1. [20 marks] A regular expression definition for numeric constants is given below.

Digit
OptionalSign
Number
Integer
RealNumber

Digit ( 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 )
OptionalSign ( ‘+’ | ‘-’ | λ )
Number Integer | RealNumber
Integer Digits+
RealNumber Digit* ( ‘.’ | λ ) Digit* ( ( E | e ) OptionalSign Digit+ | λ )

Assume you are working with a very simple scanner that only returns single character lexical tokens. Write a LL(1) grammar for parsing Number. Your grammar should enforce the requirement that the mantissa part of a real number must contain at least one digit. Prove that your grammar is LL(1) by showing the director sets for each rule. You may assume the set follow(Number), but describe what it can not contain.

2. [20 marks] A proposal has been made to slightly simplify the project language by removing the result reserved word and replacing it with return. The two return-type statements would then be:

return
return expression

a) [5 marks] Does this change have any significant effect on language?
b) [15 marks] How will this change affect parsing the language? Explain your answer.

3. [10 marks] In some languages new variable names can be dynamically created at run time. For example build up a string by concatenation and then use the value of the string as the name of a new variable. This feature is heavily used to create map-like associative tables. Newly created variables have a null/zero value until assigned to.

Describe a set of runtime mechanisms that will support the creation and use of dynamically created variables.
4. [20 marks] One problem that a compiler has to deal with initialization of variables in their declaration. For example:

```c
#define BIGDATASIZE <some large integer constant>

typedef struct {
    int count;
    char * name;
    float value;
} dataStruct;

dataStruct bigData[ BIGDATASIZE ] =
    { { 3, "scorp", 4.7 }, { 6, "podger", 9.2 },
      { 12, "framus", 3.8 }, { 19, "spanner", 7.3 }.
    /* Assume many more values here */
```

a)[10 marks] Describe the semantic analysis checking required for this type of declaration.

b)[10 marks] Describe a complete scheme for the general case of handling storage and access to the initial value for arrays and structs with large initial values. Describe how the values are stored and how they are made available to later passes of the compiler. How hard would it be to implement this scheme in a strongly typed language like Java?

5. [15 marks] Given the C structure declaration:

```c
struct L1 {
    char key;
    double value;
    short kind;
    union {
        struct L1 * self;
        double value2;
        struct {
            char key2;
            double value3;
        } bigVal;
        short sData[ 9 ];
    } bigU;
    struct L1 * next;
}
```

Show how this structure would be laid out in memory using the depth first structure mapping algorithm discussed in lecture. You should assume the size and alignment factors (in bits) in the table below.

<table>
<thead>
<tr>
<th>Type</th>
<th>Size</th>
<th>Alignment</th>
</tr>
</thead>
<tbody>
<tr>
<td>char</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>short</td>
<td>16</td>
<td>16</td>
</tr>
<tr>
<td>int</td>
<td>32</td>
<td>32</td>
</tr>
<tr>
<td>float</td>
<td>32</td>
<td>32</td>
</tr>
<tr>
<td>double</td>
<td>64</td>
<td>64</td>
</tr>
</tbody>
</table>

6. [15 marks] Show the symbol and type table entries that a typical compiler would make for the structure declaration in Question 5.