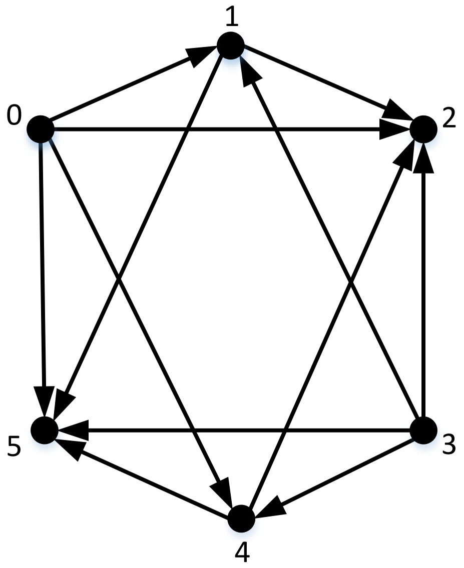
PART A – GRAPH THEORY – 20 MARKS

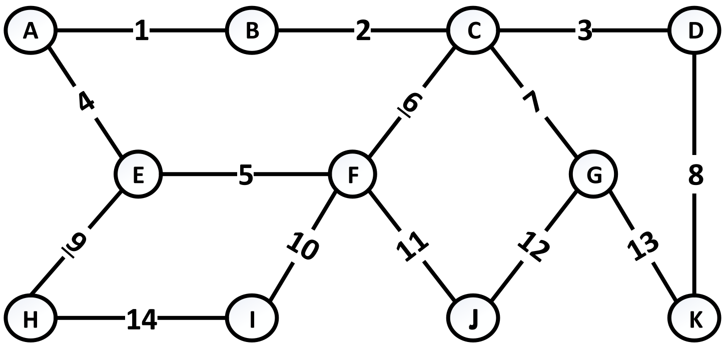
1. Graph of a Relation (4 marks)

Draw the directed graph of the following relation R in the set of vertices S={0,1,2,3,4,5}
∀x,y∈S xRy iff x mod 3 < y mod 3



2. Circuits (6 marks)

This question is based on the following graph G (the edge numbers are edge names):



a) Starting at vertex A, give an Euler circuit for G (listing the **vertices and edges** as they are traversed) or explain why this cannot be done

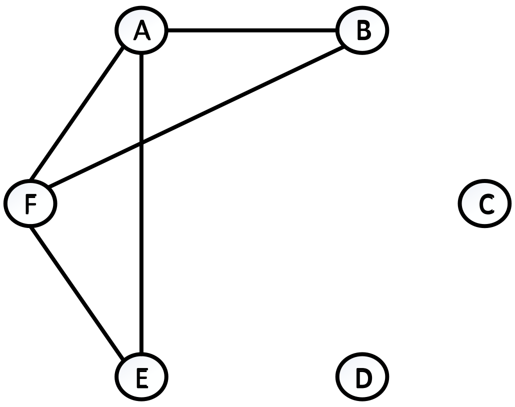
The graph has no Euler circuit because verticess E and G have odd degrees.

b) Starting at vertex A, give a Hamiltonian circuit for G (listing the **vertices and edges** as they are traversed) or explain why this cannot be done.

A1B2C3D8K13G12J11F10I14H9E4A

3. Connectedness and Complements (10 marks)

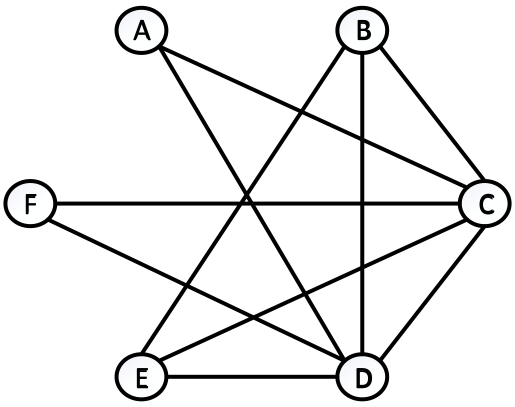
This question is based on the following graph G:



a) List all the connected components of G. Each connected component should be described as the set of all the vertices in the connected component.

G has 3 connected components: {C}, {D}, and {A,B,E,F}

b) Draw the complement G^c of the graph G



c) Using the same format as in a) list all the connected components of G^c

G^c has one connected component: {A, B, C, D, E, F}

PART B – REGULAR EXPRESSIONS AND FINITE STATE AUTOMATA – 40 MARKS

1. Operations on Languages (10 marks)

Define the following two languages of the alphabet Σ={0,1,2}:
L_1 = {0, 01, 02}
L_2 = {ε, 2, 02}

a) List all the elements of $L_1 \cap L_2$

{ 02 }

b) List all the elements of $L_1 \cup L_2$

{ 0, 01, 02, ε, 2 }

c) List all the elements of $L_1 \times L_2$

{ (0,ε), (0,2), (0,02), (01,ε), (01,2), (01,02), (02,ε), (02,2), (02,02) }

d) List all the elements of $L_1 L_2$

{ 0, 02, 002, 01, 012, 0102, 022, 0202 }

2. Regular Expression (10 marks)

Write a regular expression to match all *sets* in a new programming language. *Sets* are strings like "{}", "{740}", "{hello,799,0,55,friend}" and they are defined as follows:

- A *set* is a list of zero of more *entries* surrounded by curly parentheses.
- If the list contains more than 1 *entry*, the *entries* are separated by commas.
- An *entry* is either a *name* or an *integer*
- A *name* is a string of 1 or more lower-case letter (i.e. a to z)
- An *integer* is either the digit 0 or a string of one or more digits which does not start with the digit 0

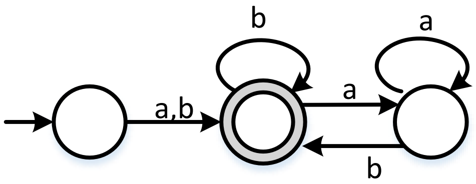
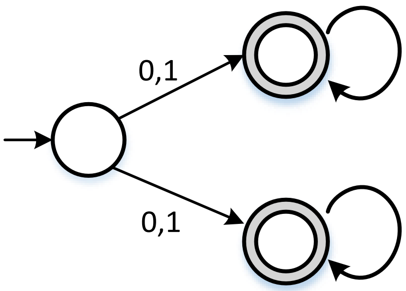
You do **not** need to simplify your regular expression

{ ε | (0 | [1-9][0-9]* | [a-z]+) (, (0 | [1-9][0-9]* | [a-z]+))* }

(matching parentheses are shown in colour to improve legibility.)

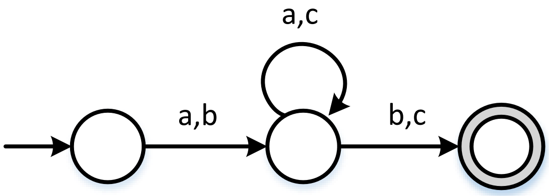
3. Finite State Automata (20 Marks)

a) Give a regular expression for each of the following finite state automata. Make these regular expressions as simple as possible.

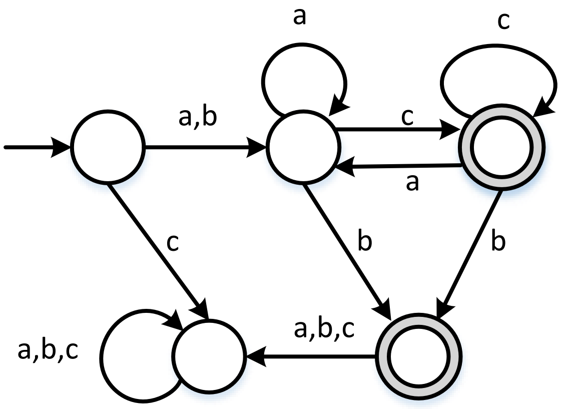
Automaton	Regular expression
	(a|b) (b | a^+b)*
	(0|1) (0* | 1*)

In the next two questions the simplest possible automaton refers to an automaton with as few states as possible.

b) Draw the simplest possible NFA (**non-deterministic** finite state automaton) on an input alphabet I={a,b,c} which recognizes the following regular expression: (a|b)(a|c)*(b|c)



c) Draw the simplest possible DFA (**deterministic** finite state automaton) on an input alphabet I={a,b,c} which recognizes the following regular expression: (a|b)(a|c)*(b|c). Your DFA should handle all possible inputs


a
c
a,b
c
a
c
b
b
a,b,c
a,b,c