

Practical exercises : Conditions

```
$ cat on
#
# determine if someone is logged on
#
# see if the correct number of arguments were supplied
#
if [ "$#" -ne 1 ]
then
    echo "Incorrect number of arguments"
    echo "Usage: on user"
else
    user="$1"
    if who | grep "^$user " > /dev/null
    then
        echo "$user is logged on"
    else
        echo "$user is not logged on"
    fi
fi
$ on roshan
```

```
$ cat rem2
#
# Remove someone from the phone book -- version 3
#if [ "$#" -ne 1 ]
then
    echo "Incorrect number of arguments."
    echo "Usage: rem name"
    exit 1
fi
name=$1
#
# Find number of matching entries
#
matches=$(grep "$name" phonebook | wc -l)
#
# If more than one match, issue message, else remove it
#
if [ "$matches" -gt 1 ]
then
    echo "More than one match; please qualify further"
elif [ "$matches" -eq 1 ]
then
    grep -v "$name" phonebook > /tmp/phonebook
    mv /tmp/phonebook phonebook
else
    echo "I couldn't find $name in the phone book"
```

```
fi
$ rem Susan Tahoma

$ cat greetings
#
# Program to print a greeting -- version 2
#
hour=$(date | cut -c12-13)
if [ "$hour" -ge 0 -a "$hour" -le 11 ]
then
    echo "Good morning"
elif [ "$hour" -ge 12 -a "$hour" -le 17 ]
then
    echo "Good afternoon"
else
    echo "Good evening"
fi
$ greetings
```

```
$ cat number
#
# Translate a digit to English#
if [ "$#" -ne 1 ]
then
    echo "Usage: number digit"
    exit 1
fi
case "$1"
in
    0) echo zero;;
    1) echo one;;
    2) echo two;;
    3) echo three;;
    4) echo four;;
    5) echo five;;
    6) echo six;;
    7) echo seven;;
    8) echo eight;;
    9) echo nine;;
    *) echo "Bad argument; please specify a single digit";;
esac
$ number 9
$ number 99
```

```
$ cat ctype
#
# Classify character given as argument
#
```

```
if [ $# -ne 1 ]
then
    echo Usage: ctype char
    exit 1
fi
#
# Ensure that only one character was typed
#
char="$1"
numchars=$(echo "$char" | wc -c)
if [ "$numchars" -ne 1 ]
then
    echo Please type a single character
    exit 1
fi
#
# Now classify it
#
case "$char"
in
    [0-9] ) echo digit;;
    [a-z] ) echo lowercase letter;;
    [A-Z] ) echo uppercase letter;;
    ?      ) echo special character;;
    *      ) echo Please type a single character;;
esac
$ ctype a
$ ctype abc
$ ctype 7
```

```
$ cat greetings2
#
# Program to print a greeting -- case version
#
hour=$(date +%H)
case "$hour"
in
    0? | 1[01] ) echo "Good morning";;
    1[2-7]     ) echo "Good afternoon";;
    *          ) echo "Good evening";;
esac
$ greetings2
```

```
$ cat msgcheck.sh
#!/bin/bash
echo "This scripts checks the existence of the messages file."
echo "Checking..."
if [ -f /var/log/messages ]
```

```
then
    echo "/var/log/messages exists."
fi
echo
echo "...done."
$ msgcheck.sh
```

```
$ cat weight.sh
#!/bin/bash
# This script prints a message about your weight if you give it your
# weight in kilos and height in centimeters.
if [ ! $# == 2 ]; then
    echo "Usage: $0 weight_in_kilos length_in_centimeters"
    exit
fi
weight="$1"
height="$2"
idealweight=$((height - 110))
if [ $weight -le $idealweight ]; then
    echo "You should eat a bit more fat."
else
    echo "You should eat a bit more fruit."
fi
$ weight.sh 70 150
```

```
$ fileinfo.sh
#!/bin/bash
# This script gives information about a file.
FILENAME="$1"
echo "Properties for $FILENAME:"
if [ -f $FILENAME ]; then
    echo "Size is $(ls -lh $FILENAME | awk '{ print $5 }')"
    echo "Type is $(file $FILENAME | cut -d':' -f2 -)"
    echo "Inode number is $(ls -li $FILENAME | cut -d' ' -f1 -)"
    echo "$(df -h $FILENAME | grep -v Mounted | awk '{ print "On",$1, \"\
    which is mounted as the",$6,\"partition.\"}')"
else
    echo "File does not exist."
fi
$ fileinfo /etc/passwd
```

```
$ cat testleap.sh
#!/bin/bash
# This script will test if we're in a leap year or not.
year=`date +%Y`
if [ $($year % 400) -eq "0" ]; then
    echo "This is a leap year. February has 29 days."
elif [ $($year % 4) -eq 0 ]; then
```

```
    if [ "$year % 100" -ne 0 ]; then
        echo "This is a leap year, February has 29 days."
    else
        echo "This is not a leap year. February has 28 days."
    fi
else
    echo "This is not a leap year. February has 28 days."
fi
```

```
$ cat penguin.sh
#!/bin/bash
# This script lets you present different menus to Tux. He will only be happy
# when given a fish. We've also added a dolphin and (presumably) a camel.
if [ "$menu" == "fish" ]; then
    if [ "$animal" == "penguin" ]; then
        echo "Hmmmmmm fish... Tux happy!"
    elif [ "$animal" == "dolphin" ]; then
        echo "Pweetpeettreetppeterdepweet!"
    else
        echo "*prrrrrrt*"
    fi
else
    if [ "$animal" == "penguin" ]; then
        echo "Tux don't like that. Tux wants fish!"
        exit 1
    elif [ "$animal" == "dolphin" ]; then
        echo "Pweepwishpeeterdepweet!"
        exit 2
    else
        echo "Will you read this sign?!"
        exit 3
    fi
fi
```

```
$ cat feed.sh
#!/bin/bash
# This script acts upon the exit status given by penguin.sh
export menu="$1"
export animal="$2"
feed="penguin.sh"
$feed $menu $animal
case $? in
    1)
        echo "Guard: You'd better give'm a fish, less they get violent..."
        ;;
    2)
        echo "Guard: It's because of people like you that they are leaving earth all the time..."
        ;;
    *)
        echo "Guard: I don't know what that means."
        ;;
esac
```

```
3)      echo "Guard: Buy the food that the Zoo provides for the animals, you ***, how
        do you think we survive?"
        ;;
*)      echo "Guard: Don't forget the guide!"
        ;;
esac
$ ./feed.sh apple penguin
```
