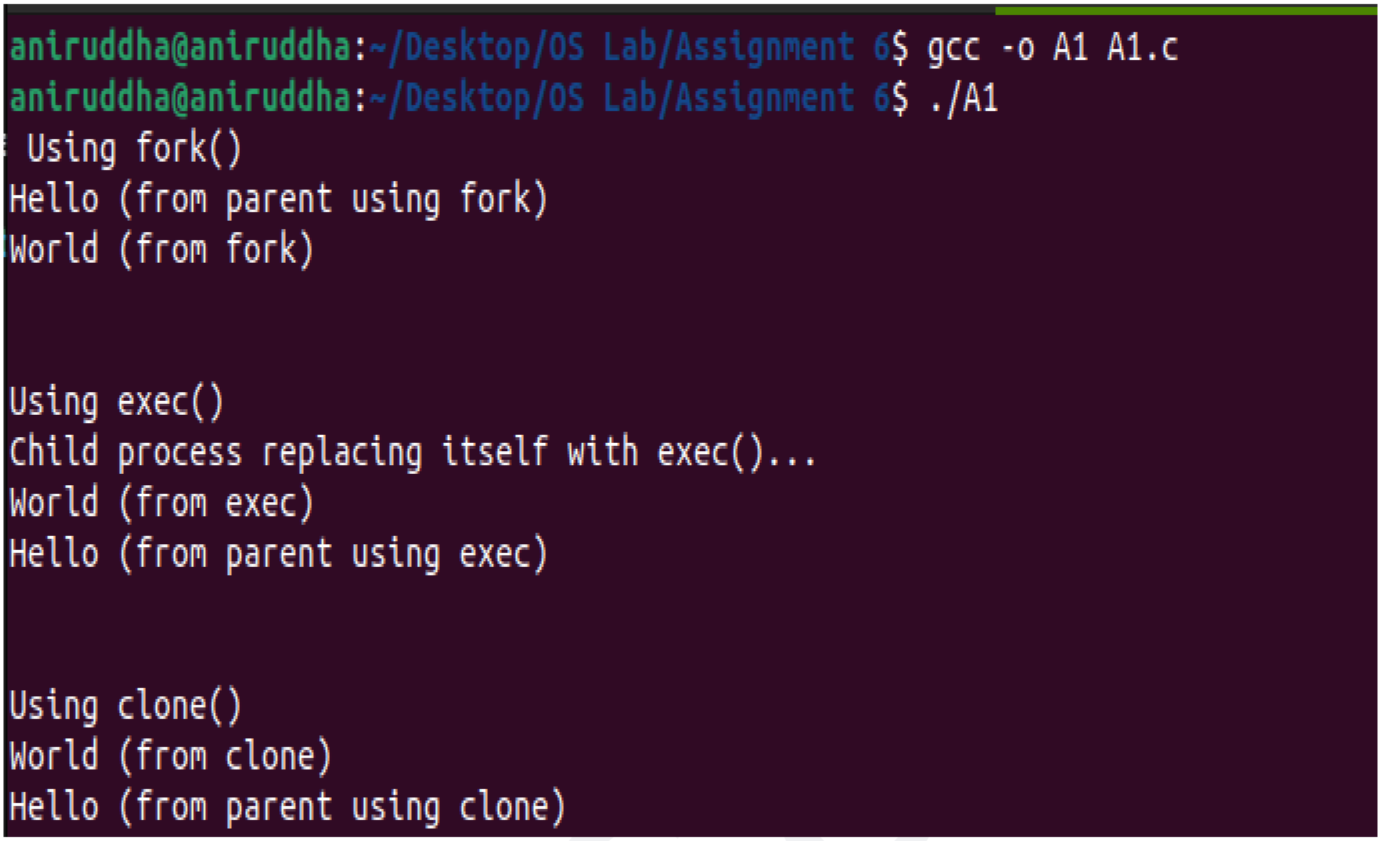
CSS-553 Assignment-5 Outputs

ANIRUDDHA MONDAL

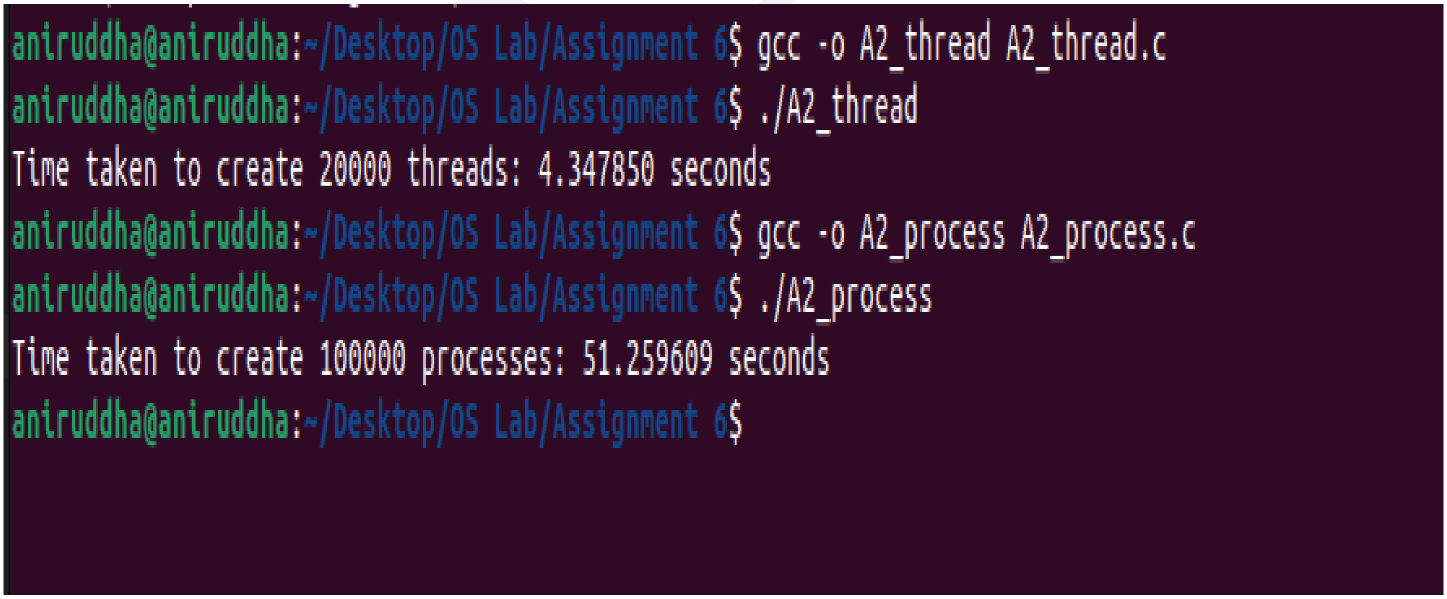
CS-X | 22CS8073

A1.)



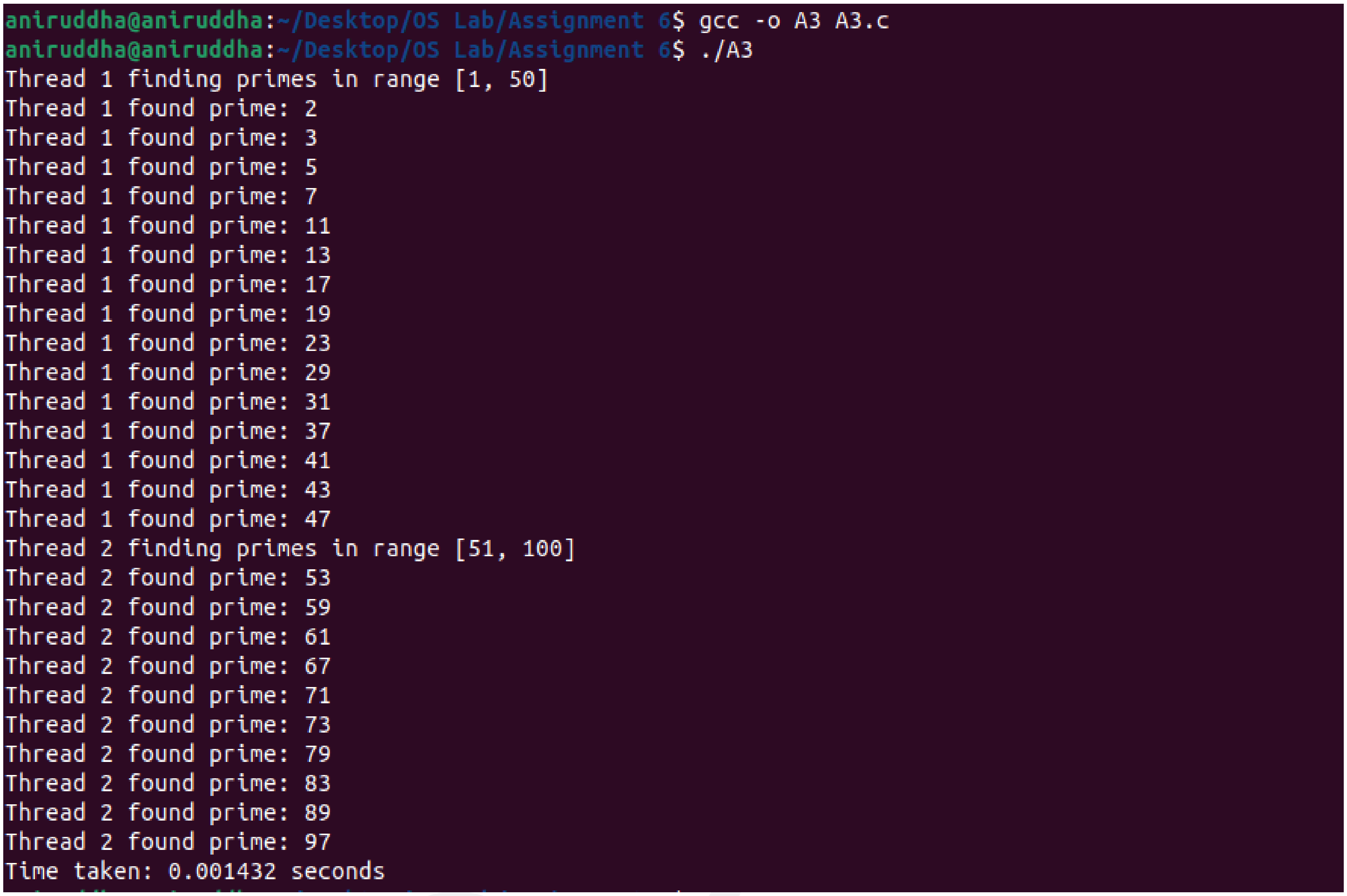
```
aniruddha@aniruddha:~/Desktop/OS Lab/Assignment 6$ gcc -o A1 A1.c
aniruddha@aniruddha:~/Desktop/OS Lab/Assignment 6$ ./A1
 Using fork()
Hello (from parent using fork)
World (from fork)


Using exec()
Child process replacing itself with exec()...
World (from exec)
Hello (from parent using exec)


Using clone()
World (from clone)
Hello (from parent using clone)
```

A2.)



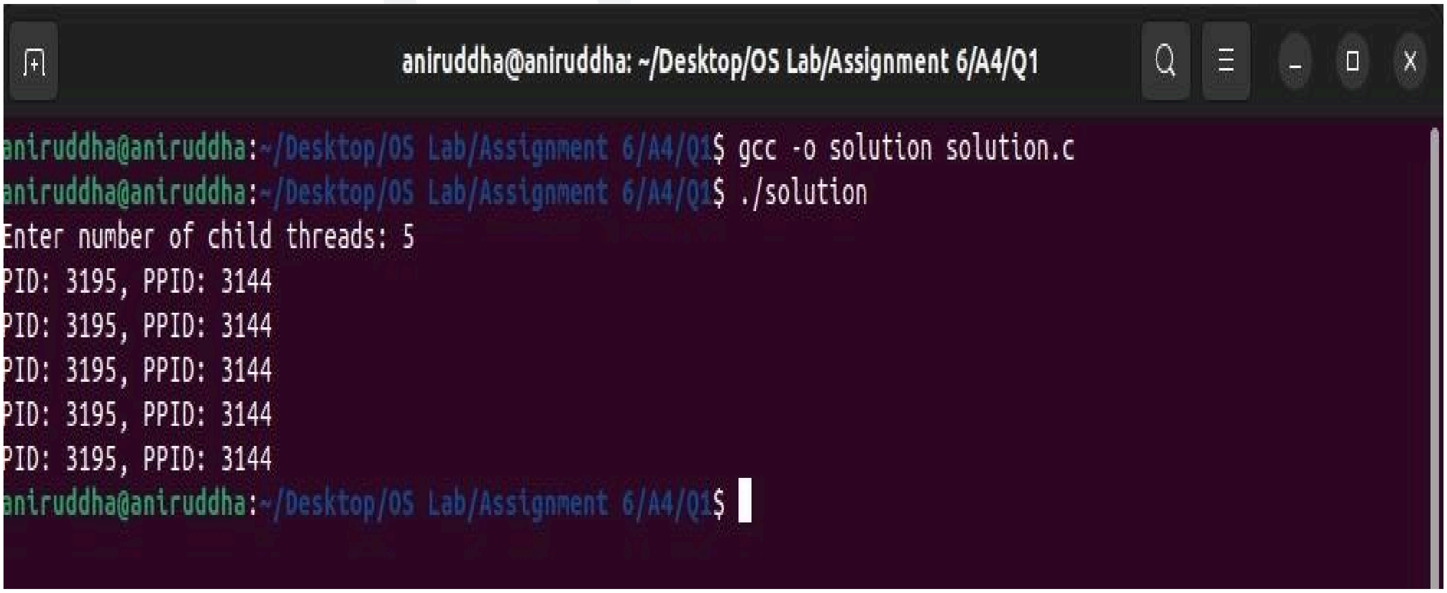
```
aniruddha@aniruddha:~/Desktop/OS Lab/Assignment 6$ gcc -o A2_thread A2_thread.c
aniruddha@aniruddha:~/Desktop/OS Lab/Assignment 6$ ./A2_thread
Time taken to create 20000 threads: 4.347850 seconds
aniruddha@aniruddha:~/Desktop/OS Lab/Assignment 6$ gcc -o A2_process A2_process.c
aniruddha@aniruddha:~/Desktop/OS Lab/Assignment 6$ ./A2_process
Time taken to create 100000 processes: 51.259609 seconds
aniruddha@aniruddha:~/Desktop/OS Lab/Assignment 6$
```

A3.)



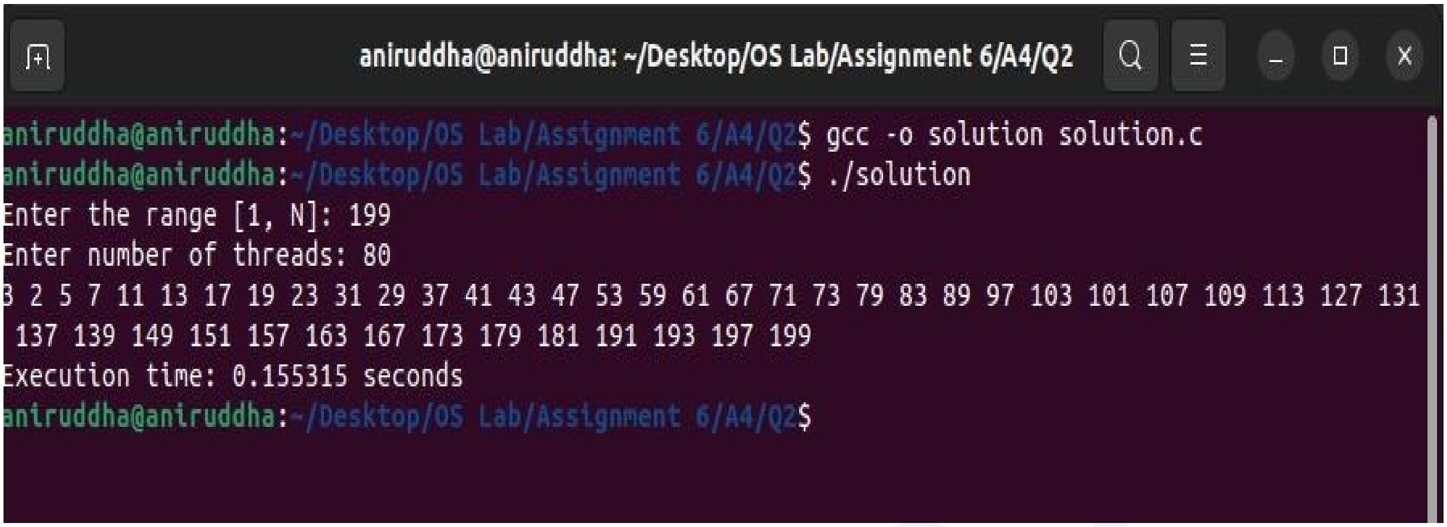
```
aniruddha@aniruddha:~/Desktop/OS Lab/Assignment 6$ gcc -o A3 A3.c
aniruddha@aniruddha:~/Desktop/OS Lab/Assignment 6$ ./A3
Thread 1 finding primes in range [1, 50]
Thread 1 found prime: 2
Thread 1 found prime: 3
Thread 1 found prime: 5
Thread 1 found prime: 7
Thread 1 found prime: 11
Thread 1 found prime: 13
Thread 1 found prime: 17
Thread 1 found prime: 19
Thread 1 found prime: 23
Thread 1 found prime: 29
Thread 1 found prime: 31
Thread 1 found prime: 37
Thread 1 found prime: 41
Thread 1 found prime: 43
Thread 1 found prime: 47
Thread 2 finding primes in range [51, 100]
Thread 2 found prime: 53
Thread 2 found prime: 59
Thread 2 found prime: 61
Thread 2 found prime: 67
Thread 2 found prime: 71
Thread 2 found prime: 73
Thread 2 found prime: 79
Thread 2 found prime: 83
Thread 2 found prime: 89
Thread 2 found prime: 97
Time taken: 0.001432 seconds
```

A4 Question 1.)



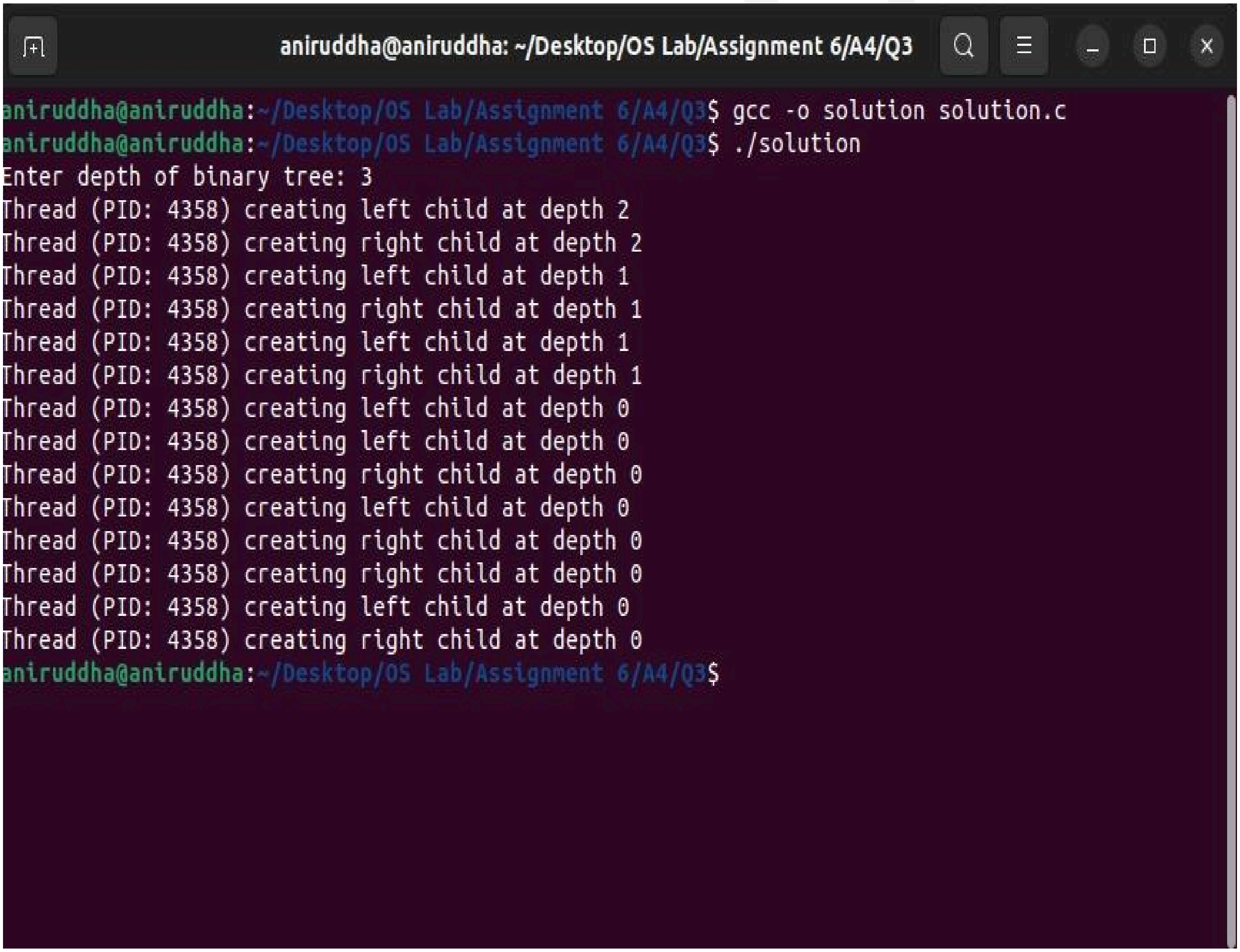
```
aniruddha@aniruddha:~/Desktop/OS Lab/Assignment 6/A4/Q1$ gcc -o solution solution.c
aniruddha@aniruddha:~/Desktop/OS Lab/Assignment 6/A4/Q1$ ./solution
Enter number of child threads: 5
PID: 3195, PPID: 3144
PID: 3195, PPID: 3144
PID: 3195, PPID: 3144
PID: 3195, PPID: 3144
PID: 3195, PPID: 3144
aniruddha@aniruddha:~/Desktop/OS Lab/Assignment 6/A4/Q1$
```

A4 Question 2.)



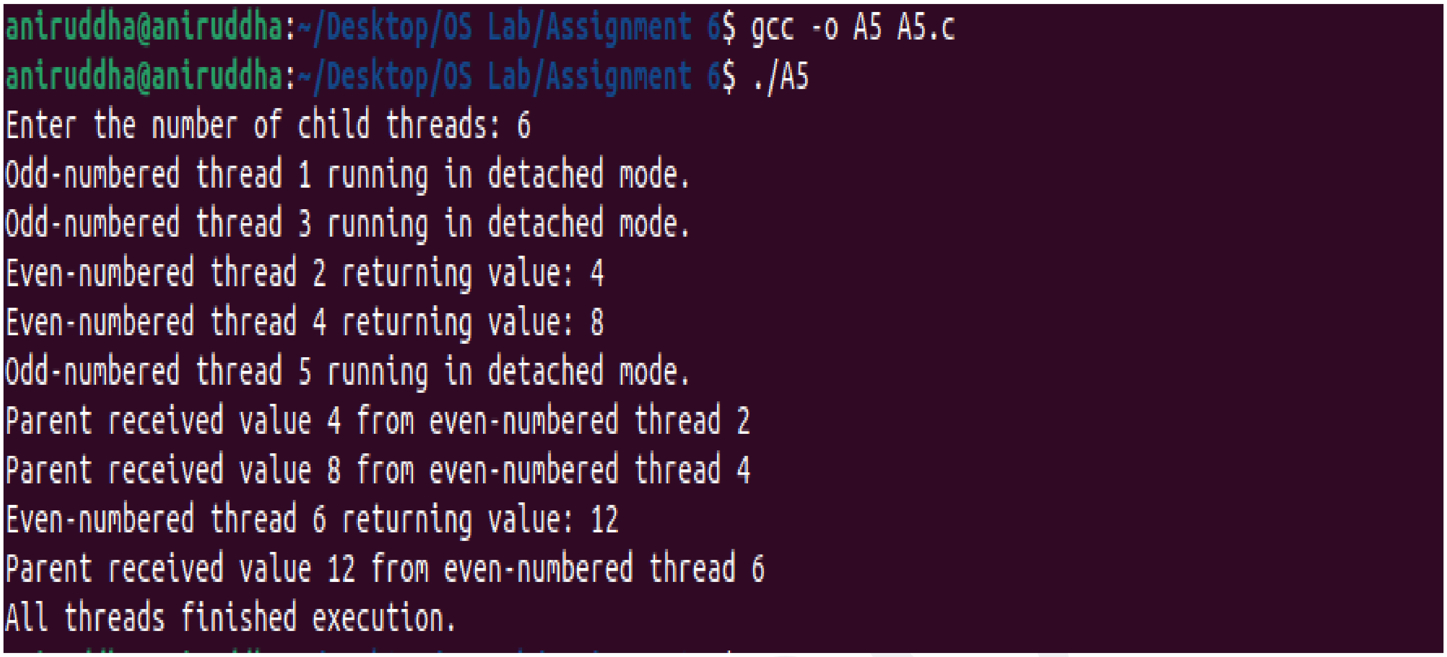
```
aniruddha@aniruddha:~/Desktop/OS Lab/Assignment 6/A4/Q2$ gcc -o solution solution.c
aniruddha@aniruddha:~/Desktop/OS Lab/Assignment 6/A4/Q2$ ./solution
Enter the range [1, N]: 199
Enter number of threads: 80
3 2 5 7 11 13 17 19 23 31 29 37 41 43 47 53 59 61 67 71 73 79 83 89 97 103 101 107 109 113 127 131
 137 139 149 151 157 163 167 173 179 181 191 193 197 199
Execution time: 0.155315 seconds
aniruddha@aniruddha:~/Desktop/OS Lab/Assignment 6/A4/Q2$
```

A4 Question 3.)



```
aniruddha@aniruddha:~/Desktop/OS Lab/Assignment 6/A4/Q3$ gcc -o solution solution.c
aniruddha@aniruddha:~/Desktop/OS Lab/Assignment 6/A4/Q3$ ./solution
Enter depth of binary tree: 3
Thread (PID: 4358) creating left child at depth 2
Thread (PID: 4358) creating right child at depth 2
Thread (PID: 4358) creating left child at depth 1
Thread (PID: 4358) creating right child at depth 1
Thread (PID: 4358) creating left child at depth 1
Thread (PID: 4358) creating right child at depth 1
Thread (PID: 4358) creating left child at depth 0
Thread (PID: 4358) creating left child at depth 0
Thread (PID: 4358) creating right child at depth 0
Thread (PID: 4358) creating left child at depth 0
Thread (PID: 4358) creating right child at depth 0
Thread (PID: 4358) creating right child at depth 0
Thread (PID: 4358) creating left child at depth 0
Thread (PID: 4358) creating right child at depth 0
aniruddha@aniruddha:~/Desktop/OS Lab/Assignment 6/A4/Q3$
```

A5.)



```
aniruddha@aniruddha:~/Desktop/OS Lab/Assignment 6$ gcc -o A5 A5.c
aniruddha@aniruddha:~/Desktop/OS Lab/Assignment 6$ ./A5
Enter the number of child threads: 6
Odd-numbered thread 1 running in detached mode.
Odd-numbered thread 3 running in detached mode.
Even-numbered thread 2 returning value: 4
Even-numbered thread 4 returning value: 8
Odd-numbered thread 5 running in detached mode.
Parent received value 4 from even-numbered thread 2
Parent received value 8 from even-numbered thread 4
Even-numbered thread 6 returning value: 12
Parent received value 12 from even-numbered thread 6
All threads finished execution.
```

Thanks

For

Reviewing