Assignment1: Bash Scripting
ISS

- **Deadline:** 11:55 pm, 18 January 2020
- This assignment will be evaluated automatically. ***Make sure you submit your scripts in the format specified, otherwise they will not be evaluated.***
  Format: Name each file as ‘q_number.sh’. For example, script for question 5 must be named as q_5.sh
- Put all the scripts in a folder, name it as ‘Rollnumber’ and compress it as ‘Rollnumber_Assignment1.tar.gz’.
  Sample inputs and outputs will be uploaded soon.
- **Instructions for Q4 and Q8:** Make a file *ans.txt* and put your answers to Q4 and Q8 in two consecutive lines. No need to submit q_4.sh ad q_8.sh

  $ cat *ans.txt*
  > Q4 Command
  > Q8 Command

Plagiarism will earn you a straight ZERO in this assignment. Automatic plagiarism detection scripts will be run on all your submissions.

Q1. Write a bash program which take two variables (i.e. warning level and critical level
percentage) and check disk space left. Finally print the output as shown below.
***Note* - Color the statements accordingly, for example, if the disk space is 'OK' then it must be shown in blue color, 'warning' in yellow and 'critical' in red.**

```
$ bash answer.sh 75 90
> OK, /dev/sda1, 40%
> WARNING, /dev/sda2, 78%
> CRITICAL, /dev/sda3, 99%

$ bash answer.sh 80 95
> OK, /dev/sda1, 40%
> OK, /dev/sda2, 78%
> CRITICAL, /dev/sda3, 99%
```

Explanation: All the disks with percentage between 75% and 90% are printed as WARNING, while greater than 90% CRITICAL

Q2. Write a bash program which print all the directories present in current folder twice. Once sorted by size of folder, and once by files count. Script should consider all the files recursively and not the files just immediately inside the folder.
Usage of only "find", "wc" and "ls" command is allowed.

```
$ bash answer.sh
> folder1 20G
> folder2 10G
> folder3 1G

> folder2 212 files
> folder1 100 files
> folder3 2 files
```

Q3. Write a bash script which takes a number N as argument and pings 'google.com' N times with ordered listing.
Your output must look like:

```
$ bash script.sh 5
PING google.com (172.217.163.46) 56(84) bytes of data.
1. 64 bytes from maa05s01-in-f14.1e100.net (172.217.163.46): icmp_seq=1 ttl=128
time=20.6 ms
2. 64 bytes from maa05s01-in-f14.1e100.net (172.217.163.46): icmp_seq=2 ttl=128
time=28.2 ms
3. 64 bytes from maa05s01-in-f14.1e100.net (172.217.163.46): icmp_seq=3 ttl=128
time=46.8 ms
4. 64 bytes from maa05s01-in-f14.1e100.net (172.217.163.46): icmp_seq=4 ttl=128
time=21.5 ms
5. 64 bytes from maa05s01-in-f14.1e100.net (172.217.163.46): icmp_seq=5 ttl=128
time=28.9 ms

--- google.com ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4007ms
rtt min/avg/max/mdev = 20.678/29.258/46.817/9.400 ms
```

Q4. Write one line command (using pipes) to use in following scenario:

```
$ ls
> to_delete.txt
> delete/

$ cat to_delete.txt
> a b f g

$ ls delete/
> a.txt b.txt c.txt d.txt e.txt f.txt g.txt h.txt i.txt … x.txt y.txt z.txt

$ Your command here

$ ls delete/
> c.txt d.txt e.txt h.txt i.txt … x.txt y.txt z.txt
```

Explanation: Delete all the files from delete folder, which are listed in to_delete.txt file. Hence in the above example we deleted files a.txt, b.txt, f.txt and g.txt.

Q5. Ask the user their SGPA (don't ask in real). If the SGPA is less than 5, then output 'PROBATION'. Also output the required SGPA to be scored in the next semester such that CGPA after the first year (i.e. after 2nd semester) is 8. If not possible, print 'AGLE SEM PHOD DENGE'.

Your output must look like:

$ bash script.sh
> Please enter your SGPA:
Input - 7.5
> You need SGPA 8.5 next sem

$ bash script.sh
> Please enter your SGPA:
Input - 4
> PROBATION
> AGLE SEM PHOD DENGE

$ bash script.sh
> Please enter your SGPA:
Input - 5.85
> AGLE SEM PHOD DENGE

Q6. Write functions to calculate GCD and LCM of two numbers. Now take two numbers A and B as input and output their GCD and LCM using the functions.

$ bash script.sh
> Enter A:
Input - 100
> Enter B:
Input - 150
> GCD: 50
> LCM: 300

Q7. Ask user for their name and birth date(month and year). Output their exact age in months.

$ bash script.sh
> Enter name:
Input - XYZ
> Enter DOB:
Input: 11-2000 ***(Note input format - only month and year)***
> Hello XYZ your age is 230 months.

Q8. Write a one line command to search a string (given as an argument) in a directory recursively. Output the documents in descending order of their names.

Q9. Write a bash script to convert a natural number (<=899) into a roman number.

$ bash script.sh 77
> LXXVII

Q10. Write a bash script that can sort a list of command line parameters in ascending order. There is no limit to the number of arguments passed as list parameters.
For example, your command will look something like:

$ bash script.sh 8 27 9 -2 7 92 -9 0
> -9 -2 0 7 8 9 27 92

Use only basic commands and array. Do not use any built-in commands that sort array.