TU Dublin, Tallaght
Department of Computing
IT Scripting and Automation
Lab 6

Instructions:

Create a MS Word document with the solutions of each exercise. Include your name and x-number.

Exercise 1.

Create a text file (called employee) with the following information:

```
student@ITSA-Server:~/awk$ cat employee
1 John Manager Sales $ 40000
2 Mark Manager Technology $ 50000
3 Joe Developer Technology $ 30000
4 Sean Officer HR $ 35000
5 David Officer Sales $ 36000
```

1.1 Try and then explain the outputs of the following AWK statements.

a) **$ awk 'BEGIN { FS = " " ;} {print $2;} ' employee**

b) **$ awk '{sum=sum+$6;} END{print "Total salary: " sum;}' employee**

c) **$ awk '{sum=sum+$6; print "Total salary: " sum;}' employee**

d) **$ awk '{if($1 > 40000){ print $2" – "$6; }}' employee**

e) **$ awk '$6>40000' employee**

f) **$ awk 'BEGIN {print "Name\tRole\tDepartment\tSalary";}**
{print $2,"\t",$3,"\t",$4,"\t",$NF;}
END{print "End of Report \n----------";}' employee

1.2 Using the previous text file (employee) create an AWK statement to calculate:

a) The average salary in this company
b) The average salary for "Technology" department only
c) The total salary paid in "Sales" department only
d) Number of "Officer" employees

Exercise 2.

Create the following script (called script2.sh) to process "`/etc/passwd`" file. Show execution of your script (screenshot). Explain the output of the script.

```
#!/bin/bash
directory=$1
file=$2
path=$1/$2
cat $path | awk 'BEGIN{FS=":";}{print $1;}' | sort
```

Script execution:

./script2.sh /etc passwd

Exercise 3.

Create AWK statements to do the following:

- Print the total number of kilobytes used by files with extension 'sh'.
- Count the number of lines in a file.

Exercise 4.

Create a Python script with the following functions:

- A function to show the content of "`/etc/passwd`" file (use **cat** command).
- A function to list the contents of "`/tmp`" directory. (Use **ls** command).
- A function to list all the processes running on your system (use **ps** command).
- A function to display the disk space (in MB) used in the current working directory (use **du** command).

Note: You may need to check the arguments that are needed in each command, for this use **man** command.