

Project – Assessed Work (50%)

You are to design, implement, test and compare 2 separate solutions to a **string search problem**:

1. Sequential
2. Parallel using any library/language you wish (eg: MPI, OpenMP, HADOOP, ...)

Problem Overview:

Scientists have discovered different types of alien genetic material, sharing common structure:

- 3 dimensional lattice of cubes of distinctive components (letters)
- Each lattice has a finite number of letters in its alphabet – from 2 up to 100
- The lattice sizes range from $4*4*4$ up to $1000*1000*1000$

Scientists have also discovered dictionaries of words from the genetic alphabet:

- The dictionary size ranges from 3 up to 1000 words
- All words in a common dictionary share the same length, from 2 up to 100

The scientists hypothesise that dictionaries are **associated** to some of the genetic material, provided all words in the dictionary can be found in sequence (horizontal, vertical or diagonal) in the material lattice. You are to write a program that takes as input any lattice material and any dictionary, and returns a boolean stating whether there is an **association** between the two.

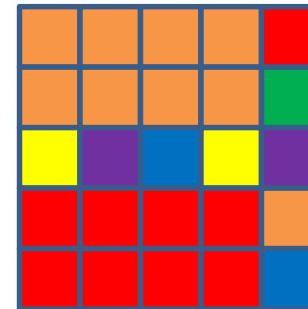
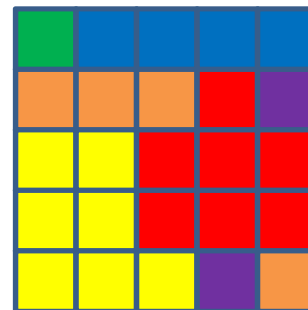
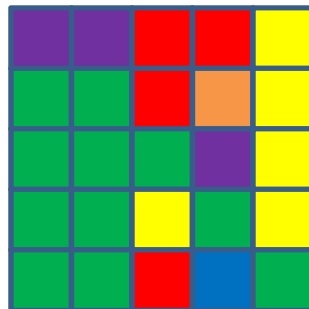
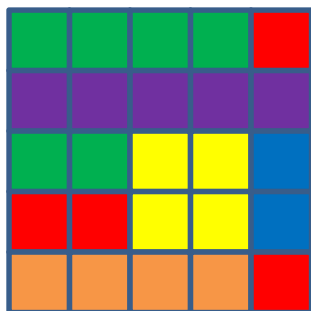
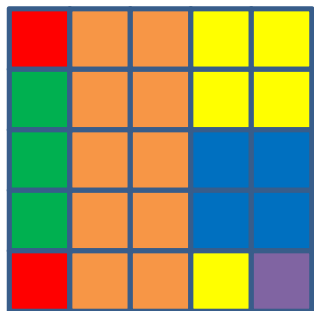
Project – Assessed Work (50%)

Example

Dictionary1 = **a** **b** **c** **b**, bcca, aabc

Dictionary2 = **a** **b** **c** **d** **e**, ebcda, edbac

Dictionary3 = **a** **b** **b** **b** **a**, bbbba, aabaa



Which of these dictionaries are associated to the 5*5 lattice of genetic material (if any)?

The Assessment

You are to write 2 programs that take as input any lattice material and any dictionary, and return a boolean stating whether there is an **association** between the two:

1. Sequential, 2.Parallel

All code should be well documented and well tested. Code which does not function correctly must be documented as such. You will be penalised for not documenting known bugs or problems. You are encouraged to re-use code that you believe will help you to implement your solution. All re-used code must be documented as such, and you are advised to comprehensively test such code against your specific requirements.

You are required to use a unit test tool and a documentation generation tool.

- By default, if you are programming in C, we recommend CUnit and Doxygen.

You must also analyse and compare the 2 different solutions:

- Performance - Scaleability
- Design Quality - Maintainability/Extensibility