

# VE 280 Lab 1

---

## Ex0. A fun bash game BashCrawl (not mandatory)

---

A fun game that helps you get familiar with Linux command.

The game is attached in the files. See `/bashcrawl`. It has its README. It will be uploaded on Gitea soon. Have fun.

## Ex1. Hiding Photos

---

Related Topics: *Linux*.

Kyon is thinking of moving his favourite Mikuru's photos from SOS Quantum Library `sql/` to his own webserver `webserver/` without being noticed.

"If Haruhi knows this, Haruhi will destroy the world. We have to avoid this".

Yuki suggests Kyon to use command lines. So Kyon decides to implement a shell script to automate command line tasks. He may need these tasks:

- Create: create the working directory of the database and webserver, and initialize the database;
- Dump: dump the data in the database into the webserver;
- List: list all the files stored in the webserver (no need for showing hidden file);
- Display: display the data in the webserver;
- Destroy: remove the working directory of database and webserver.

He has already implemented the skeleton of `ex1.sh`, but now he . There are 5 of such lines in the script that are marked as `TODO` comments:

```
1 | # TODO: Replace this line with a linux command line.
```

Please help him finish this script by replacing the `TODO`s with a Linux command line you learned from lecture.

1. Create a directory named `sql/`.
2. Copy the file `sql/database.txt` to the directory `webserver/`.
3. List all files in directory `webserver/`.
4. Display `webserver/database.txt` in `stdout`.
5. Remove the `webserver/` and `sql/` directories.

# Testing

To test the script, please first [make it executable](#) by running:

```
1 | chmod +x ./ex1.sh
```

Then, you can test the script by running the following commands and observe the results.

```
1 | ./ex1.sh create  
2 | ./ex1.sh dump  
3 | ./ex1.sh list  
4 | ./ex1.sh display  
5 | ./ex1.sh destroy
```

## Ex2. Validating Password

Related Topics: *loops, arrays, boolean, ASCII*.

Miyamori Aoi is designing the website for Musani Animation. When designing the register page, she plans to write a function that checks whether the password that the user types in meets all the requirements below (or say it is valid):

- Contains at least 1 alphabetic character;
- Contains at least 1 numerical character;
- Contains at least 1 non-alphanumeric character.
- The max length of a password is 50 characters.

The function takes a password (an array of chars) as input, returns true if the password is valid and returns false if not.

```
1 | bool isValidPassword(char password[ ]){  
2 |     // TODO: Implement this function.  
3 | }
```

Miyamori is busy with their new animation. Please help her implement the function.

Note: In fact, when writing a web page, we use **Regular Expression** to validate whether a password meets our requirement. We do not use **C++** but we use **JavaScript** for web programming. For those interested, you can Google it yourself.

## Example

Example input:

```
1 | kibo~no~hana~~1
```

Example output:

```
1 | 1
```

## Ex3. Flipping an image

Related Topics: *loops, arrays*

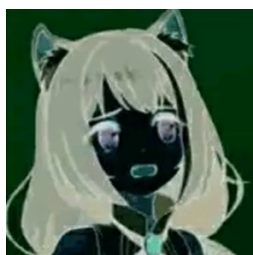
### Description

Here our topic is about the principle behind image processing software, like how to flip an image horizontally and invert it.

Like



Will result in



Of course this image will be too complicated for us to handle here. Let's simplify the problem.

We are given a  $n \times n$  binary matrix as a simple image.

To flip an image horizontally means that each row of the image is reversed.

- For example, flipping `[1,1,0]` horizontally results in `[0,1,1]`.

To invert an image means that each `0` is replaced by `1`, and each `1` is replaced by `0`.

- For example, inverting `[1,0,1]` results in `[0,1,0]`

# TODO

Just modify on the image. There is no need to return a new image here.

```
1 void flipAndInvertImage(int** image, int size){
2     // TODO: Implement this function.
3 }
```

## Input

The first line contains a number `n`, which is the size of the matrix `image`.  $1 \leq n \leq 50$

The following lines represents the matrix, which only contains `0` or `1`. Each row separated by a `\n` and each column separated by space.

## Output

The flipped and inverted matrix

## Example

**Example input:**

```
1 3
2 1 1 0
3 1 0 1
4 0 0 1
```

**Example output:**

```
1 1 0 0
2 0 1 0
3 0 1 1
```

**Explantion:**

First flip horizontally

```
1 0 1 1
2 1 0 1
3 1 0 0
```

Then invert it

```
1 1 0 0
2 0 1 0
3 0 1 1
```

