

LAB 9 (virtual) – Bourne shell script

In this exercise you will be writing Bourne shell scripts to solve simple problems. The purpose of this exercise is to help you gain better understanding about shell script concepts including variable assignment and access, variable input and output, branches and loops.

1. Problem A

1.1 Specification

Write a Bourne shell script that keeps on reading user input of integers and for each integer examines (and displays) if the integer is positive, negative or zero.

1.2 Implementation

The program should:

- terminate when the user enters `quit`.

1.3 Sample Inputs/Outputs: assume the script is named `lab9a.sh`

```
indigo 338 % lab9a.sh
enter a number: 22
    a positive number
enter a number: 33
    a positive number
enter a number: -1
    a negative number
enter a number: 0
    this is zero
enter a number: -13
    a negative number
enter a number: quit
    bye bye
```

2. Problem B

2.1 Specification

Write a Bourne shell script that keeps on reading user inputs of integers and operators, and then calculates and displays the results for each line of inputs.

2.2 Implementation

Assume the input is of the form `operator 1 operator operator2`

The program should:

- accept operator + - x /
- give error message invalid operator (+ - x /) if other operator characters are entered
- terminates when the user enters quit.

2.3 Sample Inputs/Outputs:

```
indigo 406 % lab9b
enter op1 opr op2: 2 + 4
result: 2 + 4 = 6
enter op1 opr op2: 2 - 4
result: 2 - 4 = -2
enter op1 opr op2: 2 x 7
result: 2 x 7 = 14
enter op1 opr op2: 14 / 2
result: 14 / 2 = 7
enter op1 opr op2: 2 ^ 4
invalid operator ( + - x / )
enter op1 opr op2: 2 & 6
invalid operator ( + - x / )
enter op1 opr op2: 2 h 5
invalid operator ( + - x / )
enter op1 opr op2: quit
bye bye
indigo 407 %
```

3. Problem C

3.1 Specification

Extend problem B by also accepting ^ as a valid operator. Giving input a^b , the script calculates a to the power of b . That is, a^b

Hint: you need to write a loop to do the calculation manually. Perferable write a function power to do the calculation.

3.2 Sample Inputs/Outputs:

```
indigo 406 % lab9c
enter op1 opr op2: 2 x 4
result: 2 x 4 = 8
enter op1 opr op2: 3 ^ 4
result: 3 ^ 4 = 81
enter op1 opr op2: 2 ^ 5
result: 2 ^ 5 = 32
enter op1 opr op2: 4 ^ 2
result: 4 ^ 2 = 16
enter op1 opr op2: 4 + 3
result: 4 + 3 = 7
enter op1 opr op2: quit
bye bye
indigo 407 %
```

4. Problem D

4.1 Specification

Write a Bourne shell script that examines (and displays) the status of its command line input, which is treated as a filename.

4.2 Implementation

The program should:

- take (at least) one command line argument
- if no command line argument is given, output 'usage: lab9c filename'
Note that you should not hard code 'lab9c' in your script, as the script should be used for general purposes (e.g., when the name is changed.)
- if at least one command line argument is entered, then check if the first argument represents a file in the current directory.
- if the file does not exist, output 'filename does not exist as a file', where filename is the first command line argument
- if the file does exist, then check if it is empty. If it is empty, output 'filename is an empty file' where filename is the first argument
- if it is not empty, then list the status of the file, and then
 - check if it is executable. If it is executable, display 'filename is executable'
 - if it is not executable, make it executable, and display the status of the file again.

4.3 Sample Inputs/Outputs

```
indigo 420 % ls -l 0 00 000
-rwx--x--x  1 huiwang faculty 56 Jul 16 14:18 0
-rw-----  1 huiwang faculty 56 Jul 20 12:44 00
-rw-----  1 huiwang faculty  0 Jul 20 12:59 000
indigo 421 % lab9d
usage: lab9c file_name
indigo 422 % lab9c xx
xx does not exist as a file
indigo 423 % lab9c 000
000 is an empty file
indigo 424 % lab9c 0
-rwx--x--x  1 huiwang faculty 56 Jul 16 14:18 0
0 is an executable file
indigo 425 % lab9c 00
-rw-----  1 huiwang faculty 56 Jul 20 12:44 00
00 is executable now
-rwx--x--x  1 huiwang faculty 56 Jul 20 12:44 00
```

5. Problem E

5.1 Specification

Write a Bourne shell script that searches all the files whose name begins with the pattern given in its first command line argument, and then make a copy of all of them.

5.2 Implementation

The program should:

- be named `lab9e`
- take (at least) one command line argument
- if no command line argument is given, output `'usage: lab9d filenamePattern'`
- if at least one command line argument are entered, search and display all the files whose name begins with the input pattern (i.e., `filenamePattern*`).
- then copy each such file to a new file whose name is the name of the original file with a suffix `.xxx`. E.g., `abc` is copied to `abc.xxx`
- display (again) all the files whose name begins with the input pattern. Now the newly created files should also be displayed.

5.3 Sample Inputs/Outputs

```
indigo 396 % ls -l
total 16
-rwx----- 1 huiwang grad 56 Mar 29 20:01 0
-rwx----- 1 huiwang grad 56 Mar 29 20:01 0a
-rw----- 1 huiwang grad 0 Mar 29 20:01 0bc
-rwx----- 1 huiwang grad 56 Mar 29 20:01 0bc.bak
-rwx----- 1 huiwang grad 217 Mar 29 20:06 lab9e
indigo 397 % lab9e
Usage: lab9d file_namePattern
indigo 398 % lab9e 0
-rwx----- 1 huiwang grad 56 Mar 29 20:01 0
-rwx----- 1 huiwang grad 56 Mar 29 20:01 0a
-rw----- 1 huiwang grad 0 Mar 29 20:01 0bc
-rwx----- 1 huiwang grad 56 Mar 29 20:01 0bc.bak

-rwx----- 1 huiwang grad 56 Mar 29 20:01 0
-rwx----- 1 huiwang grad 56 Mar 29 20:01 0a
-rwx----- 1 huiwang grad 56 Mar 29 20:09 0a.xxx
-rw----- 1 huiwang grad 0 Mar 29 20:01 0bc
-rwx----- 1 huiwang grad 56 Mar 29 20:01 0bc.bak
-rwx----- 1 huiwang grad 56 Mar 29 20:09 0bc.bak.xxx
-rw----- 1 huiwang grad 0 Mar 29 20:09 0bc.xxx
-rwx----- 1 huiwang grad 56 Mar 29 20:09 0.xxx
indigo 399 %
```

Hint: a traditional way to do the problem is to use a loop. But you can also use one command to do all the searching and copying (and if you like, the listing too). How?