QA EA3

GRIFFITH COLLEGE DUBLIN

COMPUTING ASSIGNMENT TITLE SHEET

Course:	B.Sc. in Computing (Level 7 and Level 8)
Stage/Year:	I
Module:	Computer Programming
Semester:	II
Assignment Number:	IV
Date of Title Issue:	10/03/20
Assignment Deadline:	**31/03/20**
Assignment Submission:	Submitted on Moodle
Assignment Weighting:	**10%**

Assignment Title

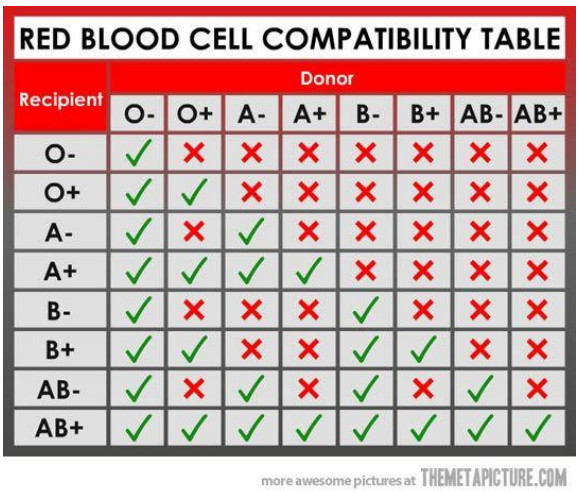
Blood Transfusion Manager

Make an application that efficiently chooses donors for blood transfusion.

When the application is launched it should try to open two files: "donors.txt" and "recipients.txt". If one of the files is missing the program should announce the problem and exit. Both files should be formatted in the following way: each row should contain a person's full name and their blood type separated by semicolon. The program should first read donors.txt, split each line into name and blood type and store the resulting array as a new element in a donors arraylist. It should also print the list on the screen and check that each person's blood type is valid. If an invalid blood type is found that entry should not be added to the arraylist and the user should be notified which entry had a problem. Recipients should then be read, processed and stored (in recipients arraylist) in a similar manner.

20 marks

Your application should be able to take blood type of a potential recipient and blood type of a potential donor and answer whether a blood transfusion is possible. To implement this you should use the table below: (try not to use if else statements as they will get confusing for debugging.Instead use but some kind of structure 2 dimensional Boolean array 8x8 replace green with true and red exes with false statements from above table)



Because it is very important that mistakes are not made when choosing a compatible blood type, information/logic from the table above should be stored in a manner that is easy to verify and debug. Ideally it should be a 2D Boolean array with hardcoded values.

20 marks

As can be seen from the table, some recipients have many options when it comes to donor blood type, while others are very limited. Also, there is a limit on how many blood donations one person can do per year. Therefore, we will try to match donors and recipients efficiently. Our goal is to have each donor have as few transfusions as possible thus reducing waiting time. To do this we should first find out which donors each recipient can receive blood from and record this information in an arraylist where each element corresponds to a recipient and contains an array with booleans that show their compatibility with each of the donors. The result will function like a 2D Boolean array where rows correspond to recipients and columns correspond to donors. However, thanks to it being an arraylist we will be able to remove rows as we match recipients and donors.

.

20 marks

The next step is to efficiently match recipients and donors. We will assume that each recipient only needs one transfusion. When a blood type is chosen the whole array, row is removed. First, we will try to match recipients that have limited options. If a recipient has only one potential donor, they are matched immediately. If they have more than one option, we need to look at how many potential (plus existing) matches each of their potential donors has and choose the one with the fewest potential matches. If these numbers are identical, the first/random donor is chosen. Once a match is decided, the recipient-donor pair is recorded and the recipient's row is removed from the arraylist. This process is repeated until all recipients are matched with a donor.

20 marks

After all the matches were made, appointments need to be arranged for each transfusion. The hospital cannot do more than 12 transfusions per day, the procedure is not done on Saturday or Sunday, and a donor needs to wait at least 56 days before their next transfusion. As such it makes sense to start with the busiest donor. Appointment information should be printed on the screen and to a file "appointments.txt". The earliest appointment date is 2 days after the day of program execution. Information about each appointment should appear in the following format: instead of dates use calendar class.

```
-----------------------------------------------------------------
Recipient: <Recipient Name>    Blood Type: <Recipient's Blood Type>
Donor: <Donor Name>            Blood Type: <Donor's Blood Type>
Date: <Date of Appointment>
-----------------------------------------------------------------
```

20 marks

Your donors file should have 8 donors with O+, 5 donors with A+, 4 with B+, 4 with O-, 2 with A-, 1 B-, 1 AB+ and 1 AB-. Your recipients file should have twice as many people with a similar proportion of blood types. The order should be random.

QA EA3

You will be required to demonstrate your program during the submission week.