1. [10 marks] The grammar shown below is LL(2).

\[
\begin{align*}
1 & \quad A \rightarrow a \ B \ a \ a \\
2 & \quad \rightarrow \ b \ C \ b \ a \\
3 & \quad B \rightarrow \ b \\
4 & \quad \rightarrow \ \lambda \\
5 & \quad C \rightarrow \ b \\
6 & \quad \rightarrow \ \lambda
\end{align*}
\]

Where \( \lambda \) is the empty string

a) Show the First and Follow sets for the non terminals A, B and C.

b) Show the Predictor Sets for this grammar.
2. [20 marks] Consider the program fragment shown below.

```
20  integer J, K, L, N ;
21  real A [ 1 .. 100 , 1 .. 101 ] ;
   /* reals use 4 bytes of storage */

   /* Assume A is given a value here */

30  N := 100
31  for J := 1 to N - 1 do
32     for K := J + 1 to N do
33        for L := J + 1 to N + 1 do
35        end
36     end
37  end
38  end
```

Describe the classical machine independent optimizations that an optimizing compiler would perform on the code fragment on lines 30 to 38

3. [15 marks] Describe the semantic analysis checks that a compiler would perform on the code fragment in Question 2.

4. [10 marks] During the course project two changes were made to the definition of anonymous functions in the project language.

   1. exit and exit when can not be used in the statement part of an anon function unless they are inside a while or loop construct.

   2. Both forms of the return statement cannot be used in the statement part of an anon function unless they are inside a procedure or function.

   Explain why each of these changes simplified the implementation of the project language. Give examples of problems that would arise if these changes had not been made.
5. [20 marks] Some programming languages include a tagged version of the case / union data structure. For example in Modula-2:

```plaintext
type shapeKind = (square, rectangle, circle);
type shape =
  record
    centerx : integer ;
    centery : integer ;
  case kind : shapeKind of
    square : ( side : integer ) ;
    rectangle : ( length, height : integer ) ;
    circle : ( radius : integer ) ;
end ;
```

The record type `shape` can be used to create variables like any other type. The case tag `kind` indicates which alternative of the case is currently active.

Languages with a tagged case usually enforce strict access control to the alternatives of a case. For example:

*Any attempt to access a member of the data structure whose existence depends on a particular value of the case tag, while the case tag is not the expected one, raises a run time error.*

For example trying to use the value of `radius` when `kind` has the value `square`.

Describe the run time mechanisms necessary to enforce the restrictions on access to fields in the tagged data structure.

6. [15 marks] Given the declarations:

```plaintext
boolean P, Q, R[20]
integer J
```

Using the quadruples from the lecture notes show the quadruples that would be generated for the statement:

```plaintext
```
7. [10 marks] In the ANSI C programming language the definition for floating point constants is:

- a floating-point constant can take one of two forms:
  - a group of digits from 0 to 9 which includes a single period, like 3.141593, 0. or .1
  - a group of digits from 0 to 9, possibly containing a period, followed by 'e' or 'E' and an exponent, like 1.0e3, 2.593e+7 or 3141593e-6

- a floating-point constant may be followed by: an 'f' or 'F' to indicate a float constant, like 35.79f Note that 123f is an error, because 123 is not a floating-point constant

Write one regular expression that could be used to recognize floating point constants in ANSI C.

---

8. [20 marks] A proposal has been made to extend the project language to allow arrays in procedures and functions to have a dynamic size, i.e. the dimensions of the array are determined by runtime expressions. Example:

   ```
   procedure flexi ( integer N, integer M )
   begin
       integer J , K
       % some code here
       loop
       get J , K
       boolean P[ J , K ]
       % more code here
   end
   % more code here
   end
   % flexi
   ```

1. Describe the runtime mechanisms necessary to implement this proposed feature.

2. You still have a chance to influence the language designer over this proposal. What changes would you request to make the proposal easier to implement?

---

Total Marks = 120