PART A

1 HELLO, WORLD!

3 creates file.txt if doesn't exist
4 displays all hidden files
5 delete file.txt
6 copy file1.txt to file2.txt
7 moves the file to specified path
8 makes the permissions of owner rwx group r and x and make other users r and x
9 searches pattern in file.txt
10 kills Process with ID PID
11 prints Hello, World!
12 displays ls -l on .txt files
13 concats both files and sorts and prints unique strings
14 it searches for directorys and displays all the permisions by ls -l command
15 Searches recursively for files containing a specific pattern within the specified directory and its subdirectories.
16 Concatenates the contents of file1.txt and file2.txt, sorts the lines alphabetically, and then filters out duplicate lines, displaying only the duplicate lines.
17 Changes the permissions of file.txt to read and write for the owner, and read-only for group and others.
18 Copies the contents of source_directory to destination_directory, including all subdirectories and files.
19 Searches recursively for files with a .txt extension within the specified directory (/path/to/search) and its subdirectories.
20 Adds executable permission for the owner of file.txt.
21 Prints the value of the PATH environment variable, which contains a list of directories where executable files are located. Each directory is separated by a colon.

PART B
1 true
2 mv is used for moving and renaming
3 cd is used to enter a directory
4 pwd print working directory
5 true
6 true
7 true -p directory1/directory2 it makes dire1 into dire2
8 true rm -rf forcefully deletes the file

1 chmod
2 cp

3 mkfile is wrong
4 cat is used
5 mv is used for rename

PART C

```
kushal@Mysekai:~$ vi assign.sh
kushal@Mysekai:~$ chmod +x assign.sh
kushal@Mysekai:~$ ./assign.sh
Hello, World!
enter the string
CDAC  MUMBAI
CDAC  MUMBAI
enter n1
1
enter n2
2
3
kushal@Mysekai:~$ cat assign.sh
#!/bin/bash
echo "Hello, World!"
echo "enter the string"
read string
echo "${string}"
echo "enter n1"
read n1
echo "enter n2"
read n2
(( result = n1 + n2 ))
echo $result
```

```
kushal@Mysekai:~$ ./assloop.sh
enter ur number
2
even
kushal@Mysekai:~$ ./assloop.sh
enter ur number
3
odd
kushal@Mysekai:~$ cat assloop.sh
```

```
#!/bin/bash
echo "enter ur number"
read n1
if((n1%2==0));then

        echo "even"
else
        echo "odd"
fi
```

```
kushal@Mysekai:~$ vi assloop2.sh
kushal@Mysekai:~$ ./assloop2.sh
1
2
3
4
5
kushal@Mysekai:~$ cat assloop2.sg
cat: assloop2.sg: No such file or directory
kushal@Mysekai:~$ cat assloop2.sh
#!/bin/bash
for((i = 1; i<6;i++));do
        echo "$i"
done
```

```
kushal@Mysekai:~$ vi assloop3.sh
kushal@Mysekai:~$ chmod +x assloop3.sh
kushal@Mysekai:~$ ./assloop3.sh
1
2
3
```

```
4
5
kushal@Mysekai:~$ cat assloop3.sh

#!/bin/bash

counter=1

while [ $counter -le 5 ]
do
    echo $counter
    ((counter++))
done
```

```
kushal@Mysekai:~$ vi cond.sh
kushal@Mysekai:~$ chmod +x cond.sh
kushal@Mysekai:~$ ./cond.sh
File exists
kushal@Mysekai:~$ cat cond.sh

#!/bin/bash

# Check if "file.txt" exists in the current directory
if [ -e "file.txt" ]; then
    echo "File exists"
else
    echo "File does not exist"
fi
```

```
kushal@Mysekai:~$ ./cond2.sh
ur number
11
 true
kushal@Mysekai:~$ cat cond2.sh
#!/bin/bash
echo "ur number"
read n1
```

```
if [[ $n1 -gt 10 ]];
then

        echo " true "
else
        echo "false"
fi
```

```
kushal@Mysekai:~$ vi mutable.sh
kushal@Mysekai:~$ ./mutable.sh
 enter ur number
3
 3 x 1 = 3
 3 x 2 = 6
 3 x 3 = 9
 3 x 4 = 12
 3 x 5 = 15
kushal@Mysekai:~$ cat mutable.sh
#!/bin/bash
echo " enter ur number"
read n1
for(( i = 1 ;i < 6; i++));do
        n=$((i*n1))
        echo " $n1 x $i = $n"
done
```

```
kushal@Mysekai:~$ vi loops3.sh
kushal@Mysekai:~$ vi loops3.sh
kushal@Mysekai:~$ chmod +x loops3.sh
kushal@Mysekai:~$ ./loops3.sh
Enter a number (enter a negative number to exit):
2
Square of 2 is: 4
4
```

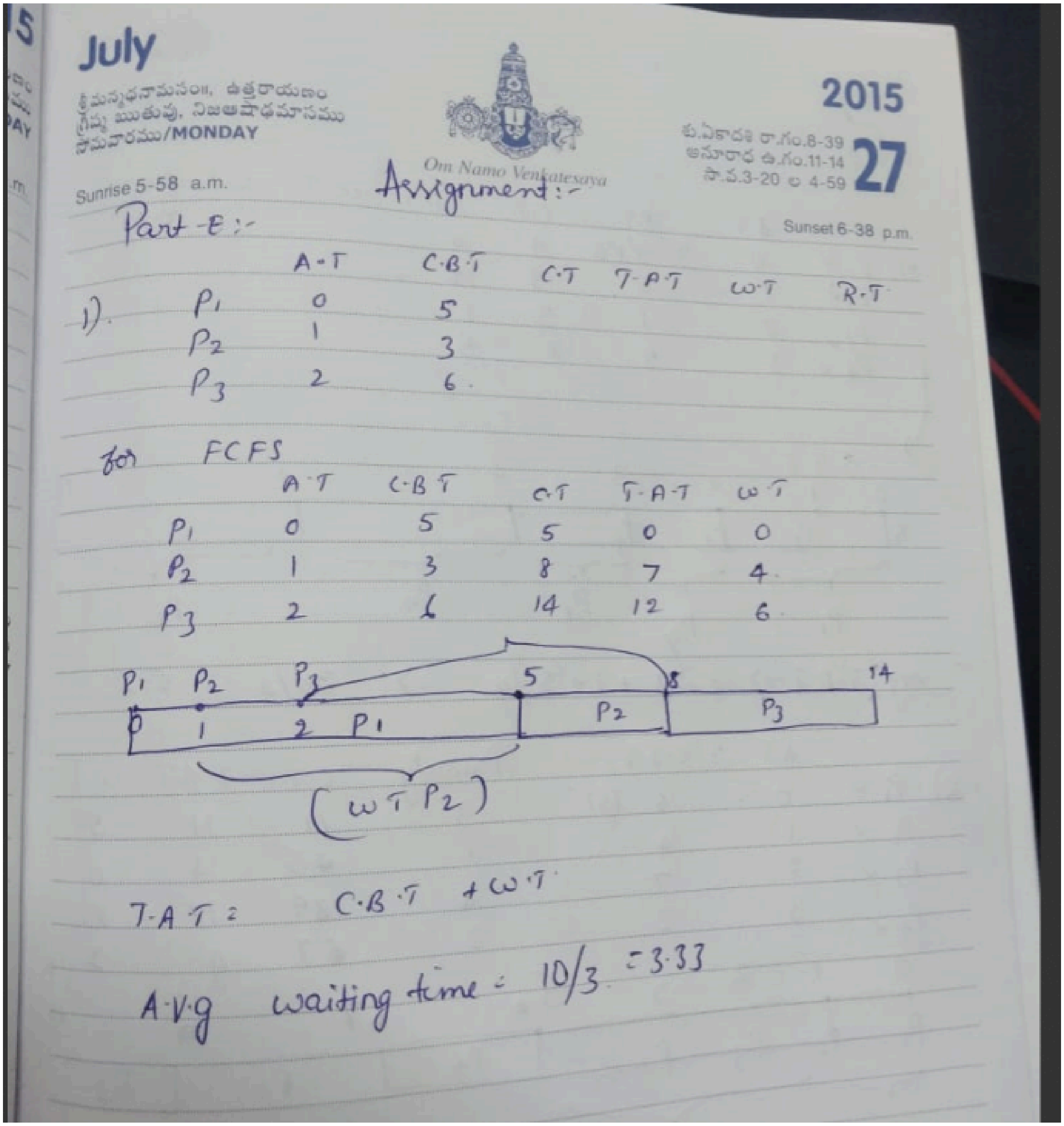
```
Square of 4 is: 16
-2
Exiting the loop.
kushal@Mysekai:~$
```

```bash
#!/bin/bash
echo "Enter a number (enter a negative number to exit):"

while true; do
    read num

    if ((num < 0)); then
        echo "Exiting the loop."
        break
    fi

    square=$((num * num))
    echo "Square of $num is: $square"
done
```



July

శ్రీ మన్మథనామసం॥, ఉత్తరాయణం
గ్రీష్మ ఋతువు, నిజఆషాఢమాసము
సోమవారము/MONDAY

Sunrise 5-58 a.m.

Om Namo Venkatesaya

2015

శు.ఏకాదశి రా.గం.8-39
అనూరాధ ఉ.గం.11-14
సా.వ.3-20 ల 4-59
27

Sunset 6-38 p.m.

Assignment :-

Part -E :-

		A·T	C·B·T	C·T	T·A·T	W·T	R·T
1).	P_1	0	5				
	P_2	1	3				
	P_3	2	6.				

for FCFS

	A·T	C·B·T	C·T	T·A·T	W·T
P_1	0	5	5	0	0
P_2	1	3	8	7	4.
P_3	2	6	14	12	6.

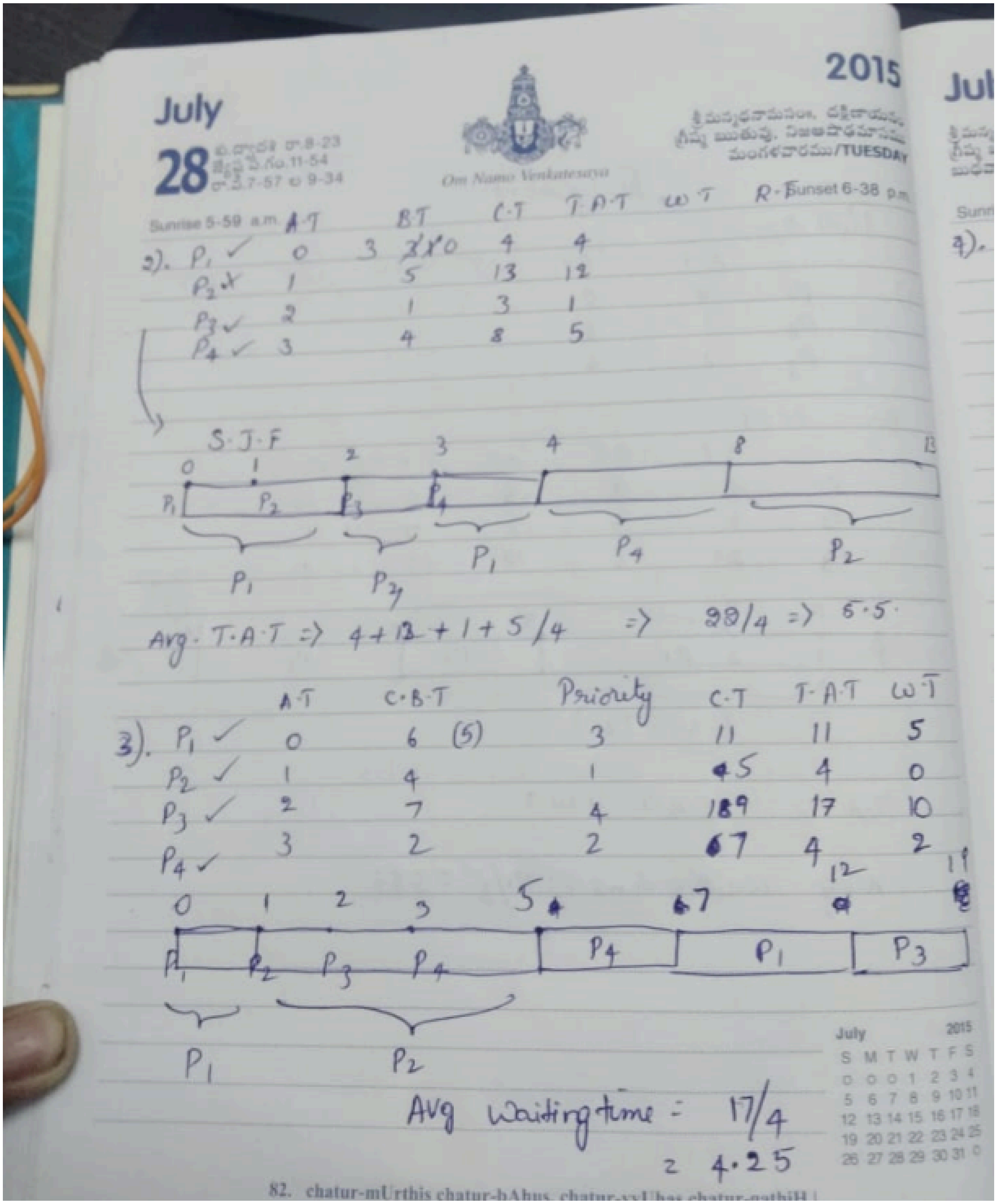
P_1 P_2 P_3 5 8 14

0 1 2 P_1 | P_2 | P_3

(W·T P_2)

T·A·T = C·B·T + W·T

A·V·g waiting time = 10/3 = 3·33



July

28 శు.ద్వాదశి రా.8-23 [illegible] 11-54 [illegible] 7-57 ల 9-34

Om Namo Venkatesaya

2015

శ్రీ మన్మథనామసంవత్సరం, దక్షిణాయనం, గ్రీష్మ ఋతువు, నిజఆషాఢమాసం, మంగళవారము/TUESDAY

Sunrise 5-59 a.m. Sunset 6-38 p.m.

	A.T	B.T	C.T	T.A.T	W.T	R-B
2). P_1 ✓	0	3 ~~3 1~~ 0	4	4		
P_2 ✗	1	5	13	12		
P_3 ✓	2	1	3	1		
P_4 ✓	3	4	8	5		

S.J.F

0 1 2 3 4 8 13

P_1 P_2 | P_3 | P_4 | | |

$\underbrace{\quad}_{P_1}$ $\underbrace{\quad}_{P_3}$ $\underbrace{\quad}_{P_1}$ $\underbrace{\quad}_{P_4}$ $\underbrace{\quad}_{P_2}$

Avg. T.A.T => 4+12+1+5/4 => 22/4 => 5.5.

	A.T	C.B.T	Priority	C.T	T.A.T	W.T
3). P_1 ✓	0	6 (5)	3	11	11	5
P_2 ✓	1	4	1	~~4~~5	4	0
P_3 ✓	2	7	4	1~~8~~9	17	10
P_4 ✓	3	2	2	~~6~~7	4	2

0 1 2 3 5 ~~4~~ ~~6~~7 12 ~~9~~ 19

P_1 | P_2 P_3 P_4 | P_4 | P_1 | P_3

$\underbrace{\quad}_{P_1}$ $\underbrace{\quad\quad\quad}_{P_2}$

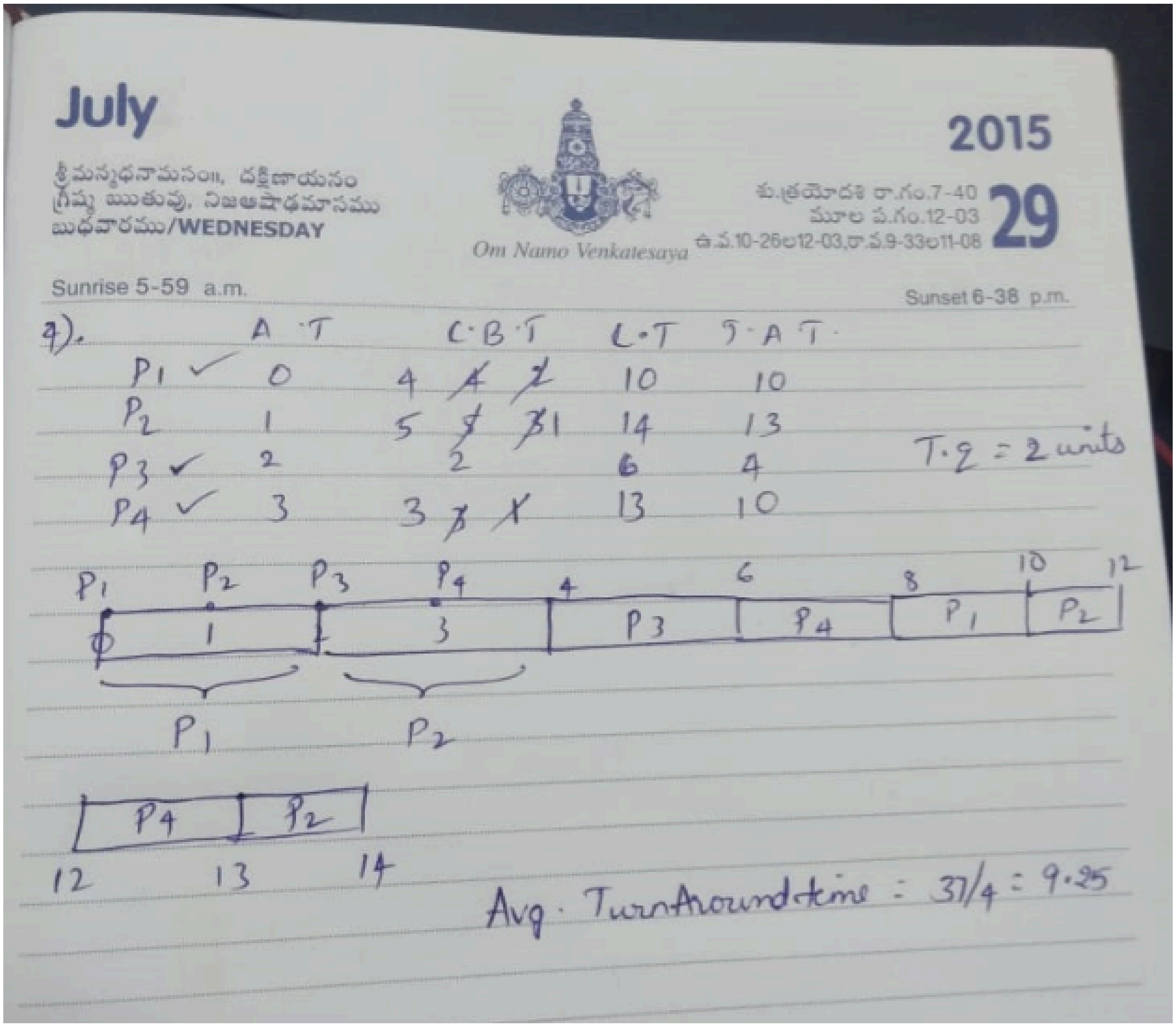
Avg Waiting time = 17/4

= 4.25

July 2015

S	M	T	W	T	F	S
0	0	0	1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	0

82. chatur-mUrthis chatur-bAhus, chatur-vyUhas chatur-gathiH |



July

శ్రీమన్మథనామసం॥, దక్షిణాయనం
గ్రీష్మ ఋతువు, నిజఆషాఢమాసము
బుధవారము/WEDNESDAY

Om Namo Venkatesaya

2015

శు.త్రయోదశి రా.గం.7-40
మూల ప.గం.12-03
ఉ.వ.10-26ల12-03,రా.వ.9-33ల11-08
29

Sunrise 5-59 a.m.

Sunset 6-38 p.m.

4).

	A·T	C·B·T	C·T	T·A·T
P1 ✓	0	4 ~~4~~ ~~2~~	10	10
P2	1	5 ~~5~~ ~~3~~1	14	13
P3 ✓	2	2	6	4
P4 ✓	3	3 ~~3~~ ~~1~~	13	10

T·Q = 2 units

P1 P2 P3 P4 4 6 8 10 12

ϕ 1	3	P3	P4	P1	P2

P1 P2

P4	P2

12 13 14

Avg. Turnaround time : 37/4 = 9.25

5) after fork both the values of parent and child are incremented by 1