

Department of Computer Technology & Information Systems

CTIS 166: Information Technologies

Spring 2014 – 2015 Semester

Lab Final: Sections 1, 2, 3- SOLUTION

Date: 11/05/2015

Surname, Name:

Section:

Student No:

NOTICE TO THE STUDENTS

Read the instructions carefully listed below and sign the box:

1. Textbooks, lecture notes, calculators with extensive memories, and any kind of computers are not permitted in the classrooms: if you have any, leave them on the instructor's desk.
2. Cell phones should be **totally switched off** (not in silent or flight modes) and do **not keep them with you**: either put them in your bags or leave them on the instructor's desk.
3. Permitted material to be kept on your desks are: pencils, sharpeners, erasers (and in case you may need: water and tissues). **Pencil boxes are strictly forbidden.**
4. Check your desk for any **graffiti**; the graffiti related to the course will be treated as an attempt to cheat.

You are not allowed to **talk** to other students during the exam.

Signature:

5. Disobeying the above rules will be severely penalized and a **disciplinary action** will be conducted.
6. Please **place your ID (with photo)** on your desk for identity check.

INSTRUCTIONS BEFORE STARTING THE EXAM

Step 1: `mkdir yourname-surname` (DO NOT use Turkish characters. For example, `mkdir Murat-Yildirim`)

Step 2: `cd yourname-surname` (For example, `cd Murat-Yildirim`)

INSTRUCTIONS FOR UPLOADING YOUR WORK

Step 1: `cd ..` (Assuming that your present directory is yourname-surname)

Step 2: `zip -r yourname-surname yourname-surname`

(For example, `zip -r Murat-Yildirim Murat-Yildirim`)

Step 3: See the contents of your zip file: `unzip -t yourname-surname`

Step 4: Upload the files and your **history file** (`history > History.sh`) `yourname-surname.zip`
(For example, `Murat-Yildirim.zip`)

1) Given `a.txt`, we want to determine the lines containing string `Net` and not containing `Fox` and put them in files `Q1-1.txt`, `Q1-2.txt` and `Q1-3.txt` How would you do it using: (Do not forget to put your commands at the end of the files)

- `grep` (5 pts): `grep Net a.txt | grep -v Fox > Q1-1.txt`
- `sed` (5 pts): `sed -n '/Net/p' a.txt | sed '/Fox/d' > Q1-2.txt`
- `awk` (5 pts): `awk '/Net/' a.txt | awk '!/Fox/' > Q1-3.txt`

2) (15 pts) Write an Awk script `Q2.awk` which will compute grade point average of the course by using `grades.txt`. Also prints named of students with grade F.

```
BEGIN{print "Students with F grade:\n"}
```

```
/A/ {countA=countA+1}
```

```
/B/ {countB=countB+1}
```

```
/C/ {countC=countC+1}
```

```
/D/ {countD=countD+1}
```

```
/F/ {countF=countF+1; print $1, "got F\n"}
```

```
END{print "Total Numbers:\n\t A=", countA "\n\t B=", countB "\n\t C=", countC, "\n\t D=", countD}
```

Example Run:

```
awk -f Q2.awk grade.txt
```

```
Students with F grade:
```

```
fatma Got F
```

```
sibel Got F
```

```
Total Numbers: A= 3 B= 2 C= 1 D= 1 F= 2
```

The above solution is acceptable. But grade point average can be computed as:

```
sum=4 * countA + 3* countB + 2 * countC + countD, ave=sum/NR
```

or a direct Awk Script would be:

```
/A/ {sum=sum+4 }
/B/ {sum=sum+3 }
/C/ {sum=sum+2 }
/D/ {sum=sum+1 }
/F/ {print $1 " got F " }
END{ print "average=", sum/NR }
```

- 3) 3 Recursively search for files inside */usr/include* directory which contain pattern "math.h" and write the file names to *q3.txt*. (output redirection, do not forget to put the command at the end of file)
`grep -rl math.h /usr/include/ >q3.txt`

- 4) (20 pts) Write down a shell script *q4.sh* that takes one option either M, A or S and two integers a and b (as command arguments) in order to carry out the related operation:

Hint: To compare strings in the if statement, you can use the operator =.

M: Multiplying of two integers a and b: $a*b$

A: Adding of two integers a and b: $a+b$

S: Subtracting integer b from integer a: $a-b$

For example, you can run this script as:

`./q4.sh 14 2 M` Here the corresponding output is The product of 14 and 2 is 28

`./q4.sh 6 5 A` Here the corresponding output is The sum of 6 and 5 is 11

`./q4.sh 19 3 S` Here the corresponding output is Subtraction 3 from 19 is 16

```
#!/bin/bash
case $3 in
    "M")
        echo "The product of $1 and $2 is $((($1*$2)))"
        ;;
    "A")
        echo "The sum of $1 and $2 is $((($1+$2)))"
        ;;
    "S")
        echo "Subtraction $2 from $1 is $((($1-$2)))"
        ;;
    *) echo " You should give 2 integers and one of A, M, S
        ;;
esac
```

- 5) (15 pts) Write down a shell script that takes an arbitrary number of arguments. Using for x construct compute sum of even numbered terms and compute number of such terms, named *q5.sh*. (If there are 9 terms, you should sum terms 2, 4, 6, 8. You are not allowed to use number of terms explicitly.)

Example Run:

```
./q5.sh 1 2 3 4 5 6 7 8 9
Sum of even terms is 20
```

```
#!/bin/bash
term=$#
sayi=0
sum=0
for ((i=1;i<=$term;i++))
do
test=$(( i % 2 ))
```

```

if [ $test -eq 0 ]
then
let sum=sum+$i
((sayi++))
fi
done
echo "sum of even $sayi terms is $sum "

```

6) (15 pts) Write the bash shell script **basic.sh** that reads any three integers as inputs (not as command arguments) to store them in three different variables, prints their sum if this sum is greater than 100 and their average if their sum is greater than 50 and otherwise "Their sum is less than 50" on the screen.

For example, you can run this script as

```

./basic.sh
Enter your three integers
30
50
60
Sum of (30,50,60) = 140

```

```

./basic.sh
Enter your three integers
10
20
30
Average of (10,20,30) = 20

```

```

./basic.sh
Enter your three integers
10
20
5
Their sum is less than 50

```

```

#!/bin/bash
echo "Enter your three integers each on a separate line"
read num1
read num2
read num3
sum=`expr $num1 + $num2 + $num3`
if [ $sum -gt 100 ]
then
    echo "Sum of ($num1,$num2,$num3) = $sum"
elif [ $sum -gt 50 ]
then
    echo "Average of ($num1,$num2,$num3) = `expr $sum / 3`"
else
    echo "Their sum is less than 50"
fi

```

7) (5 pts) Write a cron entry to run **clean.sh** at every 4 hours, 15 minutes past, between June to August, every Sunday and copy your cron content to **q7.txt**.

```

crontab -e
15 */4 * 6-8 0 clean.sh

```

8) (21 pts) Write each command to **q8.txt** in the same order that you used

- Compile `sleep.c` and create an object file called `run1`
`gcc sleep.c -o run1`
- Run `run1` on the background
`./run1 &`
- Copy `run1` as `run2`, run on foreground and then suspend.
`cp run1 run2`
`./run2`
`^Z`
- How would you put the process with job id 2 running in the foreground and then kill/terminate this process abnormally?
`fg 2`
`^C`
- How would you put the process with job id 1 in the foreground?
`fg 1`
- How would you suspend the process with job id 1 running in the foreground?
`^Z`
- How would you put the process with job id 1 in the background?
`bg 1`

9) (5 pts) Find the files recursively under `/usr/include` directory whose name contains 7 characters, second character is `p` or `r` and ends with `.h` and copy these files to `LabE/folder2`.

```
find /usr/include -name "[p-r]????.h" -exec cp {} LabE/folder2/ \;
```

10) (5 pts) Create `myfiles.tar.bz2` which contains `LabE` directory using a single command.
`tar -jcvf myfiles.tar.bz2 LabE/`