CSC5524
Problem 1: Quality
Scenario: Job Application & Final
Selection Procedure

Welcome to the recruitment center for *Amafacesoftle.* As you know, our company is famous for its innovative and pioneering 3Q-approach: Quality People, Quality Process and Quality Products. It is perhaps, for this reason that you have applied for the post of quality manager at 1 of our 5 software and system quality control divisions around the world?

You have already passed through a very competitive selection process; and those of you that are present today can be proud of the fact that we would like to employ each and every one of you. However, as I said earlier, there are only 5 posts available. As a consequence, we invite you to participate in this final selection procedure.

Everything that you do is being observed (!)... from the moment that you started to read this sheet of paper. The selection will be based on the qualities that we search in our engineers and managers: honesty and integrity, technical ability, application and motivation, and team-work. Please read the following instructions very carefully, and follow them precisely:

INSTRUCTIONS

You are to organize yourselves into teams of 4 people. I will decide what to do with any remaining 1,2 or 3

people. In the end, each of you will be in a team of 4 (or 3).

Give your team a name. Email me – at paul.gibson@telecom-sudparis.eu - your team name, team members' names, and cc the email to each member.

The selection is a competition between each team – all members of the winning team will be given a job. The remaining position(s) will be allocated by me (based on my observations of individual candidates).

Each team will be evaluated by its peers, based on criteria decided-upon by you. The work is to be done in 2 stages:

1) TODAY

Each team will work on 3 software problems. Each team will submit a solution (full or partial) to each problem at the end of today's session (by email to my address, above).

Each team will also identify a set of <u>generic criteria</u> by which they will judge the quality of their solution. They will then self-assess their 3 solutions based on these criteria. Finally, they will weight the criteria in order to normalize a final mark in the range 0-100. This self-evaluation will also be submitted to me at the end of today's session (also by email)

2) PREPARATION FOR NEXT SESSION

I will gather and distribute all the submissions by email (before Friday 19th May), and each team will then

evaluate the other teams' solutions to the chosen problems, based on <u>their own generic quality criteria</u>. Based on their evaluations, they will rank the solutions to each problem (team-by-team) and email me the evaluations and rankings (before Monday 29th May).

RULES OF THE GAME:
  1) All code must be written in Java
  2) You have access to the web for whatever you want
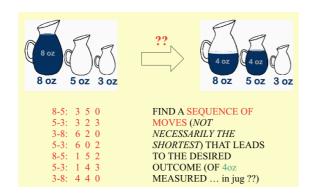
PROBLEM 1 – Triangle classification

There are 3 types of triangle – equilateral, isosceles and scalene.

You are to write a program which will take as input the specification of a triangle, which may be in 1 of 2 forms:

  1) 3 co-ordinates in 2 dimensional space
  2) 3 lengths for each of the sides

and produce as output the type of triangle that corresponds to the input specified

PROBLEM 2 – Water Jugs

8-5:	3 5 0	FIND A SEQUENCE OF
5-3:	3 2 3	MOVES (*NOT*
3-8:	6 2 0	*NECESSARILY THE*
5-3:	6 0 2	*SHORTEST*) THAT LEADS
8-5:	1 5 2	TO THE DESIRED
5-3:	1 4 3	OUTCOME (OF 4oz
3-8:	4 4 0	MEASURED … in jug ??)

A 'Formal' Description of the Problem

Consider three jugs that (when full) measure integer quantities x,y and z.

Starting with 1 jug full, x say, measure some quantity q in (any) one of the 3 jugs by making a sequence of valid moves

A valid move pours water from one jug into the next, either:
- •Emptying the first jug, and perhaps filling the second; or
- •Filling the second jug, and perhaps leaving some water in the first.

Input, x,y,z,q
Output: sequence of moves leaving q in either x, y or z, or an empty sequence when no solution is possible.

PROBLEM 3 – Overlapping Lines.

Consider the integer number line:

... -8 -7 -6 -5 -4 -3 -2 -1 0 +1 +2 +3 +4 +5 +6 +7 +8 ...

We can define a segment on this line by a range (minimum ... maximum)
Below, we illustrate 2 segments: **(-3, 1)** and **(0,6)**

... -8 -7 -6 -5 -4 -3 -2 -1 0 +1 +2 +3 +4 +5 +6 +7 +8 ...

In this example, the 2 segments are said to overlap on the line because they share at least 1 point in common.
The overlap in this case is the segment (0, 1).

The problem is to write a program that can calculate whether any 2 segments overlap on the integer line.