Week 2: Assignment 4
Find Valid IP Addresses

[*Last Updated on:* ***11th April 2021, 23:00 Hrs***]

Aim

Your computer networks project deadline is knocking on the doors and you have a considerable amount of work left. A major portion of the project requires you to validate IP addresses and map the valid IP addresses to their respective classes. I am sure you can easily find out on your own what a valid IP address is.

You have a text file of a log that contains some addresses and you want to write a script that will automate this task.

Write a shell script that will extract valid IP addresses from a given text. For every valid IP address found, print its corresponding class.

You have a text file that contains some addresses and you want to write a script that will automate this task. Write a **Shell (.sh)** script that will extract valid IP addresses from a given text. For every valid IP address found, print its corresponding class.

If there are **n** valid IP addresses in the text, then your script's output should be **n** IP addresses followed by their class separated by a space.

IPv4 Address Format:

w.x.y.z where **w, x, y, z** are in range **[0, 255]**

- **Class A**: **w** is in range **[0, 127]**
- **Class B**: **w** is in range **[128, 191]**
- **Class C**: **w** is in range **[192, 223]**
- **Class D**: **w** is in range **[224, 239]**
- **Class E**: **w** is in range **[240, 247]**
- **Not Defined**: **w** is in range **[248, 255]**

For example, if the given text contains 6 valid IP addresses, then your output should look like:

```
10.12.144.36 A
178.152.36.45 B
192.168.12.1 C
232.56.23.1 D
245.23.123.234 E
250.2.3.45 Not Defined
```

NOTE: The IP addresses should be printed in the same order as their order of occurrence in the input file.

Given

One file is provided to solve this assigment.

- Skeleton program file: **`assignment4.sh`**
- Sample TXT file: **`assignment4_sample.txt`**

Procedure

- Open the skeleton program file, **`assignment4.sh`** .
- You will notice pre-written comments included in skeleton program for your assistance to solve the assignment.
- To run and debug your solution, type the below command in Terminal:

  ```
  bash assignment4.sh assignment4_sample.txt
  ```

  This command will run the shell script **`assignment4.sh`** with the input argument of provided TXT file.

- Refer the **Expected Output** section below and debug your code to get the correct output.
- **NOTE:** To solve this assignment, you are free to use any commands and their combinations learnt so far may it be **grep**, **sed**, **awk**, etc.

Expected Output

- The provided sample TXT file, **`assignment4_sample.txt`** consists of some random text with few valid and invalid IP addresses.
- For example, the contents of this TXT file are as shown below:

  ```
  An IP address serves two principal functions. It identifies the host, or more specif

  The header of each IP packet contains the IP address of the 456.258.2.3 sending host

  Two versions of the Internet Protocol 12.59.126.23 are in common use in the Internet

  The rapid exhaustion of IPv4 address space available 195.26.56.45 for assignment to

  IANAs primary IPv4 address pool was exhausted on 3 February 2011, when 250.0.0.0 the

  Today, these two versions of the Internet Protocol are in simultaneous 156.25.78.89
  ```

- You can test your solution with following values and their expected outputs:

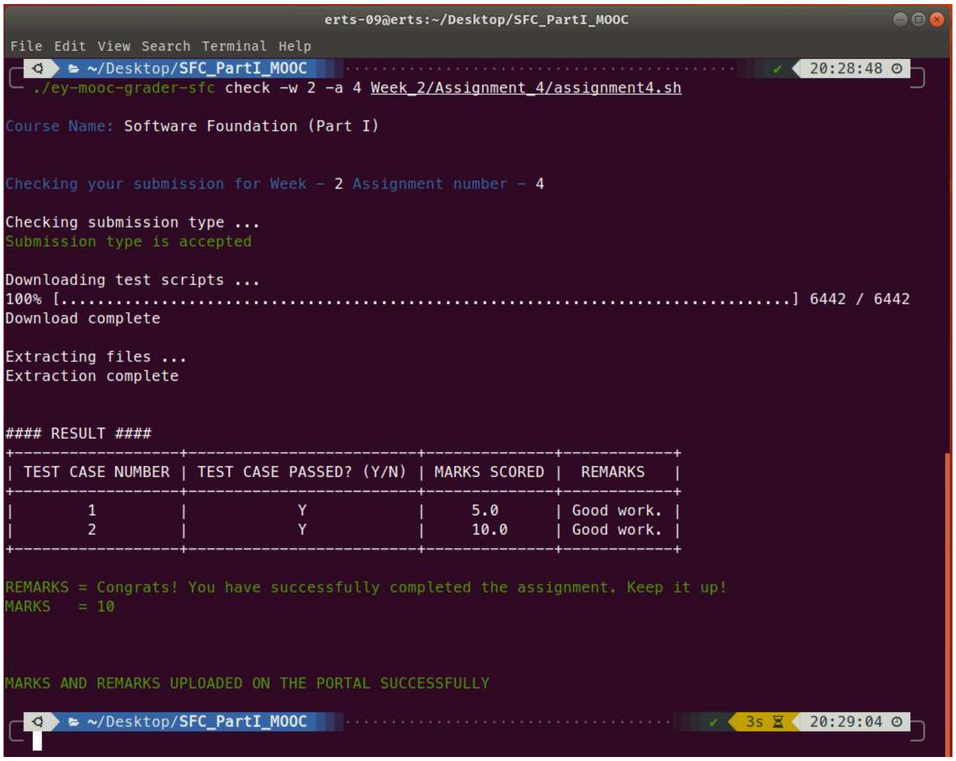
  ```
  249.56.2.3 Not Defined
  0.0.0.0 A
  255.255.255.255 Not Defined
  12.59.126.23 A
  123.45.78.1 A
  195.26.56.45 C
  250.0.0.0 Not Defined
  156.25.78.89 B
  ```

Grading and Submission Instructions

- Navigate to the folder where the ***ey-mooc-grader-sfc*** application resides.
- To grade your solution, run the `check` command of the application as follows:

```
./ey-mooc-grader-sfc check -w 2 -a 4 Week_2/Assignment_4/assignment4.sh
```

- This will run your program `assignment4.sh` against random test cases and grade it. Marks and appropriate remarks will be provided as shown in Figure 1.
- Your program file `assignment4.sh`, marks scored and remarks will get uploaded to the MOOC portal.

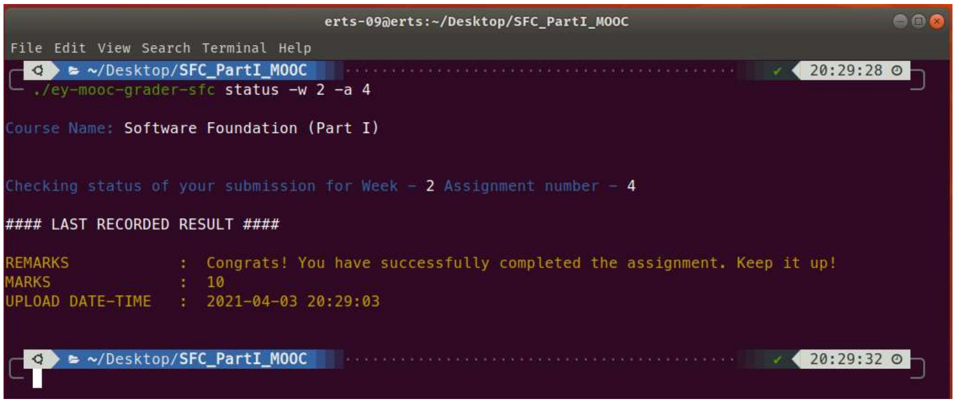


```
erts-09@erts:~/Desktop/SFC_PartI_MOOC
File  Edit  View  Search  Terminal  Help
 ~/Desktop/SFC_PartI_MOOC ........................................ ✓ 20:28:48
 ./ey-mooc-grader-sfc check -w 2 -a 4 Week_2/Assignment_4/assignment4.sh

Course Name: Software Foundation (Part I)

Checking your submission for Week - 2 Assignment number - 4

Checking submission type ...
Submission type is accepted

Downloading test scripts ...
100% [..............................................................................] 6442 / 6442
Download complete

Extracting files ...
Extraction complete


#### RESULT ####
+------------------+-------------------------+--------------+------------+
| TEST CASE NUMBER | TEST CASE PASSED? (Y/N) | MARKS SCORED |  REMARKS   |
+------------------+-------------------------+--------------+------------+
|        1         |            Y            |     5.0      | Good work. |
|        2         |            Y            |     10.0     | Good work. |
+------------------+-------------------------+--------------+------------+

REMARKS = Congrats! You have successfully completed the assignment. Keep it up!
MARKS   = 10


MARKS AND REMARKS UPLOADED ON THE PORTAL SUCCESSFULLY

 ~/Desktop/SFC_PartI_MOOC .................................. ✓ 3s ⧗ 20:29:04
```

Figure 1: Output of running check command for Week 2 Assignment 4

- You can verify this by running the `status` command of the application as given below, refer Figure 2.

```
./ey-mooc-grader-sfc status -w 2 -a 4
```



```
erts-09@erts:~/Desktop/SFC_PartI_MOOC
File  Edit  View  Search  Terminal  Help
 ~/Desktop/SFC_PartI_MOOC ........................................ ✓ 20:29:28
 ./ey-mooc-grader-sfc status -w 2 -a 4

Course Name: Software Foundation (Part I)


Checking status of your submission for Week - 2 Assignment number - 4

#### LAST RECORDED RESULT ####

REMARKS            :  Congrats! You have successfully completed the assignment. Keep it up!
MARKS              :  10
UPLOAD DATE-TIME   :  2021-04-03 20:29:03


 ~/Desktop/SFC_PartI_MOOC ........................................ ✓ 20:29:32
```

Figure 2: Output of running status command for Week 2 Assignment 4

References

- Nano Editor
  - How to use Nano Text Editor
  - Nano Editor Official Docs
- Vim Editor
  - Interactive Vim Tutorial
- Online RegEx Tools to test and validate your expressions
  - RegExr
  - REGEXPER Documentation
  - REGEXPER Example
- SED and AWK
  - SED: Introduction and Tutorial
  - AWK: Introduction and Tutorial
  - Very Useful Command Line Utilities
  - SED AWK Examples by Unix School
  - Use of Flow Control Statements with AWK
  - 20 AWK Examples

All The Best!