1. (a) Parser-generators such as Yacc, Bison and CUP classify errors in grammars as being either *shift/reduce conflicts* or *reduce/reduce conflicts*. Explain the difference between these and say how they are resolved by any one of the parser-generators listed above.

[7 marks]

(b) Give an example of a simple grammar which has a *shift/reduce* conflict.

[5 marks]

(c) Give an example of a simple grammar which has a *reduce/reduce* conflict.

[5 marks]

(d) A modification to the Tiger language is being considered which will allow the result of the value computed for the expression within an assignment statement to be used in an enclosing expression, as is the case with assignment in C. In order to evaluate this idea it has been decided to implement parsing and abstract syntax generation for the fragment of the language which includes only this modified form of assignment, integer constants, variables and the addition operator. The grammar should ensure that

```
x := y := z + 1
```

is parsed as

```
x := (y := (z + 1))
```

Give a grammar for this fragment of the language together with the semantic actions for construction of the abstract syntax representation of the input. You should state your answer in the form of the input to the Yacc/Bison parser generator (with the semantic actions coded in C) *or* in the form of the input to the CUP parser generator (with the semantic actions coded in Java). In either case, you may assume the existence of the constructor functions `IntExp`, `VarExp`, `PlusExp` and `AssignExp` with the obvious definitions.

[8 marks]

2. (a) Activations of functions within a program are allocated a *stack frame*. What is the layout of a typical stack frame on the stack? Identify the kinds of data values which are stored there.

[6 marks]

(b) Computer manufacturers often suggest a preferred frame layout to be used on their architecture. What is the benefit to be gained by using the suggested frame layout?

[3 marks]

(c) In a language such as Tiger or C, what are the conditions under which a local variable of a function is said to *escape* from the function? If a variable escapes then in what way does this influence the allocation of storage for that variable?

[6 marks]

(d) The following Tiger function has formal parameters `a` and `b` and local variable `A`.

```
function f (a : int, b : int) : int =
let  type intArray = array of int
     var A := intArray[12] of 0
     function g (A : intArray) : int = A[a]
in   (A[b] := a; g (A))
end
```

(i) Which of the three variables `a`, `b` and `A` escape from the function? Explain your answer in the case of each variable.

[6 marks]

(ii) What optimisation could be applied to the function `f` in order to reduce the number of variables which escape? Which of `a`, `b` and `A` would be affected by this optimisation?

[4 marks]

3. (a) Describe the *liveness analysis* phase of compilation. What is its purpose and how is it performed? Give the dataflow equations for liveness analysis.

[8 marks]

(b) There are two significant variants on the iterative algorithm which performs liveness analysis. What are these two variants? Which one has faster convergence in general?

[3 marks]

(c) Consider the following program.

$$
\begin{array}{rl}
 & a \leftarrow 0 \\
 & b \leftarrow 1 \\
L_1: & c \leftarrow a + 1 \\
 & b \leftarrow b * 2 \\
 & \text{if } c = N \text{ goto } L_2 \\
 & a \leftarrow c \\
 & \text{goto } L_1 \\
L_2: & \text{return } b
\end{array}
$$

(i) Determine the liveness of the variables a, b and c.

[9 marks]

(ii) Based on this analysis, how could later phases of the compilation process optimise both the storage requirements and the run-time of the program?

[5 marks]