CSE 4034 – ADVANCED UNIX PROGRAMMING
Project # 1
(Due: 18.04.2021 23:59)

Q1. In this question, you will implement a shell script code (named as *Q1.sh*) to give statistics about each word of a given file. You should find each word in a line and report statistics about in the given format: *x1.x2.x3.x4,* where x1 is the total number of uppercase letters in the word; x2 is the total number lowercase letters in the word; x3 is the total number of digits in the word; x4 is the total number of other characters in the word.

As an example, the statistics regarding the word CSE4034 should be 3.0.4.0.

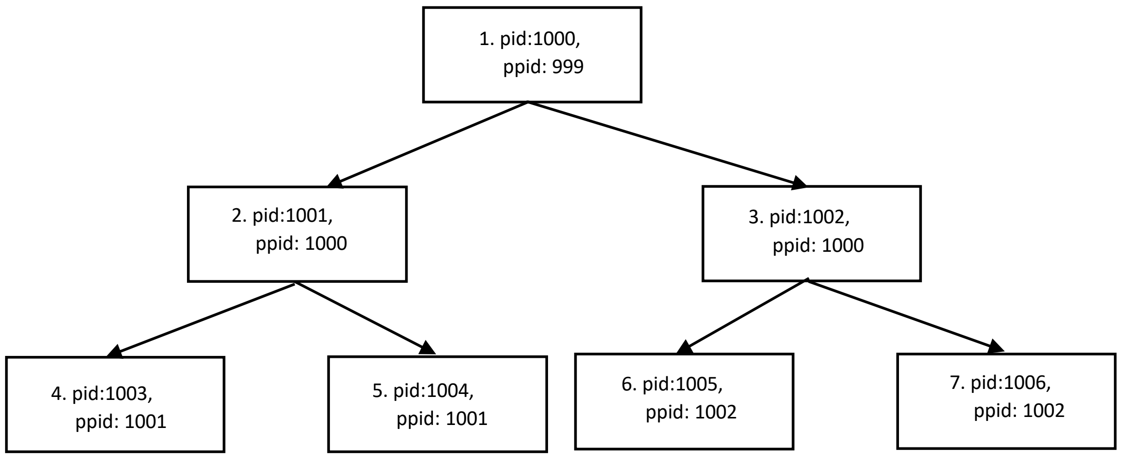
If the user does not provide a filename, then your program should take the input from the standard input (keyboard) until the user enters Ctrl-D. Then it should report the statistics of each word at each line. You can assume that each word is separated by a space (either single or multiple) in the given input file.

As an example, if the file *mydata* contains just one line:

CSE4034:   Advanced Unix Programming

then " ./*Q1.sh mydata*" should return "3.0.4.1 1.7.0.0 1.3.0.0 1.10.0.0"

Q2. In this question, you will implement a C program to construct a process binary tree. Your program should take a single command line argument, which is the depth of the tree, and each process should take an integer value that shows its traversal position at each level. If the user enters the value of 3 as the depth value, then you should construct a binary tree of processes with the following hierarchy:



You do not need to print the output as in the figure; it is enough to print the tree structure as follows:

```
./Q2.out 3
 [1] pid 1000, ppid 999
 [1] pid 1000 created left child with pid 1001
 [1] pid 1000 created right child with pid 1002
  [2] pid 1001, ppid 1000
  [3] pid 1002, ppid 1000
  [2] pid 1001 created left child with pid 1003
  [3] pid 1002 created left child with pid 1005
  [3] pid 1002 created right child with pid 1006
   [4] pid 1003, ppid 1001
  [2] pid 1001 created right child with pid 1004
   [6] pid 1005, ppid 1002
   [7] pid 1006, ppid 1002
   [5] pid 1004, ppid 1001
  [3] right child 1006 of 1002 exited with status 7
  [3] left child 1005 of 1002 exited with status 6
  [2] right child 1004 of 1001 exited with status 5
  [2] left child 1003 of 1001 exited with status 4
 [1] right child 1002 of 1000 exited with status 3
 [1] left child 1001 of 1000 exited with status 2
```

Note that each line of output is indented according to the depth of the node in the binary tree and begins by printing the traversal position of the process that prints it.

The steps of the program should be as follows:

- Each process in the program should print its own position with its pid and parent pid values.
- When a process creates a new process, the parent process should print its position, its own pid and the pid of the newly created child process (as left child or right child).
- When a child exits from the system, it should return its traversal id to the parent. This value should be obtained by the parent. Then, the parent should print the child (left or right child), the child pid and the exit status of the child process.

Since the output order might change, you can put sleep(3) calls for each process.

Important Notes:

- You can work in groups of 2 people.
- Please write a minimum 2-pages long project report that describes how you implemented the project and add several screenshots that show sample runs.
- We will use tools that automatically detect plagiarism among the submissions!
- In case of any form of copying and cheating on solutions, you will get **FF** grade from the course! You should submit your own work. In case of any forms of cheating or copying, both giver and receiver are equally culpable and suffer equal penalties.
- Please zip and submit your files using filename Student1Number_Student2Number_Project1.zip (ex: 150713852_150713098_Project1.zip) to Canvas system (under Assignments tab).
- No late submission will be accepted.