1. Define a Lisp function `mix` of two arguments `(x y)` that are lists, and that forms a new list where the elements of x and y occur alternatingly. For example,

```
(mix '(a b c) '(m n p))
```

should have the value

```
(a m b n c p)
```

Decide yourself what to do if the two arguments do not have the same length, and explain in words what you have decided. (When your solution is graded, it will be important that your program is consistent with your description in words of your decision).

It is recommended to solve the problem choosing only among the following Lisp functions and operations:
car, cdr, cons, equal, if, cond

2. Two function definitions in Lisp are said to be equivalent if the functions have the same number of arguments, and for every sequence of arguments of the right length, either of the following two applies:
(a) both functions run into an error (for example, an undefined function), or
(b) neither of them runs into an error, and both have the same value and the same side-effects (if any).
This must hold regardless of context, that is, regardless of what are the values of variables or definitions of other functions when the functions in question are called.

With this definition, are the following two function definitions equivalent or not? If yes, explain why. If no, give an example that proves your answer.

```
(defun func1 (x) #'(lambda (y)(+ x y)))
(defun func2 (y) '(lambda (x)(+ x y)))
```

Then, suppose criterium (a) above is changed to be
(a´) at least one of the functions runs into an error
Question: does this change the answer to the question? In what way?

For your information. The definition of func1 can also be written as follows:

```
(defun func1 (x) (function (lambda (y)(+ x y))))
```

according to the definition of the symbol `#'`

3. a) Define what it means for a partial-order plan to be consistent.

b) Is it possible that a linearization of a consistent partial-order plan is inconsistent? If yes, give an example. If no, explain why it is not possible.

4. Consider the following two formulas in first-order predicate logic:

$$\forall\ x, y\ [\ \mathrm{Haseaten}(x,y) \rightarrow \neg\ \mathrm{Hungry}(x)\ ]$$
$$\forall\ x, y\ [\ \mathrm{Haseaten}(x,y)\ \Lambda\ \mathrm{Badfood}(y) \rightarrow \mathrm{Sick}(x)\ ]$$

a) Give two examples of interpretations for these two formulas, one that is a model and one that is not.

b) Rewrite these two formulas as clauses, i.e. on the form that is used by the resolution method.

5. Consider the following three effect axioms for actions as expressed in the situation calculus:
type(y,lamp) → haslight(x, switchon(x,y,s))
type(y,newspaper) → haslight(x, ignite(x,y,s))
type(y,window) Λ daylight-outside(s) → haslight(x, open(x,y,s))

a) Explain in ordinary words the plausible meanings of these axioms.

b) Write the reverse frame axioms that correspond to the given axioms.

6. Consider the following learning problem. One is given a function f of two arguments,
`f(p,x) = y`
and a set of pairs $(x_1, y_1), (x_2, y_2), ...$
It is desired to find a value for `p` such that $f(g,x_i)$ is equal to, or approximates y_i as well as possible for all i. Describe briefly the following two methods that can be used for this problem:
a) Simulated annealing
b) Genetic programming

7. What is the order of magnitude for the space and the time that is required for parsing a sentence using a chart parser, expressed as a function of the number of words in the sentence.