Mathematics, C++ Assessment 1, 2015-6

This assessment is worth 40% of the total marks available for C++ programming. The hand-in date is 4 December 2015.

Constructing a seating plan for your daughter's wedding.
Your daughter is getting married and it is necessary to construct a seating plan for the reception. There are M = 40 guests sitting at K = 4 circular tables, each with N = 10 place settings, so M = KN. One place at each table is a seat of honour, the head seat. Members of F families, F = 6, have to be invited. The 6 families are the Scarletts, the Mustards, the Whites, the Greens, the Peacocks, and the Plums. Each family has a husband and wife and various other members. The head seat at each table must be occupied by either the husband or wife of a family.

Unfortunately members from these dysfunctional families do not always get along with one another. You must construct a seating plan that minimizes the potential conflicts.

Computing potential conflict
Every ordered of pair of individuals has an animosity index measuring the degree of dislike felt by the first individual towards the second individual. At each table, k = 1, ..., K, index the seats from 1 to N, in order clockwise from the table's head seat, with the head seat having index 1. Write x^k_{ij} for the animosity felt by the person in seat i towards the person in seat j at table k, $i \neq j \in \{1, ..., N\}$, $k \in \{1, ..., K\}$.
Write $y^k_{ij} = (\delta_j x^k_{ij} + \delta_i x^k_{ji})\|i - j\|^2$ where

$$\delta_i = \begin{cases} R, \text{ with } R = 5, & \text{if } i = 1, \\ 1, & \text{otherwise,} \end{cases}$$

and $\|i - j\|$ is the number of places around the table, in the *longest* direction, between i and j. The closer i and j are the *greater* the potential conflict.

y^k_{ij} is the conflict potential between individuals in positions i and j at table k. Pairs of people where the people in the pair are sitting at different tables do not contribute to potential conflict. Animosity is multiplied by R = 5 if the antagonist is in the head seat because of the resentment felt towards that person by those not sitting in place of honour.

Your task is to seat the guests so as to minimise, as far as you can, the conflict potential at the reception. Use two measures of conflict potential:

$$J_1 = \sum_{k=1}^{K} \sum_{\substack{i,j \in \{1,\dots,N\} \\ i \neq j}} y^k_{ij}.$$

$$J_2 = \max_{k=1,\dots,K} \max_{\substack{i,j \in \{1,\dots,N\} \\ i \neq j}} y^k_{ij}.$$

Under each measure find two arrangements: the first in which guests may be seated anywhere, and the second in which in each family the husband and wife must be seated at the same table.

The families and their animosities
The family members are:

Scarlett	Mr Scarlett, Mrs Scarlett, and 5 others, labelled S1, ..., S5.
Mustard	Colonel Mustard, Mrs Mustard, and 5 others, labelled M1 ,..., M5.
White	Mr White, Mrs White, and 5 others, labelled W1, ..., W5.
Green	Reverend Green, Mrs Green, and 5 others, labelled G1, ..., G5.
Peacock	Mr Peacock, Mrs Peacock, and 4 others, labelled Pe1, ..., Pe5.
Plum	Mr Plum, Professor Plum, and 4 others, labelled Pl1, ..., Pl5.

All animosities are 0 except between the husbands and wives of the 6 families. In each family the husband and wife have the animosities towards one another shown in Table 1.

Family:	Scarlett	Mustard	White	Green	Peacock	Plum
Husband to wife:	4	3	1	2	3	4
Wife to husband:	4	3	4	3	2	1

Table 1. Husband and wife animosities

All other animosities are represented in figure 1. The arrows represent the animosity felt in the direction of the arrow. by the husband and wife in one family towards the husband and wife of the other family. The label on each arrow denotes the strength of the animosity.

In the figure x′ = x mod 3 where x is the final digit in your student ID, and y′ = y mod 3 where y is the penultimate digit in your student ID.

Note that the level of conflict is determined solely by the placement of the husband and wife in each family. The placement of the other family members has no effect on the conflict level. When you describe an arrangement you need specify only the placements of the husbands and wives.

You must write a C++ application to

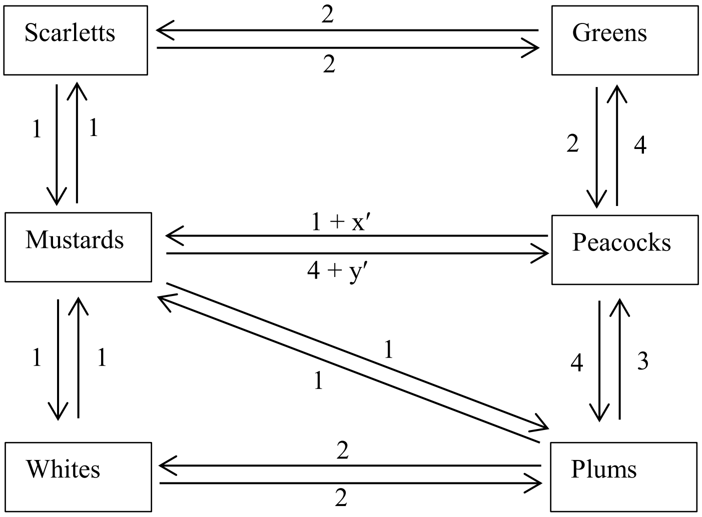
i) Construct data on individuals and their animosities.

ii) Find the four seating arrangements.

iii) Prints out to console your seating plans in a client friendly form, together with the value of the conflict potential of the arrangement.

You will need to find and implement an algorithm to perform the task set. The algorithm must be general in that K and M and the various animosities between the husbands and wives can be arbitrary.

Figure 1. Inter-family animosities



You do *not* need to find an optimal arrangement. The emphasis for the assessment is on the quality of your code, not on the sophistication of the algorithm you have implemented. There are various algorithms that can be applied to this problem, for example genetic algorithms, but unsophisticated heuristic algorithms are quite acceptable for this assessment. A brute force comparison of seating arrangements would need (KN)!/(KN – 2F)! comparisons. In general this is too many.

One acceptable approach would be to first use a heuristic to assign husbands and wives to tables, and then to optimize the seating at each table. The implementation should be computationally efficient in that it should avoid performing redundant or duplicated calculation.

There are advantages in making the application object oriented but it is not required that you should do so. However you must avoid using C-style constructs, including C-style arrays, unless necessary. Code should be split appropriately between different files.

You should construct a clear user interface and write code in a clear and maintainable style. The code you submit must to be able to run without addition or modification with the DevCpp installation on the machines in the teaching area.

The report

Your report must include a clear description of the algorithm you implement and an assessment of your results. See the general requirements. You will be assessed on

a) The quality of your code in terms of readability, maintainability, clarity, sophistication, generalizability, and general presentation.

b) The degree to which your code has the required functionality.

c) Your assessment of the efficacy of your implementation.

d) The presentational quality of the assessment as a whole.

Group work is not permitted

You are encouraged to discuss the assessment with other students but the code you submit must be yours alone. Clear similarities in code, in the write-up or in the results, will be taken as evidence of group work (for instance if identical or very similar tables of results are given, or if code is clearly shared.)

Code closely modelled on that from other sources is not permitted

Code to perform components of this exercise is likely be found in various places and might, for instance, be downloadable from the web. Use of code taken from, or cosmetically altered from, such sources is not permitted. You must devise your code from scratch.

Specifically *excluded* from this prohibition is code I have declared that you may use. In particular you may use the library code I have made available, without attribution.

Nick Webber