This MP is to be done individually. You may wish to consult the Russell & Norvig recommended text sec. 9.2.2, alg. p. 328 (copies on reserve). You may also consult web resources. Wikipedia gives a more-than-adequate discussion; the second link below gives and explains the algorithm in Russell & Norvig.


http://cs.stackexchange.com/questions/35280/can-someone-clarify-this-unification-algorithm

**Part A:** Write two Python functions:

**unify(e1, e2)** that returns ‘Fail’ if the expressions e1 and e2 cannot be unified or returns a Python dictionary (possibly empty) which is the most general unifier of the two expressions.

**apply(e, subst)** that returns a new expression which is the expression e with the unifier subst applied.

An expression is either a single string, a single number, or a tuple of expressions. A variable is any string starting with the question mark character.

Some examples:

1) unify('here', 'there') should return ‘Fail’
2) unify('here', '?there') should return {'?there': 'here'}
3) unify(('P', '?x', '?x'), ('P', ('Q', ('+', 3, '?y')), ('Q', ('+', '?z', '?z')))) could return
   {'?x': ('Q', ('+', 3, 3)), '?y': 3, '?z': 3} or any equivalent unifier
4) unify(('P', '?x', '?x'), ('P', ('Q', ('+', 3, '?y')), ('Q', ('+', 4, 3)))) should return ‘Fail’
5) unify(('P', '?x', '?x'), ('P', ('Q', ('+', 3, '?y')), ('Q', 8))) should return ‘Fail’
6) apply(('Q', '?x'), unify(('P', '?x', '?x'), ('P', ('Q', ('+', 3, '?y')), ('Q', ('+', '?z', '?z')))))
   should return ('Q', ('+', 3, 3))

**Part B:** Briefly answer

1) How hard would it be to have (5) above return {'?x': ('Q', ('+', 3, '?y')), '?y': 5}? Explain.
   (hint: it is not very hard, but explain)

2) Would such an extension be a good idea?
   (hint: thought question; form an interesting specific opinion and defend it)