Week 10 Assessment Questions

Please note:

- Just 1 program to submit.
- Note the early submission deadline: Monday, 10 August, at 11:00:00am.

1. (Random numbers)

   Write a C program that generates a random word (consisting of just lower-case letters a–z). When executed, the first command-line argument is the length of the word, and the second argument is the seed used by the random number generator, i.e. by `rand()`.

   An example of the program executing could be

   ```
   prompt$ ./randword 6 2
   ebgnha
   ```

   which seeds the random-number generator with 2 and generates a random word of length 6.

   We have created a script that can automatically test your program. To run this test you can execute the `dryrun` program that corresponds to this exercise. It expects to find a file named `randword.c` in the current directory.

   You can use dryrun as follows:

   ```
   prompt$ 9024 dryrun randword
   ```

   Note: It is important in this exercise that you call the random number generator as specified in the lecture, otherwise the outputs will not match. If you use a different method to generate random numbers, the testcases will 'fail', but your program could still be perfectly correct. The output may also be different on your own computer.

Assessment

Submit your solution using the following **give** command:

```
prompt$ give cs9024 week10 randword.c
```

Make sure you spell the filenames correctly. You can run **give** multiple times. Only your last submission will be marked.

The deadline for submission is **Monday, 10 August 11:00:00am**.

The program is worth 2 marks. Auto-marking will be run by the lecturer several days after the submission deadline using different test cases than `dryrun` does.

Hints:

- Programs will not be manually marked.
- It is important that the output of your program follows exactly the format shown above, otherwise auto-marking will result in 0 marks.
- **Ensure that your program compiles on a CSE machine with the standard options `-Wall -Werror -std=c11`. Programs that do not compile will receive 0 marks.**
- `dryrun` and auto-marking also check the correct usage of dynamic memory allocation and pointers as well as the absence of memory leaks.
- Do your own testing in addition to running `dryrun`.

Plagiarism

Group submissions will not be allowed. Your programs must be entirely your own work. Plagiarism detection software will be used to compare all submissions pairwise (including submissions for similar assessments in previous years, if applicable) and serious penalties will be applied, including an entry on UNSW's plagiarism register.

- ***Do not copy ideas or code from others***
- ***Do not use a publicly accessible repository or allow anyone to see your code***

Please refer to the on-line sources to help you understand what plagiarism is and how it is dealt with at UNSW:

- Plagiarism and Academic Integrity
- UNSW Plagiarism Policy Statement
- UNSW Plagiarism Procedure