Cardiff School of Computer Science and Informatics
Coursework Assessment Pro-forma

Module Code:	CMT304
Module Title:	Programming Paradigms
Lecturer:	Philipp Reinecke and Frank Langbein
Assessment Title:	Script programming and quantum computing
Assessment Number:	2
Date Set:	24 February 2020
Submission date and Time:	27 April at 9:30am
Return Date:	25 May 2020

This assignment is worth 25% of the total marks available for this module. If coursework is submitted late (and where there are no extenuating circumstances):

1. If the assessment is submitted no later than 24 hours after the deadline, the mark for the assessment will be capped at the minimum pass mark;
2. If the assessment is submitted more than 24 hours after the deadline, a mark of 0 will be given for the assessment.

Your submission must include the completed official Coursework Submission Cover sheet, which can be found here:
`https://docs.cs.cf.ac.uk/downloads/coursework/Coversheet.pdf`

Submission Instructions

All submission must be via Learning Central. Upload the following files in a **single zip file**, `[student number].zip`:

Description		Type	Name
Cover Sheet	**Compulsory**	One PDF (.pdf) file	`[student number].pdf`
Task 1.1	**Compulsory**	One Perl source file	`task1.pl`
Task 1.2	**Compulsory**	One PDF file	`task1.pdf`
Task 2	**Compulsory**	One PDF (.pdf) file	`task2.pdf`

Any Perl code submitted will be run on the system `lapis.cs.cf.ac.uk` and must be submitted as stipulated in the instructions above.

Any deviation from the submission instructions above (including the names, number and types of files submitted) may result in a mark of zero for the assessment or question part.

Staff reserve the right to invite students to a meeting to discuss coursework submissions.

Your submissions will be checked for plagiarism. Your work must be your own and you must independently solve the problem and submit your own solution. Any other material or sources of information you use must be referenced. Code and text you submit will be compared with other submissions and various other sources on and off the Internet. Any substantial similarities of your submission to unreferenced work or material not created by yourself will be subject to unfair practice procedures. Marks will only be assigned for work you have done yourself (incl. finding and discussing material from references, but not the referenced work; there are no marks for code copied from elsewhere, but for either writing your own code or integrating and adapting code that you have not written).

Assignment

Task 1.1: Consider the following situation:

> A set of image files (GIF, PNG, and JPG) have been accidentally renamed wrongly: The file endings have been removed. For instance, a file whose original name was `test.gif` is now called `test`. The files may be mixed with files whose file endings are intact and with files that are not of type GIF, PNG, or JPG, and they may be spread across subdirectories. You cannot install software on the machine you are using.

Your task is to write a Perl script that restores the original file endings for all such files in a set of directories and their subdirectories. That is, after running the script, each GIF file must have the file ending `.gif`, each PNG file must have the file ending `.png`, and each JPG file must have the file ending `.jpg`. All other files must be left untouched. In particular, image files that already have the correct file ending should be left alone.

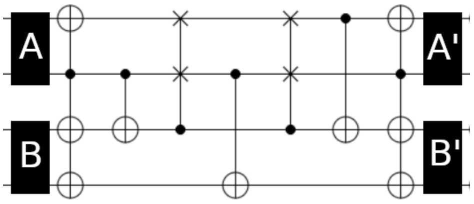
It must be possible to run the script from the command line as follows:

```
task1.pl [directories]
```

where `[directories]` is a space-separated list of directories where the file endings need to be restored. For each directory, the script should print the number of files of each type whose file endings have been restored.

Task 1.2: Discuss, in at most 300 words, why script programming is particularly suited for this problem, and describe two possible extensions of your script and how they could be implemented.

Task 2: Consider the following quantum circuit:



consisting of several multi-target CNOT and controlled swap gates. It has two two-qubit input quantum registers $|A\rangle$ and $|B\rangle$ and two two-qubit output quantum registers $|A'\rangle$ and $|B'\rangle$.

Analyse the operation of the circuit. In particular, derive the full operator of the circuit and describe the mapping from $|A\rangle|B\rangle$ to $|A'\rangle|B'\rangle$. Based on this mapping, what operation does the circuit implement?

In addition, construct a circuit that inverts this operation, mapping $|A'\rangle|B'\rangle$ back to the input $|A\rangle|B\rangle$ values. Justify that your circuit is correct.

Learning Outcomes Assessed

Discuss and contrast the issues, features, design and concepts of script programming and quantum computing.

Criteria for assessment

Task 1.1 [40 marks]

Your script must run on `lapis.cs.cf.ac.uk`. The script will be assessed as follows:

1. The script will be run against the directory `images/` in the provided archive `example.zip`. It will be run in the directory `example/` as follows:

   ```
   rm report.txt
   task1.pl images/ > report.txt
   ```

   After execution, the contents of `images/` should be exactly like the contents of the directory `images-after/` in `example.zip`, the file `report.txt` should exist, and it must have the exact same contents (including the format) as the file `report.txt` in `example.zip`.

2. Your script will be run against several other directories, using different combinations of parameters that are allowed according to the description, and checked as described above.

3. Your script will be manually inspected for code quality.

Breakdown of marks

Please note: Your script must not depend on any software not present on the specified machine (`lapis.cs.cf.ac.uk`), and you must not request that any software be installed on that machine. Any such request will be denied. The script will be executed with a Bash shell. **If the script does not run exactly as specified, no attempt will be made to make it run, and you will not earn any marks**.

Criteria	Detailed criteria	Range marks
Operation with example case	The script runs	[0,1]
	GIF files renamed correctly	[0,2]
	PNG files renamed correctly	[0,2]
	JPG files renamed correctly	[0,2]
	All other files untouched	[0,1]
	Report as specified and correct	[0,2]

Operation with unknown cases	GIF files renamed correctly	[0,4]
	PNG files renamed correctly	[0,4]
	JPG files renamed correctly	[0,4]
	All other files untouched	[0,4]
	Report as specified and correct	[0,4]
Error handling	Parameter errors	[0,1]
	File/directory errors	[0,2]
	Error messages	[0,2]
Code quality	Comments present and helpful	[0,1]
	Clear structure	[0,1]
	Use of existing tools	[0,3]

Task 1.2 (10 marks). Marks will be awarded for the clarity and correctness of the argument ([0, 6]) and for the discussion of possible extensions ([0, 4]).

Task 2 [50 marks]. Answers should be provided in a report. Marks will be awarded according to the criteria below. You may use a program/script, e.g. in Python as in the lectures, quirk, or manual calculations for your answers, but you must show the results in the report.

Criteria	Detailed criteria	Range marks
Quantum circuit analysis	Correct circuit operator	[0,15]
	Correct input-output mapping	[0,5]
	Operation identified correctly	[0,5]
Inverse quantum circuit	Correct inverse circuit	[0,15]
	Proof of correctness	[0,10]

Feedback and suggestion for future learning

Feedback on your coursework will address the above criteria. Feedback and marks will be returned on 25th May 2020 via Learning Central. This will be supplemented with oral feedback on request.

Feedback from this assignment will be useful for the final exam.