# Scripting and Computer Environments - CSE 505
# IIIT Hyderabad - Monsoon 2017

Assignment 1

September 13, 2017

---

**ASSIGNMENT DEADLINE:** 28th Sept. 2017 - 11:55 PM
**SUBMISSION FORMAT:**

1. Create a folder with your roll number <rollno>_assignment1
2. Under the folder create <section_no>_<question_no>_<subpart>.sh for each question
3. Compress the file as <rollno>_assignment1.tar.gz and then upload it to your moodle account.

**How to get zero in the Assignment?**

1. DEVIATING FROM THE UPLOAD FORMAT will get you a straight zero.
2. ANY KIND OF PLAGIARISM will also lead to a straight zero.

---

## 1 grep/egrep

**Question 1.**
For the following questions, use the **/usr/share/dict/words** dictionary file.

1. Find all words that have the string India or Africa in them.
2. Find all words that begin with a vowel. How many are there?
3. Now, find the words that begin with a vowel and end with a vowel.
4. Find all words that have two or more consecutive a in them.
5. Find all words that have any punctuation marks (such as apostrophe) in them.
6. Find words that contain an uppercase letter.

(a) at the beginning.

(b) at the end.

(c) anywhere.

7. Find all words that are twenty characters or more in length.

8. Find all five-character or ten-character words that do not contain any uppercase letter.

9. Find all words that begin and end with the same letter.(not necessarily the same vowel).

**Question 2.** Using grep perform following

1. Print count of directories present in current folder.

2. Write Regular expression to match

   (a) xxx.xxx.xxx.xxx where x will be numerals

   (b) same as above but xxx should be less than 256

   (c) valid IP address

3. Write Regular expression to validate mac address for following formats

   (a) Six groups of two hexadecimal digits separated by hyphens (-), like 01-23-45-67-89-ab.

   (b) Six groups of two hexadecimal digits separated by colons (:), like 01:23:45:67:89:ab (a & b must be validated in a single regular expression )

   (c) Verify your regex by extracting mac address of your system using ifconfig cmd.

# 2 sed

**Question 1.** Redo the above grep question number (1) using sed command only.

**Question 2.** You are given the following address book details (in CSV format) in the order: given: **name, surname, address, city, DoB, e-mail**
Alice, George, 83 First St, Howard, 12/05/1980, awatson@example.com
Bob, Charlie, 40 West Ave, Anycity, 02/11/1995, bcharlie@example.com
Steve, Park, 1981 Songdo, Smartcity, 24/1/1971, spark@exmaple.com
Mary, Sam, 40 West Ave, Anycity, 30/01/1990, msam@example.com
Kumar, Anil, Gachibowli 500032, Hyd, 11/12/1989,kanil@example.com

Write sed commands for the following:

1. Who lives in a city named Anycity?

2. Remove all lines that begin with a vowel(name column).

3. How many people are born in the 80s?

4. Replace all numeric values by ? and all punctuation marks by *.

5. Reverse the order of the Given name and Surname. (e.g. "Mary, Sam" "Sam, Mary").

6. Swap the first and last characters of a line. (e.g. "Alice ... example.com" "mlice...example.coA").

**Question 3.** Freshers are taught how to write C++ programs. Now they are very happy and have started writing programs. Seniors checked their program but couldnt understand anything since the freshers had put lot of comments in their code.

```
// My very first C++ code :)
#include <iostream> //iostream provides basic i/o services for C++ programs.

using namespace std;

//This function calculates sum and returns sum.
int func(int a,int b)
{
    int c = a+b; //Adding two variables.
    return c;  //Return sum to main function.
}

//Main function
int main()
{
    int x,y; //To declare input values
    cout<<"Enter two values :"<<endl;
    cin>>x>>y; //To take input
    int sum = func(x,y); //To calculate sum, call func
    cout<<"Sum: "<<sum<<endl;
    return 0;
}

//I am happy :)
```

Now, your task is to help Seniors to remove all comments and empty lines.
Can you write a **sed command** to achieve this?
**Note:** A comment will always start with //

# 3 awk

**Question 1.** You are given the marks of students taking a course named ABC.

| Name | Gender | Mid1 (25%) | Mid2 (25%) | Endsem (50%) |
|---|---|---|---|---|
| AA | M | 20 | 15 | 35 |
| BB | F | 22 | 17 | 44 |
| CC | F | 19 | 14 | 25 |
| DD | M | 15 | 20 | 42 |
| EE | F | 18 | 22 | 30 |
| FF | M | 0 | 20 | 45 |

Now, write an awk command to achieve the following:

1. Display just the names, genders and Endsem marks of:
   (a) all students.
   (b) the first 3 students only.
   (c) the last student.
2. Store the male students in males.txt and the females in females.txt.
3. Compute the total mark for each student and display it along with heir names.
4. Who is the topper? Which students scored above the class average?
5. Generate a grade report for each student based on the following marking scheme.
   A = [95-100], A- = [90-95), B = [85-90), B- = [80-85)
   C = [75-80), C- = [70-75), D = [60-70), F <60.

Your report must contain a header and a footer/end sections as well. The header be entitled, *****Grade Report for the ABC course*****.
At the end of the report, display the total number of students, the highest, lowest & average marks as well as the message,
***** End of Grade Report*****

**Question 2.** You are given a file named imdb-top-250.txt. This file is space separated. Your task is to convert the above file into JSON format as below:

```
[
        {
                "ID":"1",
                "Name":"The Shawshank Redemption",
                "Year":"1994",
                "Rating":"9.2"
        },
        {
                "ID":"2",
                "Name":"The Godfather",
                "Year":"1972",
                "Rating":"9.2"
        },
```

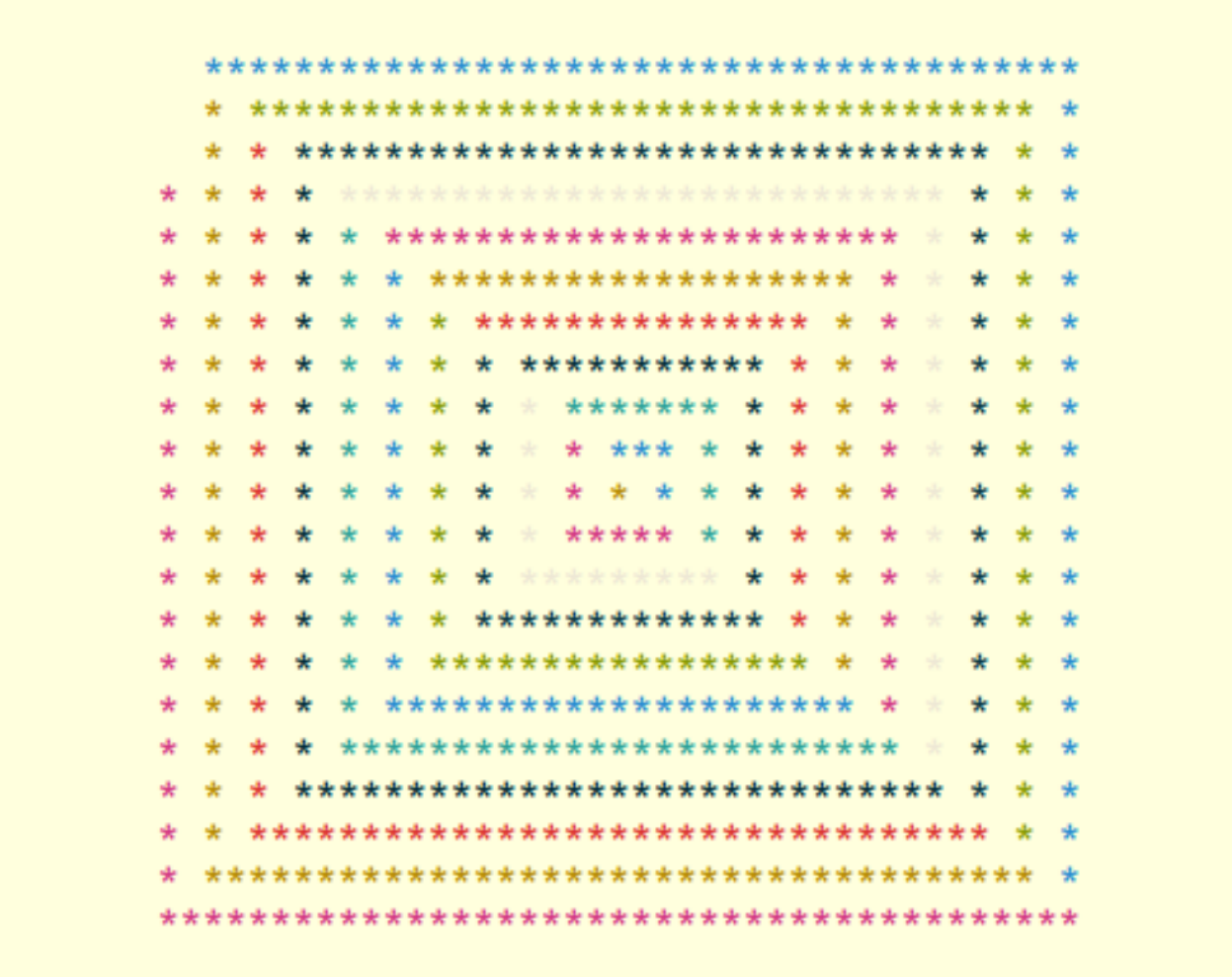
```
        {
                "ID":"3",
                "Name":"The Godfather: Part II",
                "Year":"1974",
                "Rating":"9.2"
        },
           .
           .
           .
        {
                "ID":"250",
                "Name":"PK",
                "Year":"2014",
                "Rating":"8.0"
        }
]
```

You are expected to redirect the output to a file named **imdb-top-250.json**. Finally test if it's a valid json by validating it on https://jsonformatter.curiousconcept.com/.

# 4 BASH

**Question 1.** Take an array of integer's as input. Sort the array using any of the algorithm but you cannot use inbuilt function like sort.You are expected to use algorithm like bubble sort, insertion sort etc.

**Question 2.** Jayant Tanwani wants to design a new screensaver using bash. He builds a spiral starting from middle of the screen growing upto limit of your terminal and back again to the centre. Limit of terminal is defined by min(rows,columns), where rows is rows of terminal and columns is columns in terminal. For demo, you can view spiral.gif in resources.
**Hint** : use **tput** command
**Catch:** Your script should work for all terminal sizes. even if we minimise or maximize terminal size in runtime, screen saver must keep running with new configurations.



**Question 3.** After releasing the SCE LAB-2, Rambabu(TA) went to his native place for spending the Weekend vacation with his family. Rambabu got to know that every now and then there are queries popping up on the moodle discussion thread. For answering the queries from outside the campus, Rambabu has to configure the VPN, start the VPN and then login to the moodle for answering the queries. Now Rambabu realised that whenever he visits home, hell have to repeat the same process of configuring the VPN and start it in order to access the moodle. So, he thought he could get some help from you in writing a bash script(start_vpn.sh) that could download the certificates, extract them to the relevant directories and then start the VPN. There should be another bash script file(stop_vpn.sh) for stopping the VPN. Help Rambabu by providing the bash script that could serve his need.
**Requirements:**
The bash script, start_vpn.sh should download the certificates from the https://vpn.iiit.ac.in/ and place the certificates in the appropriate directories and then start the VPN. Once the VPN is started, the script should print the message VPN Started on to the terminal. Later when stop_vpn.sh is executed the VPN should be stopped.

**Question 4.** Write a script that recursively lists all files i.e is equivalent to 'ls -R'.
**Requirements:**

1. You must use the recursive function to traverse

2. Can take any number of commandline arguments. For each of them, call this script

3. If no command line argument is provided, work on current directory (just like ls!).

Usage/Presentation Requirements:
Name the script 3.sh * Usage: bash 3.sh ¡any number of files/directories>
Usage/Presentation Requirements:
Support any number of command line arguments, including zero.
Usage/Presentation Requirements:
Validate incoming inputs. Do not stop on Errors.
Errors to handle and message Format:

1. 'Error: Invalid File/Directory!' - If a non-existent file/directory is passed.

2. Warning: No permission - If a directory has no permission, skip this and continue. Outputs to handle and message Format :

1. Output will be a directory along with the files in it. If you encounter a directory in here, call your script to traverse this directory.

Here is the format to be followed:
<PathName of Directory>:
<file1>
<file2>

**Question 5.** Write a recursive copy shell script to copy source folder to destination folder without using cp -r.
(Hint: Use export and readlink to store absolute path of the script). Input is given in format 4.sh <source><destination>
where source is a folder name and destination is a folder name.
**Requirements:**

1. You cannot use recursive functions or cp -r command.

2. If destination folder does not exist, create one.

3. Call the same script to traverse down the directory.

4. Use cp command to copy single file, mkdir command to create directory with same name.

5. Directory names with spaces must be considered.

**Errors to handle and message Format:**

1. 'Error: Invalid number of arguments!' - If number of arguments is != 2
2. 'Error: Invalid input!' If source is not directory.
3. Error: Access denied! - if source directory does not have access permission.
4. Warning: Dir <dir>has no permission. Skipping - If any of the subdirectory doesnt have execute permission. (special case to handle: Think of how to copy linked files.)