput in uniprogramming & multiprogramming

Sol: CPU utilization –

- Uniprogramming
Job1 → 90% for 10 mins = 9 mins of CPU time
Job2 → 10% for 20 mins = 2 mins of CPU time
Job3 → 10% for 15 mins = 1.5 mins of CPU time
CPU Utilization = 12.5/45 => 27.78%
- Multiprogramming
CPU Utilization is same as max time of uniprogramming i.e., 90% of 10 min.

Memory Utilization

- Uniprogramming
The maximum memory used by job i.e., Job2. which is 150M
- Multiprogramming
The total by all Jobs = 100 + 150 + 125 = 375M
So, 375M is utilized.

Throughput

- Uniprogram.
1 job in 20 min.
- Multiprogram.
3 jobs in 20 min.

Scanned with OKEN Scanner

Date: / /20

Q8. In a batch OS, 3 jobs are submitted for execution. Each job involves an I/O activity, a CPU time & another I/O activity of same time spans as 1st job. Job1 takes total 23 ms to complete, with 3ms for CPU. Job2 takes total 29 ms to complete with 5 ms for CPU. Job3 takes total 14 ms to complete with 4 ms for CPU.

Sol^n.

Uniprogramming.

Total CPU time = 3 + 5 + 4 = 12 ms.
Total time = 23 + 29 + 14 = 66 ms.

CPU utilisation = $\frac{12}{66} \times 100 = 18.18\%$.

Multiprogramming

Total CPU time = ~~longest~~ 3 + 5 + 4 = 12 ms
Total time = longest time of Job i.e., Job2
= 29 ms.
CPU utilization = $\frac{12}{29} \times 100 = 41.38\%$.

Q9. Trace following prog. segment & determine how many process created

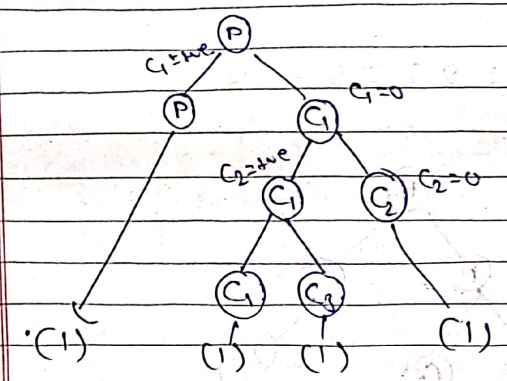
a.
```
int main(){
pid_t c1,c2;
c2=0;
c1=fork();
if(c1==0)
  c2=fork();
if(c2>0)
  fork();
printf("1");
return 0;
}
```

b.
```
int main(){
pid_t c1=1, c2=1;
c1=fork();
if(c1!=0)
  c2=fork();
if(c2==0)
  fork();
printf("1");
return 0;
}
```

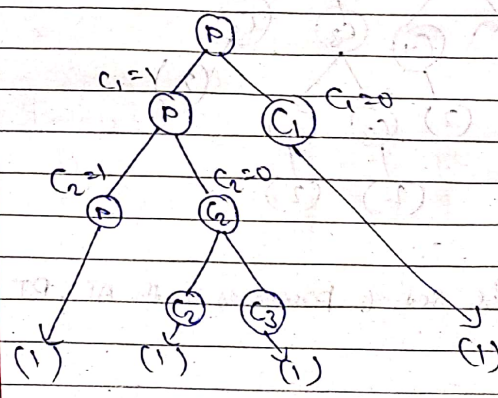
c.
```
int main(){
if(fork() || fork())
  fork();
printf("1");
return 0;
}
```

d.
```
int main(){
if(fork()&& (!fork()))
  if(fork() || fork())
    fork();
printf("2");
return 0;
}
```

(a) Output : 1 1 1 1



(b) Output : 1 1 1 1



Scanned with OKEN Scanner

Q5. In a batch OS, 3 jobs are submitted for execution. Each job involves an I/O activity, a CPU time & another I/O activity of same time span as 1st job. Job1 takes total 23 ms to complete, with 5ms for CPU. Job2 takes total 29 ms to complete with 5ms for CPU. Job3 takes total 14ms to complete with 4ms for CPU.

Sol^n:

Uniprogramming:

Total CPU time = 3 + 5 + 4 = 12 ms.
Total time = 23 + 29 + 14 = 66 ms

CPU utilisation = $\frac{12}{66} \times 100 = 18.18\%$.

Multiprogramming

Total CPU time = ~~longest~~ 3 + 5 + 4 = 12 ms
Total time = longest time of Job ie, Job2
= 29 ms.
CPU utilization = $\frac{12}{29} \times 100 = 41.38\%$.

Q6. Trace following prog. segment & determine how many process created

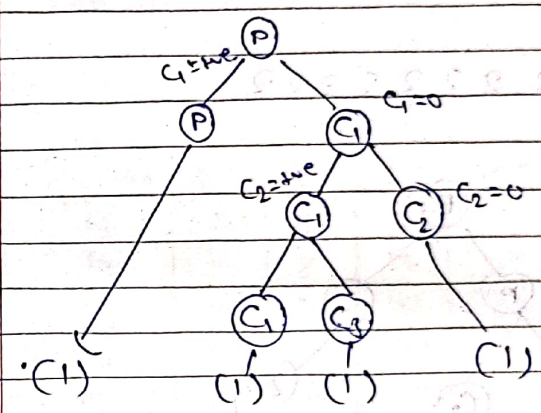
a.
```
int main(){
pid_t c1,c2;
c2=0;
c1=fork();
if(c1==0)
  c2=fork();
if(c2>0)
  fork();
printf("1");
return 0;
}
```

b.
```
int main(){
pid_t c1=1, c2=1;
c1=fork();
if(c1!=0)
  c2=fork();
if(c2==0)
  fork();
printf("1");
return 0;
}
```

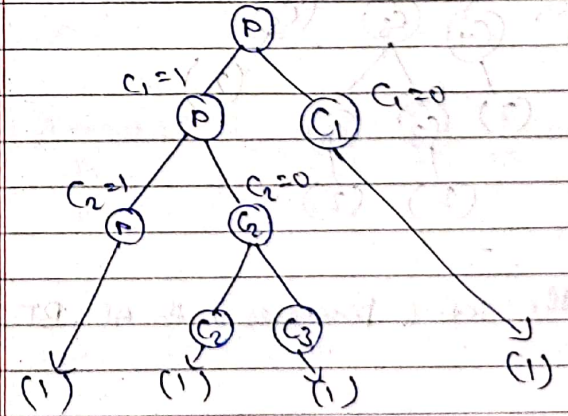
c.
```
int main(){
if(fork() || fork())
  fork();
printf("1");
return 0;
}
```

d.
```
int main(){
if(fork() && (!fork()))
  if(fork() || fork())
    fork();
printf("1");
return 0;
}
```

(a) Output : 1 1 1 1



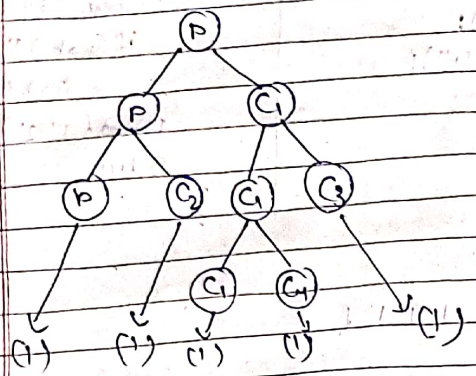
(b) Output : 1 1 1 1



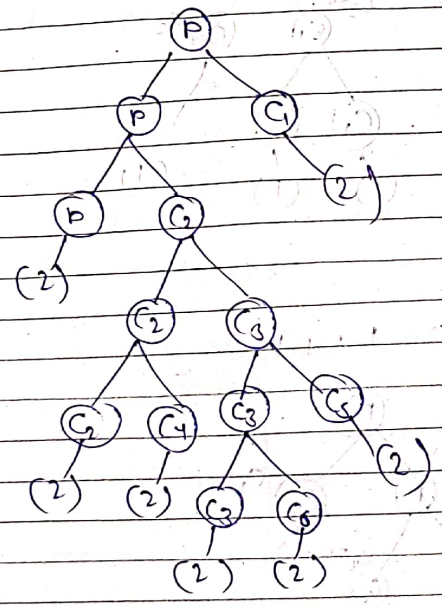
Scanned with OKEN Scanner

Date: / /20

(c) Output: 11111



(d) output: 2222222



Q7. Consider the set of processes with AT, BT & priority

Process	A.T.	B.T	Priority
P_1	0	11	1
P_2	0	8	0
P_3	12	2	3
P_4	2	6	2
P_5	9	16	4

Find avg. TAT, avg. WT & avg R.T.

- FCFS, • SJF, • SRTF, • Priority (non preemptive)
- Priority (preemptive), • Round robin, • highest response time.

FCFS:

Process	A.T	B.T	C.T.	T.A.T	W.T	R.T
P_1	0	11	11	11	0	0
P_2	0	8	19	19	11	11
P_3	12	2	43	31	29	29
P_4	2	6	25	23	17	17
P_5	9	16	41	32	16	16

P_1	P_2	P_4	P_5	P_3
0	11	19	25	41	43

avg TAT = 28.2 ms
avg WT = 14.6 ms
avg RT = 14.6 ms

SJF:

Process	A.T.	B.T.	C.T.	T.AT	W.T.	R.T.
P_1	0	11	27	27	16	16
P_2	0	8	8	8	0	0
P_3	12	2	16	4	2	2
P_4	2	6	14	12	6	6
P_5	9	16	43	34	18	18

P_2	P_4	P_3	P_1	P_5
0	8	14	16	27	43

Scanned with OKEN Scanner

Date: / /20

avg T.AT = 17ms
avg W.T = 8.4ms
avg R.T = 9.4 ms.

- SRTF

Process	A.T	B.T	C.T	T.A.T	W.T	R.T
P_1	0	11	27	27	16	16
P_2	0	8	8	8	0	0
P_3	12	2	16	4	2	2
P_4	2	6	14	12	6	6
P_5	9	16	43	34	18	18

P_2	P_2	P_4	P_4	P_4	P_3	P_1	P_5
0	2	8	9	12	14	16	27	43

$P_1=11$ $P_1=11$ $P_1=11$ $P_1=11$ $P_1=11$ $P_1=11$
$P_2=6$ $P_4=6$ $P_4=5$ $P_4=2$ $P_3=16$ $P_5=16$
$P_4=6$ $P_5=16$ $P_5=16$ $P_5=2$
$P_5=2$

avg T.AT = 17 ms
avg WT = 8.4ms
avg R.T = 8.4ms

- Non preemptive priority

Priority	Process	A.T	B.T	C.T	TAT	W.T	R.T
1	P_1	0	11	11	11	0	0
0	P_2	0	8	43	43	35	35
3	P_3	12	2	29	17	15	15
2	P_4	2	6	35	33	27	27
4	P_5	9	16	27	18	2	2

P_1	P_5	P_3	P_4	P_2
0	11	27	29	35	43

avg T.AT = 24.4ms
avg WT = 15.8ms
avg R.T = 15.8ms

- Preemptive priority

Priority	Process	A.T	B.T	C.T	TAT	WT	RT
1	P_1	0	11	35	35	24	0
0	P_2	0	8	43	43	35	35
3	P_3	12	2	27	15	13	13
2	P_4	2	6	8	6	0	0
4	P_5	9	16	25	16	0	0

P_1	P_4	P_1	P_5	P_5	P_3	P_1	P_2
0	2	8	9	12	25	27	35	43

avg TAT = 23ms
avg WT = 14.4ms
avg RT = 9.6ms

- Round Robin

Process	Priority	A.T	B.T	C.T	T.AT	W.T	R.T
P_1	1	0	11	32	32	21	0
P_2	0	0	8	28	28	20	5
P_3	3	12	2	30	18	16	28
P_4	2	2	6	31	29	23	10
P_5	4	9	16	43	34	18	20

P_1	P_2	P_4	P_1	P_5	P_2	P_3	P_4	P_1	P_5
0	5	10	15	20	25	28	30	31	32	43

avg TAT = 28.2 ms
avg WT = 19.6ms
avg RT = 12.6 ms.

- Highest response ratio

Process	A.T	B.T	C.T.	T.A.T	W.T	R.T
P_1	0	4	11	11	0	
P_2	3	8	27	27	19	
P_3	12	2	19	7	5	
P_4	2	6	17	15	9	
P_5	9	16	43	34	18	

P_1	P_4	P_3	P_2	P_5
0	11	17	19	27	43

$P_2 = \frac{11+8}{8}$ $P_3 = \frac{17+8}{8}$ $P_2 = \frac{19+8}{8}$

$P_4 = \frac{9+6}{6}$ $P_5 = \frac{5+2}{2}$ $P_5 = \frac{10+16}{16}$

$P_5 = \frac{2+16}{16}$ $P_5 = \frac{8+16}{16}$

avg T.AT. = 18.8

avg W.T. = 10.2

both SJF & SRTF have min. avg. waiting time.

Q8. Consider the set of processes with arrival time, B.T, priority (lowest no. implies high priority)

Process	A.T.	B.T.	Priority
P_1	0	4	3
P_2	0	2	1
P_3	1	3	2
P_4	2	2	4

Find avg T.A.T, avg, W.T & avg R.T

- FCFS
- SJF
- SRTF
- Non preemptive priority.
- Preemptive priority
- Highest response ratio
- round robin ($q = 2$ ms)

→ FCFS

P	A.T.	B.T.	C.T.	T.A.T	W.T	R.T
P_1	0	4	4	4	0	0
P_2	0	2	6	6	4	4
P_3	1	3	9	8	5	5
P_4	2	2	11	9	7	7

P_1	P_2	P_3	P_4
0	4	6	9	11

avg TAT = 6.75 ms

avg WT = 4 ms

avg RT = 4 ms

→ SJF

P	C.T.	T.A.T	W.T	R.T
P_1	11	11	7	7
P_2	2	2	0	0
P_3	7	6	3	3
P_4	4	2	0	0

P_2	P_4	P_3	P_1
0	2	4	7	11

avg TAT = 5.25 ms

avg WT = 2.5 ms

avg RT = 2.5 ms

→ SRTF

P	C.T.	TAT	W.T.	R.T
P_1	11	11	7	7
P_2	2	2	0	0
P_3	7	6	3	3
P_4	4	2	0	0

P_3	P_2	P_4	P_3	P_1
0	1	2	4	7	11

avg TAT = 5.25 ms

avg WT = 2.5 ms

avg RT = 2.5 ms

→ Non Primitive Priority.

P_2	P_3	P_1	P_4
0	2	5	9	11

avg TAT = 6 ms

avg WT = 2.25 ms

avg RT = 3.25 ms

Scanned with OKEN Scanner

Priority	P	C.T.	T.A.T	W.T	R.T
3	P_1	9	9	5	5
1	P_2	2	2	0	0
2	P_3	5	4	1	1
4	P_4	11	9	7	7

→ Preemptive Priority.

P	C.T.	T.A.T	W.T.	R.T
P_1	9	9	5	5
P_2	2	2	0	0
P_3	5	4	1	1
P_4	11	9	7	7

P_1	P_2	P_3	P_3	P_1	P_4

0 1 2 4 5 9 11

avg TAT = 6 ms.
avg WT = 3.25 ms
avg RT = 3.25 ms

→ Highest response ratio

P	A.T.	B.T	C.T.	T.A.T.	W.T.	R.T
P_1	0	4	4	4	0	
P_2	0	2	6	6	4	
P_3	1	3	11	10	7	
P_4	2	2	8	6	4	

P_1	P_2	P_4	P_3

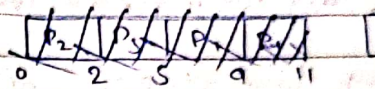
0 4 6 8 11

$P_1 = \frac{4+2}{2}$ $P_2 = \frac{5+2}{2}$
$P_3 = \frac{3+3}{3}$ $P_4 = \frac{4+2}{2}$
$P_4 = \frac{2+2}{2}$

avg TAT = 6.5 ms
avg W.T = 3.75 ms.

→ Round robin ($q = 2$ ms)

Priority	P	A.T.	B.T.	C.T.	T.A.T.	W.T.	R.T.
3	P_1	0	4	10	10	6	
1	P_2	0	2	4	4	2	
2	P_3	1	3	11	10	7	
4	P_4	2	2	8	6	4	



P_1	P_2	P_3	P_4	P_1	P_3

0 2 4 6 8 10 11

$P_1 = 2$ $P_1 = 2$ $P_1 = 2$ $P_1 = 2$
$P_2 = 2$ $P_3 = 3$ $P_3 = 1$ $P_3 = 1$
$P_3 = 3$ $P_4 = 2$ $P_1 = 2$
$P_4 = 2$

Q9. Consider the following 4 processes with arrival time & their length of CPU bursts

P	A.T.	B.T.
P_1	0	3
P_2	1	1
P_3	3	3
P_4	4	x.

Find x. Such avg. W.T. of process 1 ms using SRTF.

Soln: let x = 1

P	A.T.	B.T	C.T	TAT	W.T	R.T
P_1	0	3	4	4	1	0
P_2	1	1	2	1	0	0
P_3	3	3	8	5	2	2
P_4	4	1	5	1	0	0

avg W.T := $\frac{3}{4} = 0.75$

let x = 2.

P	A.T	B.T	C.T	T.AT	W.T	R.T
P_1	0	3	4	4	1	0
P_2	1	1	2	1	0	0

Scanned with OKEN Scanner

P	A.T	B.T	C.T	TAT	W.T	R.T
P_3	3	3	9	6	3	3
P_4	4	2	6	2	0	0

Avg W.T = $\frac{4}{4} = 1$ ms

When x = 2, avg. waiting time is 1 ms.

Q10. Multilevel feedback queue scheduling (MLFQS) with q_1, q_2, q_3. q_1 has highest priority & q_3 has lowest priority. q_1 & q_2 are round robin with time = 8 & 5 ms. q_3 follow FCFS. Find avg. W.T. & avg T.A.T

Sol.

Process	Burst T.	A.T	C.T.	T.A.T	W.T.	R.T
P_1	8	0	17	17	9	0
P_2	22	0	42	42	20	3
P_3	4	0	23	23	19	6
P_4	12	0	46	46	34	9

Q_1 | P_1 | P_2 | P_3 | P_4 |

Q_2 | P_1 | P_2 | P_3 | P_4 |

Q_3 | P_2 | P_4 |

avg T.A.T = 24.5 ms
avg WT = 20.5 ms
avg RT = 4.5 ms

P_1	P_2	P_3	P_4	P_1	P_2	P_3	P_4	P_2	P_4

0 3 6 9 12 17 22 23 28 42 46

Q11. Two procs P & Q. Synchronize the execution process P & Q with suitable semaphore operation using 2 boolean semaphore S & T. Sequence output...

Sol.

```
Process P                Process Q
while(true){             while(true){
wait(S);                 wait(T);
print("1");              print("0");
print("1");              print("0");
Signal(T);               Signal(S);
}
```

initial value S = false
T = true

Q12. Two concurrent Process P & Q are executing following instn. Synchronize execution of P & Q with suitable semaphore operation using 2 binary semaphore S & T.

```
Process P                Process Q
print("1")               print("3")
print("2")               process("4");
```

(a) 1 3 2 4

```
Process P                Process Q.
wait(S)                  wait(T);
print("1");              print("3");
Signal(T);               Signal(S);
wait(S);                 wait(T);
print(2);                print("4");
Signal(T);               signal(S);
```

S = True , T = False.

(b) 3 1 2 4

```
Process P                Process Q.
wait(S);                 wait(T);
print("1");              print(3);
print("2");              Signal(S);
Signal(T);               wait(T);
                         print(4);
                         Signal(S);
```

S = False. , T = True.

Scanned with OKEN Scanner