Problem 1 *(File System)* (4 + 5 + 5 = 14 points)

What are the outputs of the following command sequences? If several outputs are possible, explain under which conditions which output is delivered. We assume that none of the commands produces an error and that environment variables such as `PATH` have their usual values.

Part (a)

```
rm -rf test
mkdir test
cd test
echo abc > f
ln -s f g
mv f h
echo def > f
cat g
```

Part (b)

```
rm -rf test
mkdir test
cd test
echo ghi > f
ln f g
echo jkl >> f
rm f
echo mno > f
cat g
```

Part (c)

```
rm -rf test
rm -rf /tmp/h
mkdir test
cd test
echo pqr > f
ln f g
mv f /tmp/h
echo stu > g
cat /tmp/h
```

Problem 2 *(Processes)* (5 points)

Which Unix command is used to send a signal to a process?

Problem 3 *(Processes, Shell Programming)* (5 + 5 = 10 points)

The Bourne shell script `testab` contains the following lines:

```
#!/bin/sh
B=bb
echo 1 A=$A B=$B
```

The command sequence

```
A=a
B=b
./testab
echo 2 A=$A B=$B
```

produces the output

```
1 A=a B=bb
2 A=a B=b
```

Part (a)

What can one say about the variable `A`?

(a1) `A` must have been declared as an environment variable previously.
(a2) `A` cannot have been declared as an environment variable previously.
(a3) There is not enough information to tell whether or not `A` has been declared as an environment variable previously.

Part (b)

What can one say about the variable `B`?

(b1) `B` must have been declared as an environment variable previously.
(b2) `B` cannot have been declared as an environment variable previously.
(b3) There is not enough information to tell whether or not `B` has been declared as an environment variable previously.

Problem 4 *(Shell Programming)* (20 points)

Write a Bourne shell script `ts` that calls the `scp` program to copy files or directories to `labserver.cs.uni-saarland.de`. If the first argument of `ts` is the option `-t`, `ts` copies its remaining arguments to the user's `thesis` subdirectory on `labserver.cs.uni-saarland.de`, otherwise it copies the arguments to the user's home directory. In other words,

`ts -t` *files*

shall be equivalent to

`scp -rp` *files* `labserver.cs.uni-saarland.de:thesis`

and

`ts` *files*

shall be equivalent to

`scp -rp` *files* `labserver.cs.uni-saarland.de:.`

If the first argument of `ts` starts with "`-`" but is different from `-t`, `ts` prints an error message to stderr and exits. Note: `ts` shall work correctly even if the file names contain spaces.

Problem 5 *(Quoting)* (15 points)

The command `qecho` writes its arguments to standard output; each argument is enclosed in »« and arguments are separated by one space. E. g., `qecho a b '' 'c d'` yields `»a« »b« »« »c d«`.
What is the output of the following command sequence? We assume that none of the commands produces an error and that environment variables such as `IFS` and shell options have their usual values.

```
rm -rf test
mkdir test
cd test
echo x > t1
echo y > t2
A=''
B='a ; cat t1'
C='*'
qecho 1 $A $B
qecho 2 "$A" "$B"
qecho 3 $C '"$C"' "'$C'"
```

Problem 6 *(Regular Expressions)* (8 + 8 = 16 points)

We have a file in which each line contains a Unix file name, at least one space, and a number. We assume that the file name does not contain newlines or quotes; it may contain spaces, but not at the end. Example:

```
logo.gif                               73
project meeting may 18                244
unixtest.c                           1656
```

Part (a)

Give a Perl substitution command that deletes in a line of the file the number at the end and the spaces separating the file name and the number.

Part (b)

Give a Perl substitution command that converts a line of the file into the format

`ln -s "`*filename*`" numfiles/`*number*

Example:

```
ln -s "logo.gif" numfiles/73
ln -s "project meeting may 18" numfiles/244
ln -s "unixtest.c" numfiles/1656
```

Aufgabe 1 *(Dateisystem)* (4 + 5 + 5 = 14 Punkte)

Welche Ausgaben erhält man bei den folgenden Befehlsfolgen? Falls mehrere Ausgaben möglich sind, erklaren Sie, unter welchen Bedingungen welche Ausgabe geliefert wird. Wir nehmen an, daß keiner der Befehle einen Fehler produziert und daß Umgebungsvariable wie etwa `PATH` ihre üblichen Werte besitzen.

Teil (a)

```
rm -rf test
mkdir test
cd test
echo abc > f
ln -s f g
mv f h
echo def > f
cat g
```

Teil (b)

```
rm -rf test
mkdir test
cd test
echo ghi > f
ln f g
echo jkl >> f
rm f
echo mno > f
cat g
```

Teil (c)

```
rm -rf test
rm -rf /tmp/h
mkdir test
cd test
echo pqr > f
ln f g
mv f /tmp/h
echo stu > g
cat /tmp/h
```

Aufgabe 2 *(Prozesse)* (5 Punkte)

Welcher Unix-Befehl dient dazu, einem Prozeß ein Signal zu schicken?

Aufgabe 3 *(Prozesse, Shell-Programmierung)* (5 + 5 = 10 Punkte)

Das Bourne-Shell-Skript `testab` enthält die folgenden Zeilen:

```
#!/bin/sh
B=bb
echo 1 A=$A B=$B
```

Die Befehlsfolge

```
A=a
B=b
./testab
echo 2 A=$A B=$B
```

liefert die Ausgabe

```
1 A=a B=bb
2 A=a B=b
```

Teil (a)

Was kann man über die Variable `A` aussagen?

(a1) `A` muß vorher als Umgebungsvariable (Environment-Variable) deklariert worden sein.
(a2) `A` kann nicht als Umgebungsvariable deklariert worden sein.
(a3) Die vorliegenden Informationen reichen nicht aus, um zu sagen, ob `A` vorher als Umgebungsvariable deklariert wurde oder nicht.

Teil (b)

Was kann man über die Variable `B` aussagen?

(b1) `B` muß vorher als Umgebungsvariable (Environment-Variable) deklariert worden sein.
(b2) `B` kann nicht als Umgebungsvariable deklariert worden sein.
(b3) Die vorliegenden Informationen reichen nicht aus, um zu sagen, ob `B` vorher als Umgebungsvariable deklariert wurde oder nicht.

Aufgabe 4 *(Shell-Programmierung)* (20 Punkte)

Schreiben Sie ein Bourne-Shell-Skript `ts`, das das Programm `scp` aufruft, um Dateien oder Verzeichnisse nach `labserver.cs.uni-saarland.de` zu kopieren. Falls das erste Argument von `ts` die Option `-t` ist, so kopiert `ts` seine verbleibenden Argumente in das Unterverzeichnis `thesis` des Benutzers auf `labserver.cs.uni-saarland.de`; anderenfalls kopiert es die Argumente in das Home-Verzeichnis des Benutzers. In anderen Worten:

> `ts -t` *files*

soll äquivalent sein zu

> `scp -rp` *files* `labserver.cs.uni-saarland.de:thesis`

und

> `ts` *files*

soll äquivalent sein zu

> `scp -rp` *files* `labserver.cs.uni-saarland.de:.`

Falls das erste Argument von `ts` mit „`-`“ anfängt, aber verschieden von `-t` ist, so gibt `ts` eine Fehlermeldung nach Stderr aus und beendet das Programm. Man beachte: `ts` soll auch dann korrekt funktionieren, wenn Dateinamen Leerzeichen enthalten.

Aufgabe 5 *(Quoting)* (15 Punkte)

Der Befehl `qecho` schreibt seine Argumente in die Standardausgabe; dabei ist jedes Argument in »« eingeschlossen und mehere Argumente sind durch je ein Leerzeichen getrennt. Beispielsweise liefert `qecho a b '' 'c d'` die Ausgabe `»a« »b« »« »c d«`.
Was ist die Ausgabe der folgenden Befehlsfolge? Wir nehmen an, daß keiner der Befehle einen Fehler produziert und daß Umgebungsvariable wie etwa `IFS` und Shell-Optionen ihre üblichen Werte besitzen.

```
rm -rf test
mkdir test
cd test
echo x > t1
echo y > t2
A=''
B='a ; cat t1'
C='*'
qecho 1 $A $B
qecho 2 "$A" "$B"
qecho 3 $C '"$C"' "'$C'"
```

Aufgabe 6 *(Reguläre Ausdrücke)* (8 + 8 = 16 Punkte)

Wir haben eine Datei, in der jede Zeile einen Unix-Dateinamen, mindestens ein Leerzeichen und eine Nummer enthält. Wir nehmen an, daß Dateinamen keine Zeilenumbrüche oder Anführungszeichen enthalten; sie dürfen Leerzeichen enthalten, aber nicht am Ende. Beispiel:

```
logo.gif                                73
project meeting may 18                 244
unixtest.c                            1656
```

Teil (a)

Geben Sie einen Perl-Substitutionsbefehl an, der in einer Zeile der Datei die Nummer am Ende und die Leerzeichen zwischen Dateiname und Nummer entfernt.

Teil (b)

Geben Sie einen Perl-Substitutionsbefehl an, der eine Zeile der Datei in das Format

`ln -s "`*dateiname*`" numfiles/`*nummer*

konvertiert. Beispiel:

```
ln -s "logo.gif" numfiles/73
ln -s "project meeting may 18" numfiles/244
ln -s "unixtest.c" numfiles/1656
```