COMP2700 Lab 2 -- solutions for selected exercises

Note that in the following, lines staring with a `$` symbol indicates the command that follows is typed into the terminal (so `$` at the beginning of the line is not part of the command).

Exercise 1

```
$ grep -R -i password /etc/*
```

Note that you may encounter some permission denied messages, as user Alice does not have privileges to access some directories. The option -i tells grep to ignore case (so the search is case insensitive). The option -R tells grep to search recursively, so it includes all the files in all the subdirectories of /etc.

Exercise 2

```
$ find /home/alice -name ".*"
```

Exercise 3

- Step 1. Find the location of cat:

  ```
  find / -name cat
  ```

  The location you are looking for is /bin/cat.

  (You could also use the `which` command which will give you straightaway the location of cat. But find is more general and can find files which do not correspond to bash commands).

  > A more advanced version (not covered in the lab - so you can ignore this for now):
  >
  > ```
  > find / -name cat 2> /dev/null
  > ```
  >
  > The '2> /dev/null' suppresses the error messages.

- Step 2. Copy it to Alice's home directory:

  ```
  $ cp /bin/cat ~/mycat
  ```

- Step 3. Run mycat:

```
~/mycat ~/lab1/ab.txt
```

Exercise 4

```
$ cracklib-check < ~/lab1/passwords.txt | grep OK
```

Exercise 5.

```
$ mv ~/lab1/hello ~/lab1/ls
$ export PATH=/home/alice/lab1:$PATH
```

Exercise 6.

```
#!/bin/bash

echo "Input marks: "
read x

if [ $x -ge 80 ]
then
   echo HD
elif [ $x -ge 70 ]
then
   echo D
elif [ $x -ge 60 ]
then
   echo CR
elif [ $x -ge 50 ]
then
   echo P
else
   echo N
fi
```

Exercise 7.

```
#!/bin/bash

x=0
y=1

echo "Input a number: "
read n
```

```
if [ $n -ge 1 ]
then
   echo $x
fi

for ((i=2; i<= $n; ++i))
do
  echo $y
  ((z = x))
  ((x = y))
  ((y = y + z))
done
```

Exercise 8 (*).

```
#!/bin/bash
#usage: ./sort.sh [Input File]

Filename=$1

if [ -z $1 ]
then
   echo "no input found"
else
   num=( $(<$Filename) )
   numlen=${#num[*]}
   temp=0
   for ((i=0;i<$numlen;i++))
   do
     j=$(($i+1))
     while [ $j -lt $numlen ]
     do
       if [ ${num[i]} -gt ${num[j]} ]
       then
          temp=${num[i]}
          num[i]=${num[j]}
          num[j]=$temp
       fi
       ((j++))
      done
    done
    echo ${num[*]} > sort.txt
fi
```