SOMANATHAN SUBRAMANIYAN

STUDENT ID: G00364742

Computational Thinking with Algorithms

Problem Sheet Answers

2019

Table of Contents

[Q1: Question 2](#_Toc1910508)

[Q1. Answer 2](#_Toc1910509)

[Q1.2. Question 2](#_Toc1910510)

[Q1.2. Answer 3](#_Toc1910511)

[Q1.3. Question 3](#_Toc1910512)

[Q1.3. Answer 3](#_Toc1910513)

[Q1.4. Question 4](#_Toc1910514)

[Q1.4. Answer 4](#_Toc1910515)

[Q1.5. Question 5](#_Toc1910516)

[Q1.5. Answer 5](#_Toc1910517)

[Q2: Linux Assignment 5](#_Toc1910518)

[Q2.1: Question 5](#_Toc1910519)

[Q2.1: Answer 5](#_Toc1910520)

[Q2.2: Questions 7](#_Toc1910521)

[Q2.2: Answer 7](#_Toc1910522)

[1. Shell Script 7](#_Toc1910523)

[2. Commands executed on the terminal 9](#_Toc1910524)

[3. Output of the Shell Script: 9](#_Toc1910525)

[Q2.3 11](#_Toc1910526)

[Q2.3.1: Question 11](#_Toc1910527)

[Q2.3.1: Answer 11](#_Toc1910528)

[Q2.3.2: Question 12](#_Toc1910529)

[Q2.3.2: Answer 12](#_Toc1910530)

[Q2.4. Question 14](#_Toc1910531)

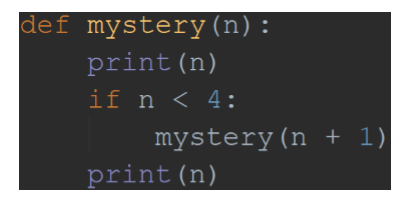
[Q2.4. Answer 15](#_Toc1910532)

[1. Shell Script 15](#_Toc1910533)

[2. Screenshots 15](#_Toc1910534)

# Q1: Question

Consider the following method:



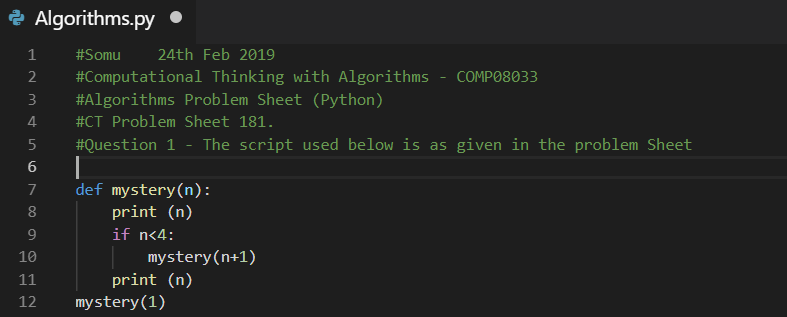
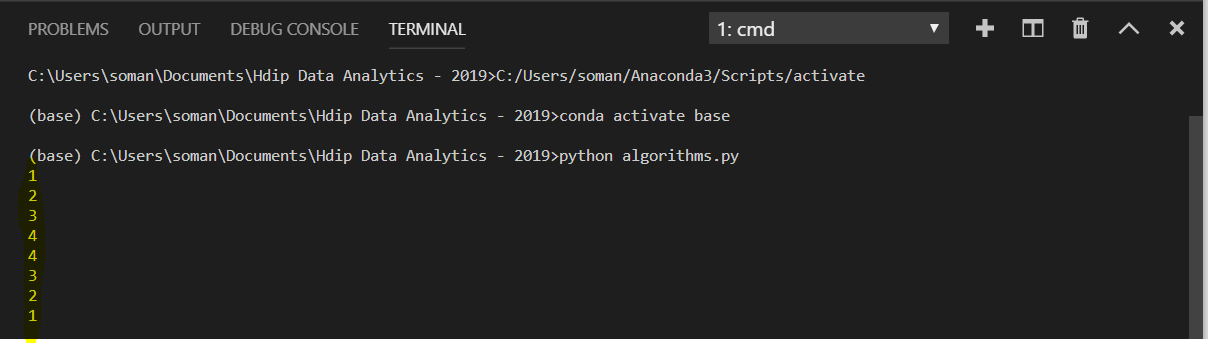
What will the output of the call mystery(1) be?

Write an explanation of the reasoning behind your answer, using the aid of either a recursion trace diagram or a stack diagram. Include any code which you write for testing or explanation purposes as part of your answer.

# Q1. Answer

The output of the given python code is 1 2 3 4

Modified Python Code: Algorithms.py execution output in python



**Output**:

**1 2 3 4 4 3 2 1**

Recursion Diagram

* The initial call to the function “Mystery” is made by the statement in line#12 “mystery(1)”
* The function call “mystery(1)” in- turn calls function “Mystery” with an argument value of “2” in line #10
* The function call “mystery(2)” in- turn calls function “Mystery” with an argument value of “3” in line #10
* The function call “mystery(3)” in- turn calls function “Mystery” with an argument value of “4” in line #10
* The function call “mystery(4)” doesn’t meet the conditional statement in line #9 and returns the control to function call “mystery(4)”
* “mystery(3)” returns the control to “mystery(2)”
* “mystery(2)” returns the control to “mystery(1)”
* “mystery(1)” completes the execution of the function.

Mystery (1)

Mystery (2)

Mystery (3)

Mystery (4)

Return

Return

Return

Call

Call

Call

Call

# Q2: Question 2

## Q2 (a) Question

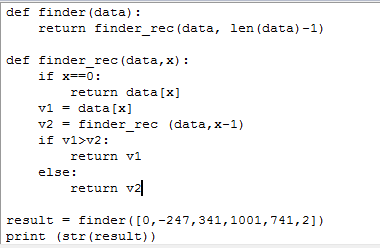
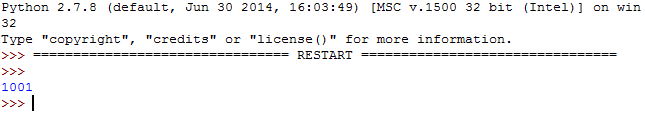
## What value is returned by a call to finder when the following array is used as input?

## [0,-247, 341,1001,741,22]

## Answer

The answer is **1001**

Python Code: Output:



## Q2 (b) Question

What characteristic of the input data set does the finder method determine? How does it determine the results?

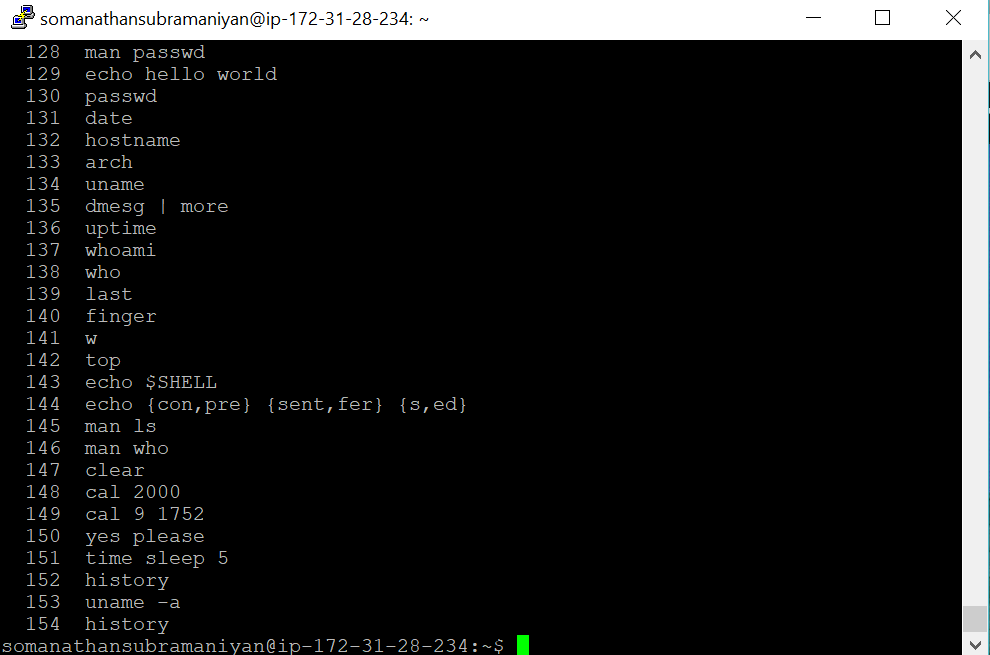
* The finder method determines the Largest data element in the input data set
* The largest data element is identified by comparing the each data element against the other data element in the

**Sequence of Actions with an example**

Example input data set :: [1,3,4]

* The “finder” method calls function “finder\_rec” with input data set and the length/ count of input data set
* Function “finder\_rec” in turn recursively calls the “finder\_rec” method with input data set and count of input data set, until count is zero
* The “v1”variable is set to 3 and 4 by the recursive function call
* The “v2” variable is set to 1 by the last recursive function call
  + finder\_rec ([1,3,4] , 2) v1 -> 4 Call 1
  + finder\_rec ([1,3,4] , 1) v1-> 3 Call 2
  + finder\_rec ([1,3,4] , 0) v2-> 1 Call 3
* When the count parameter is “0” in the “finder\_rec” function, the if statement exists from the function “finder\_rec”
* The control goes back to the “Call 2”
  + The variable values at this point are v1 = 3 and v2 = 1
  + As v1 is greater than v2, the value 3 is assigned to variable v2 by “finder\_rec” function
* The control goes back to the “Call 1”
  + The variable values at this point are v1 = 4 and v2 = 3
  + As v1 is greater than v2, the value 4 is assigned to variable v2 by “finder\_rec” function
* The method “finder” returns the value “4”

Screenshot of the History Command



## Q2.2: Questions

This is a research project. Use Google to help you identify a solution.

For each of the commands marked with an \*, group them into a shell script so that you can automate execution of the commands. Write the shell script using the Vim text editor.

Once you have verified that the script works, add output redirection to append the output of each command to a file named as follows: firstnameSurname.txt (replacing firstname and surname with your own details). When writing to this text file, make liberal use of the echo command within the shell script to format the output nicely – i.e. insert blank lines or other demarcations and headings to make your file easily readable.

For the submission: Copy and paste the contents of this auto-created “.txt” file into your “.docx” document for submission. You are also required to upload the shell script which you wrote and the text file which it generated.

## Q2.2: Answer

### Shell Script

|  |
| --- |
|  |
|  | #!/bin/bash  HISTFILE=~/.bash\_history |
|  | set -o history |
|  | var1=$(date) |
|  | var2=$(hostname) |
|  | var3=$(arch) |
|  | var4=$(uptime) |
|  | var5=$(whoami) |
|  | var6=$(who) |
|  | var7=$(finger) |
|  | var8=$(w) |
|  | var9=$(top -n 1 -b) |
|  | var10=$(history 10) |
|  | echo " " |
|  | echo "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" |
|  | echo " " |
|  | echo "Current System Date :" "$var1" |
|  | echo "The hostname is :" "$var2" |
|  | echo "The Architecture of the hostsystem is :" "$var3" |
|  | echo "The uptime of this computer/system is :" "$var4" |
|  | echo "Result of the whoami command :" "$var5" |
|  | echo "" |
|  | echo "" |
|  | echo "\*\*\*\*The results of who command starts\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" |
|  | echo "" |
|  | echo "$var6" |
|  | echo "" |
|  | echo "\*\*\*The results of who command ends\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" |
|  | echo "" |
|  | echo "\*\*\*The reults of Finger Command starts\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" |
|  | echo "" |
|  | echo "$var7" |
|  | echo "" |
|  | echo "\*\*\*The results of Finger Command ends \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" |
|  | echo "" |
|  | echo "\*\*\*The reults of w Command starts\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" |
|  | echo "" |
|  | echo "$var8" |
|  | echo "" |
|  | echo "\*\*\*The reults of w Command ends \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" |
|  | echo "" |
|  | echo "\*\*\*\*\*\*The top command results starts \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" |
|  | echo "" |
|  | echo "$var9" |
|  | echo "" |
|  | echo "\*\*\*\*\*The top command results ends\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" |
|  | echo "" |
|  | echo "\*\*\*\*\*The History command starts \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" |
|  | echo "" |
|  | echo "$var10" |
|  | echo "" |
|  | echo "\*\*\*\*\*The history command ends \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" |
|  | echo "" |
|  | Exit |

### Commands executed on the terminal

The below commands are executed on the terminal

* ./project.sh > SomanathanSubramaniyan.txt
* awk 'sub("$","\r")' SomanathanSubramaniyan.txt > win.txt

win.txt renamed into SomanathanSubramaniyan.txt to load into project repository.

### Output of the Shell Script:

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Current System Date is : Sun Apr 29 07:05:16 UTC 2018

The hostname is : ip-172-31-28-234

The Architecture of the hostsystem is : x86\_64

The uptime of this computer/system is : 07:05:16 up 48 days, 7:34, 1 user, load average: 0.00, 0.00, 0.00

Result of the whoami command : somanathansubramaniyan

\*\*\*\*The result of who command execution starts\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

somanathansubramaniyan pts/0 2018-04-29 06:58 (37.228.255.50)

\*\*\*The result of who command execution ends\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*The result of Finger Command execution starts\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Login Name Tty Idle Login Time Office Office Phone

somanathansubramaniyan pts/0 Apr 29 06:58 (37.228.255.50)

\*\*\*The result of Finger Command execution ends \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*The result of w Command execution starts\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

07:05:16 up 48 days, 7:34, 1 user, load average: 0.00, 0.00, 0.00

USER TTY FROM LOGIN@ IDLE JCPU PCPU WHAT

somanath pts/0 37.228.255.50 06:58 4.00s 0.04s 0.00s w

\*\*\*The result of w Command execution ends \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*The top command execution result starts \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

top - 07:05:16 up 48 days, 7:34, 1 user, load average: 0.00, 0.00, 0.00

Tasks: 108 total, 1 running, 107 sleeping, 0 stopped, 0 zombie

%Cpu(s): 0.1 us, 0.0 sy, 0.0 ni, 99.9 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st

KiB Mem : 1014552 total, 89904 free, 52780 used, 871868 buff/cache

KiB Swap: 0 total, 0 free, 0 used. 740968 avail Mem

PID USER PR NI VIRT RES SHR S %CPU %MEM TIME+ COMMAND

1 root 20 0 119820 5392 3456 S 0.0 0.5 0:40.85 systemd

2 root 20 0 0 0 0 S 0.0 0.0 0:00.01 kthreadd

3 root 20 0 0 0 0 S 0.0 0.0 0:16.59 ksoftirqd/0

5 root 0 -20 0 0 0 S 0.0 0.0 0:00.00 kworker/0:0H

7 root 20 0 0 0 0 S 0.0 0.0 0:28.64 rcu\_sched

8 root 20 0 0 0 0 S 0.0 0.0 0:00.00 rcu\_bh

9 root rt 0 0 0 0 S 0.0 0.0 0:00.00 migration/0

10 root rt 0 0 0 0 S 0.0 0.0 0:19.26 watchdog/0

11 root 20 0 0 0 0 S 0.0 0.0 0:00.00 kdevtmpfs

12 root 0 -20 0 0 0 S 0.0 0.0 0:00.00 netns

13 root 0 -20 0 0 0 S 0.0 0.0 0:00.00 perf

14 root 20 0 0 0 0 S 0.0 0.0 0:00.00 xenwatch

15 root 20 0 0 0 0 S 0.0 0.0 0:00.00 xenbus

17 root 20 0 0 0 0 S 0.0 0.0 0:00.80 khungtaskd

18 root 0 -20 0 0 0 S 0.0 0.0 0:00.00 writeback

19 root 25 5 0 0 0 S 0.0 0.0 0:00.00 ksmd

20 root 39 19 0 0 0 S 0.0 0.0 0:07.43 khugepaged

21 root 0 -20 0 0 0 S 0.0 0.0 0:00.00 crypto

22 root 0 -20 0 0 0 S 0.0 0.0 0:00.00 kintegrityd

23 root 0 -20 0 0 0 S 0.0 0.0 0:00.00 bioset

24 root 0 -20 0 0 0 S 0.0 0.0 0:00.00 kblockd

25 root 0 -20 0 0 0 S 0.0 0.0 0:00.00 ata\_sff

26 root 0 -20 0 0 0 S 0.0 0.0 0:00.00 md

27 root 0 -20 0 0 0 S 0.0 0.0 0:00.00 devfreq\_wq

30 root 20 0 0 0 0 S 0.0 0.0 0:00.53 kswapd0

31 root 0 -20 0 0 0 S 0.0 0.0 0:00.00 vmstat

32 root 20 0 0 0 0 S 0.0 0.0 0:00.00 fsnotify\_ma+

33 root 20 0 0 0 0 S 0.0 0.0 0:00.00 ecryptfs-kt+

49 root 0 -20 0 0 0 S 0.0 0.0 0:00.00 kthrotld

50 root 0 -20 0 0 0 S 0.0 0.0 0:00.00 bioset

51 root 0 -20 0 0 0 S 0.0 0.0 0:00.00 bioset

52 root 0 -20 0 0 0 S 0.0 0.0 0:00.00 bioset

53 root 0 -20 0 0 0 S 0.0 0.0 0:00.00 bioset

54 root 0 -20 0 0 0 S 0.0 0.0 0:00.00 bioset

55 root 0 -20 0 0 0 S 0.0 0.0 0:00.00 bioset

56 root 0 -20 0 0 0 S 0.0 0.0 0:00.00 bioset

57 root 0 -20 0 0 0 S 0.0 0.0 0:00.00 bioset

58 root 0 -20 0 0 0 S 0.0 0.0 0:00.00 bioset

59 root 0 -20 0 0 0 S 0.0 0.0 0:00.00 bioset

60 root 0 -20 0 0 0 S 0.0 0.0 0:00.00 bioset

61 root 0 -20 0 0 0 S 0.0 0.0 0:00.00 bioset

62 root 0 -20 0 0 0 S 0.0 0.0 0:00.00 bioset

63 root 0 -20 0 0 0 S 0.0 0.0 0:00.00 bioset

64 root 0 -20 0 0 0 S 0.0 0.0 0:00.00 bioset

65 root 0 -20 0 0 0 S 0.0 0.0 0:00.00 bioset

66 root 0 -20 0 0 0 S 0.0 0.0 0:00.00 bioset

67 root 0 -20 0 0 0 S 0.0 0.0 0:00.00 bioset

68 root 0 -20 0 0 0 S 0.0 0.0 0:00.00 bioset

69 root 0 -20 0 0 0 S 0.0 0.0 0:00.00 bioset

70 root 0 -20 0 0 0 S 0.0 0.0 0:00.00 bioset

71 root 0 -20 0 0 0 S 0.0 0.0 0:00.00 bioset

72 root 0 -20 0 0 0 S 0.0 0.0 0:00.00 bioset

73 root 0 -20 0 0 0 S 0.0 0.0 0:00.00 bioset

74 root 20 0 0 0 0 S 0.0 0.0 0:00.00 scsi\_eh\_0

75 root 0 -20 0 0 0 S 0.0 0.0 0:00.00 scsi\_tmf\_0

76 root 20 0 0 0 0 S 0.0 0.0 0:00.00 scsi\_eh\_1

77 root 0 -20 0 0 0 S 0.0 0.0 0:00.00 scsi\_tmf\_1

79 root 0 -20 0 0 0 S 0.0 0.0 0:00.00 bioset

83 root 0 -20 0 0 0 S 0.0 0.0 0:00.00 ipv6\_addrco+

96 root 0 -20 0 0 0 S 0.0 0.0 0:00.00 deferwq

258 root 0 -20 0 0 0 S 0.0 0.0 0:00.00 raid5wq

289 root 0 -20 0 0 0 S 0.0 0.0 0:00.00 bioset

310 root 20 0 0 0 0 S 0.0 0.0 0:07.04 jbd2/xvda1-8

311 root 0 -20 0 0 0 S 0.0 0.0 0:00.00 ext4-rsv-co+

358 root 0 -20 0 0 0 S 0.0 0.0 0:01.83 kworker/0:1H

396 root 20 0 28428 3208 2856 S 0.0 0.3 4:55.79 systemd-jou+

398 root 0 -20 0 0 0 S 0.0 0.0 0:00.00 iscsi\_eh

401 root 0 -20 0 0 0 S 0.0 0.0 0:00.00 ib\_addr

403 root 0 -20 0 0 0 S 0.0 0.0 0:00.00 ib\_mcast

405 root 0 -20 0 0 0 S 0.0 0.0 0:00.00 ib\_nl\_sa\_wq

406 root 0 -20 0 0 0 S 0.0 0.0 0:00.00 ib\_cm

407 root 0 -20 0 0 0 S 0.0 0.0 0:00.00 iw\_cm\_wq

408 root 0 -20 0 0 0 S 0.0 0.0 0:00.00 rdma\_cm

410 root 20 0 0 0 0 S 0.0 0.0 0:00.00 kauditd

436 root 20 0 102968 1296 1112 S 0.0 0.1 0:00.00 lvmetad

952 root 20 0 16120 2812 1952 S 0.0 0.3 0:01.35 dhclient

1130 root 20 0 5220 116 0 S 0.0 0.0 1:47.64 iscsid

1131 root 10 -10 5720 3512 2428 S 0.0 0.3 8:20.08 iscsid

1135 message+ 20 0 43016 3368 2772 S 0.0 0.3 0:05.16 dbus-daemon

1139 daemon 20 0 26044 1948 1752 S 0.0 0.2 0:00.04 atd

1158 root 20 0 275608 6036 4252 S 0.0 0.6 3:16.58 accounts-da+

1162 root 20 0 28616 2948 2616 S 0.0 0.3 0:03.99 systemd-log+

1165 root 20 0 65508 4320 3600 S 0.0 0.4 0:59.06 sshd

1168 syslog 20 0 260628 3640 2056 S 0.0 0.4 1:18.20 rsyslogd

1171 root 20 0 27728 2460 2188 S 0.0 0.2 0:04.75 cron

1174 root 20 0 636844 5348 1452 S 0.0 0.5 0:15.03 lxcfs

1177 root 20 0 4396 1232 1144 S 0.0 0.1 0:00.00 acpid

1184 root 20 0 281460 14404 3848 S 0.0 1.4 1:40.16 snapd

1205 root 20 0 13372 156 20 S 0.0 0.0 0:00.22 mdadm

1207 root 20 0 277176 4964 4292 S 0.0 0.5 0:00.90 polkitd

1280 root 20 0 14472 1524 1392 S 0.0 0.2 0:00.00 agetty

1283 root 20 0 14656 1508 1380 S 0.0 0.1 0:00.03 agetty

5831 root 20 0 0 0 0 S 0.0 0.0 0:06.16 kworker/u30+

13036 systemd+ 20 0 100324 2332 2116 S 0.0 0.2 0:02.60 systemd-tim+

14927 root 20 0 42120 2580 2104 S 0.0 0.3 0:02.22 systemd-ude+

19172 root 0 -20 0 0 0 S 0.0 0.0 0:00.00 xfsalloc

19173 root 0 -20 0 0 0 S 0.0 0.0 0:00.00 xfs\_mru\_cac+

22264 root 20 0 0 0 0 S 0.0 0.0 0:00.63 kworker/u30+

25335 root 20 0 0 0 0 S 0.0 0.0 0:00.50 kworker/0:3

27096 root 20 0 0 0 0 S 0.0 0.0 0:00.00 kworker/0:1

27548 root 20 0 92828 6904 5972 S 0.0 0.7 0:00.00 sshd

27550 somanat+ 20 0 45276 4612 3908 S 0.0 0.5 0:00.00 systemd

27551 somanat+ 20 0 143272 2048 0 S 0.0 0.2 0:00.00 (sd-pam)

27609 somanat+ 20 0 92828 3396 2468 S 0.0 0.3 0:00.12 sshd

27610 somanat+ 20 0 21396 5380 3400 S 0.0 0.5 0:00.04 bash

27690 root 20 0 65508 5808 5104 S 0.0 0.6 0:00.00 sshd

27691 somanat+ 20 0 11340 3124 2820 S 0.0 0.3 0:00.00 project.sh

27700 somanat+ 20 0 40388 3644 3140 R 0.0 0.4 0:00.00 top

\*\*\*\*\*The top command execution result ends\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*The History command execution starts \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

501 var1=$(date)

502 var2=$(hostname)

503 var3=$(arch)

504 var4=$(uptime)

505 var5=$(whoami)

506 var6=$(who)

507 var7=$(finger)

508 var8=$(w)

509 var9=$(top -n 1 -b)

510 var10=$(history 10)

\*\*\*\*\*The history command execution ends \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

## Q2.3

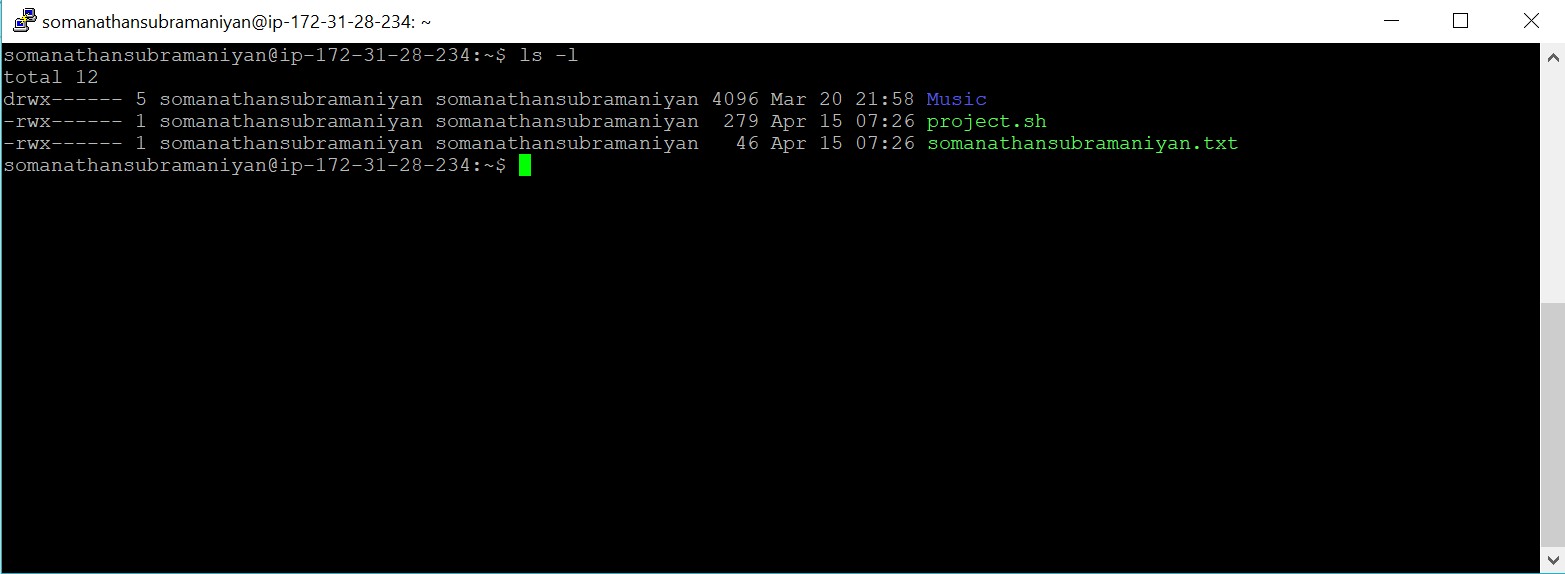
### Q2.3.1: Question

When a user account is created on Linux, it is public to all users of the machine by default. That is, anyone can view your personal files. Change the access permissions using the ‘chmod’ command so that your personal folder is fully accessible to you (read, write and execute privileges) and totally inaccessible to the group and all other users. Look at the lecture notes to work out how to do this. Run the “ls –l” command to verify that the permissions have been set correctly (Screenshot the result from this for submission).

### Q2.3.1: Answer

Command User: chmod -R 700 /home/somanathansubramaniyan

Screenshot:

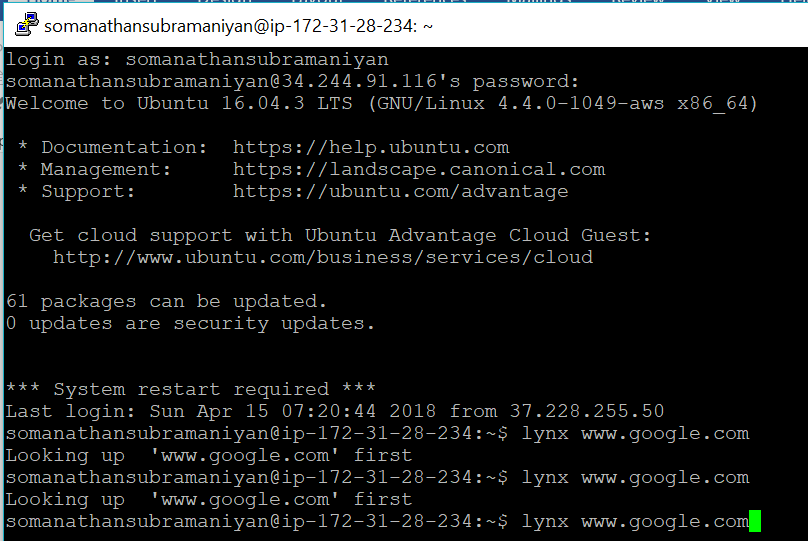


### Q2.3.2: Question

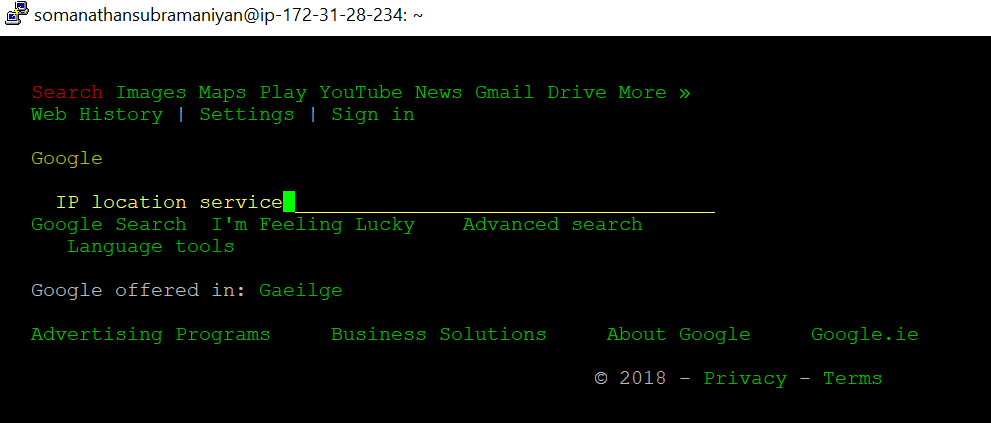
Because there is no GUI installed on the VM, all programs have to run instead in text mode. Use the ‘lynx’ text-based browser program on the VM. To run it, type:  lynx www.google.com In the lynx browser, search for an online IP location service to determine the city and country where the VM is located. Type the IP address of the VM into the IP locator website and it will tell you the VM’s location (Screenshot the result from this for submission).

### Q2.3.2: Answer

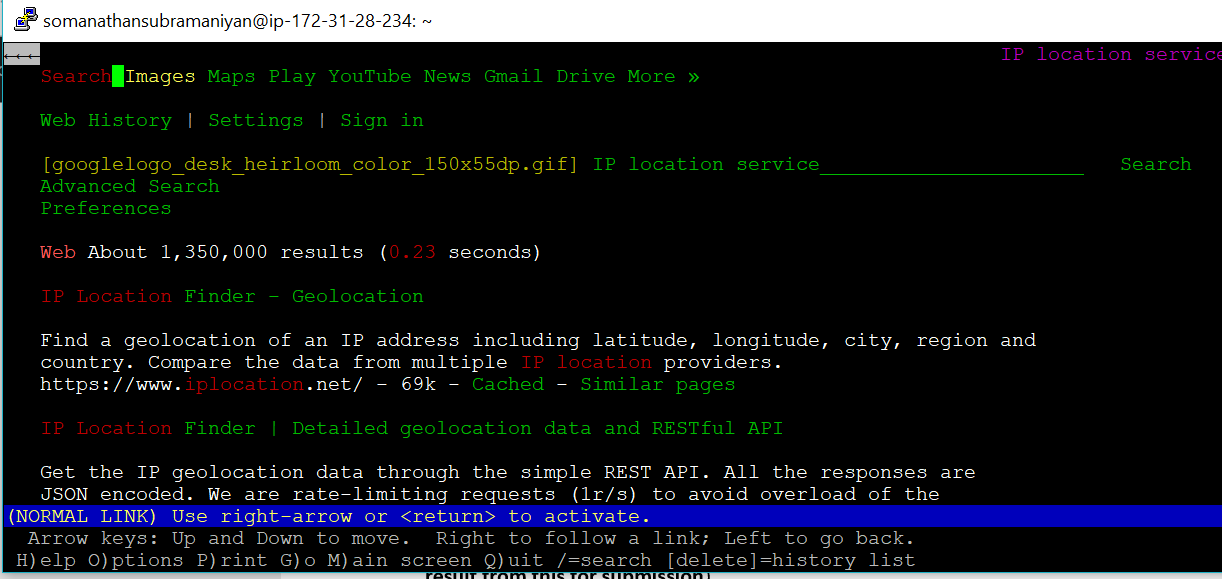
Type lynx www.google.com



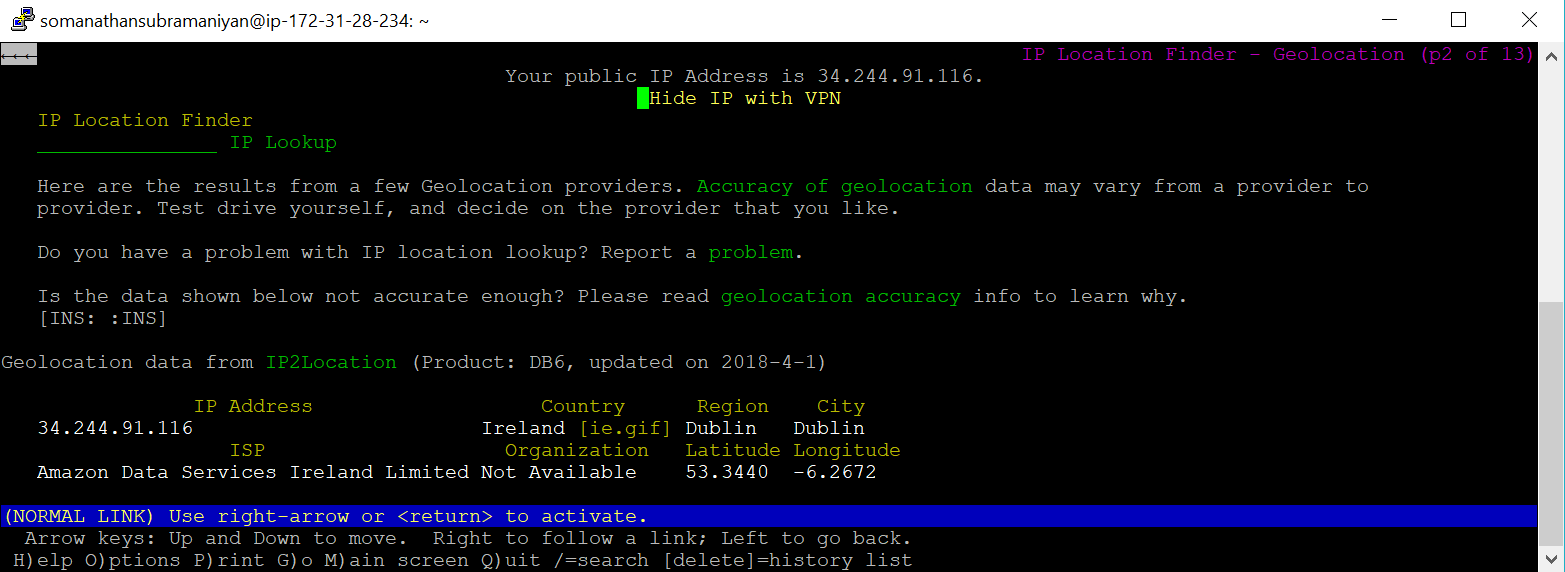
Search for “IP location service”



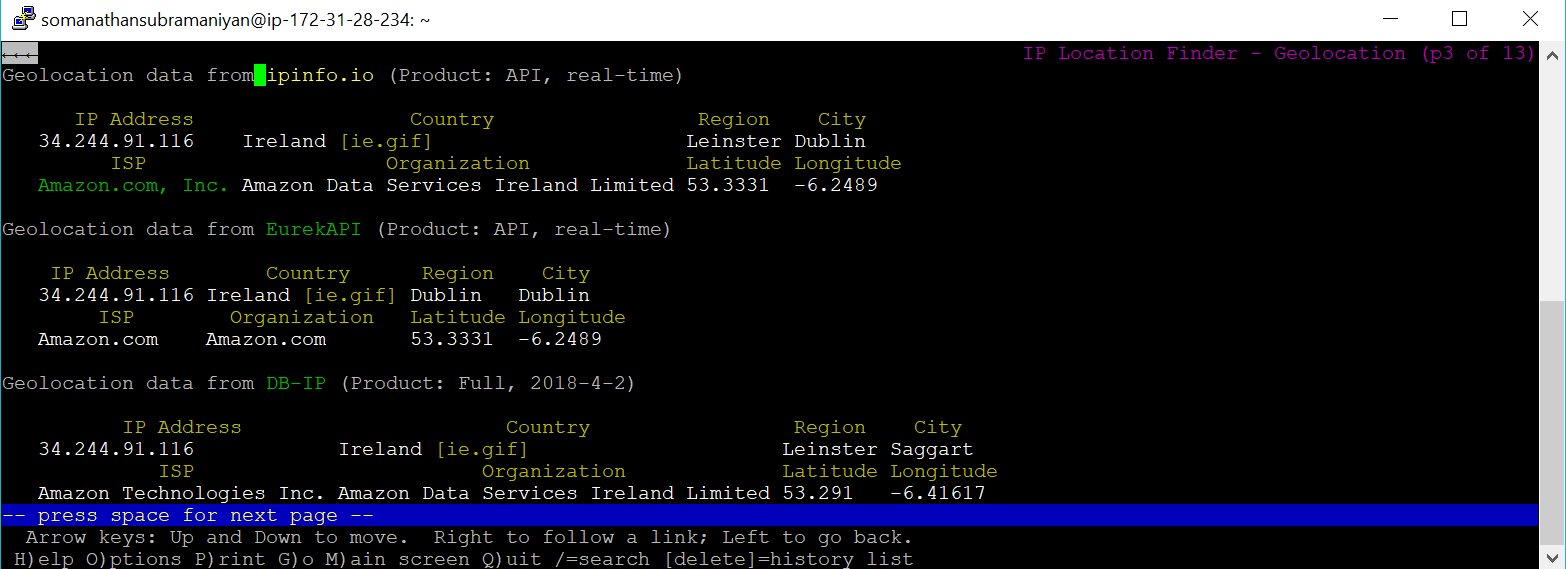
Google Search Results are show in the attached screenshot

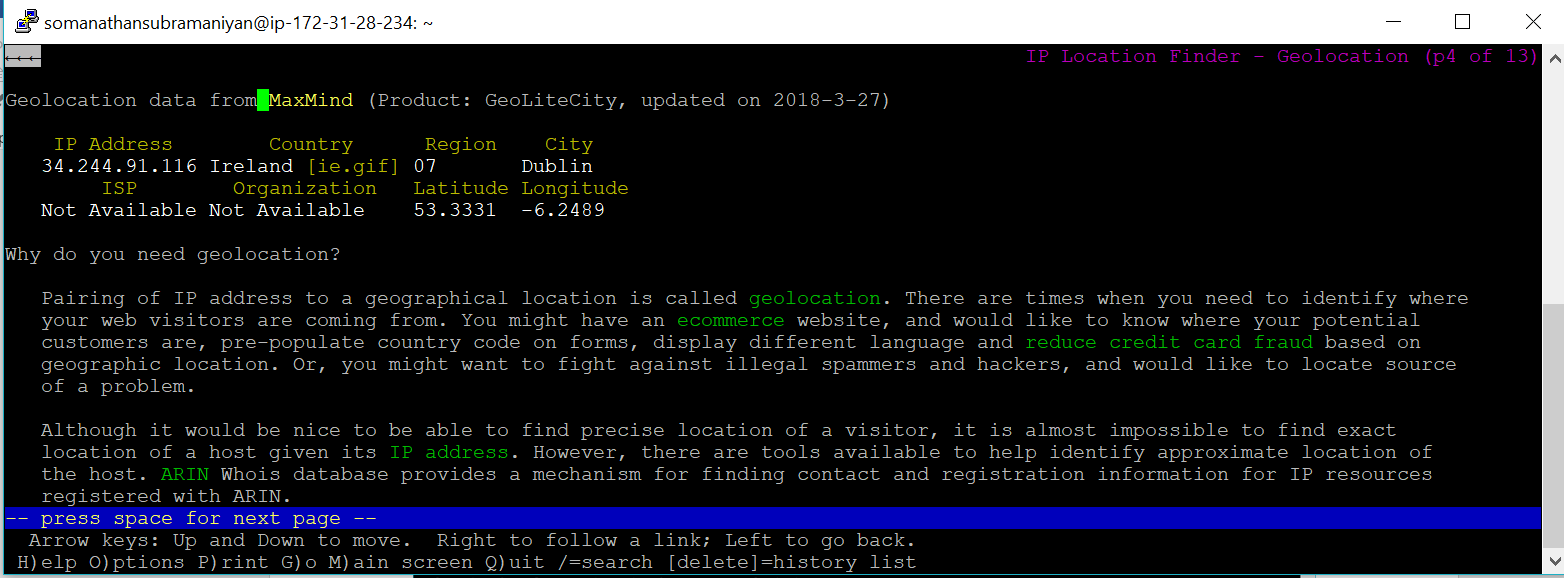


Select the first Search results provides the IP location of 34.244.91.116



The location is Amazon Data Services Ireland Limited, Leinster, Dublin





## Q2.4. Question

This is a research project. Use Google to help you identify a solution.

The objective of the task is to (using the Vim text editor) write a shell script program that behaves like an Irish person offering a cup of tea.

If the user types ‘y’ to the offer, the program displays “Great, I’ll make tea now” to the console.

If the user types ‘n’ to the offer, the program asks the user “Are you sure” 4 more times before giving up. If at any point during the 4 follow up offers, the user changes their mind and presses ‘y’, the computer will print out “Great, I’ll make tea now” to the console.

In addition to shell scripting, this assignment examines your ability to use ‘while loops’ and’ if statements’ correctly. It also examines your ability to research and locate the information required online.

For the submission: Capture a screenshot of the program in operation. Copy and paste the screenshot into your document for submission. Also, copy and paste the shell script code into your Word document for submission.

## Q2.4. Answer

### Shell Script

#!/bin/bash

counter=0

echo "Hello, do you like to have tea?"

while [ $counter -le 5 ]

do

read varname

if [ $varname == 'y' ]

then

echo "Great, I'll make tea now "

exit

elif [ $varname == 'n' ]

then

((counter++))

if [ $counter -eq 5 ]

then

exit

fi

echo "Are you sure?"

else

echo "Please type-in 'y' for yes and 'n' for no"

fi

done

### Screenshots

