1. **[20 marks]** Something like:

```
ndigit  →  1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
digit   →  0 | ndigit
group   →  , digit digit digit
number  →  $*^0(0 | ndigit ( λ | digit | digit digit | group* ) ( λ | . digit digit )
```

Where λ is the empty string and * is the Kleene star.

2. **[20 marks]**

The grammar is AMBIGUOUS, hence not LL(anything)

Predict set conflict on E → λ

<table>
<thead>
<tr>
<th>Predict Set</th>
</tr>
</thead>
<tbody>
<tr>
<td>9 E → a B</td>
</tr>
<tr>
<td>10 → b A</td>
</tr>
<tr>
<td>11 → λ</td>
</tr>
</tbody>
</table>

Most common problem was not computing the Predict Set for E → λ correctly.
3. [20 marks]

First map each case

**case 1**

\[ \begin{array}{cccc}
   x & y & \text{shadow} & z \\
   32 & 24 & 8 & 64 \\
\end{array} \]

length: 128  alignment: 64

**case 2**

\[ \begin{array}{cccc}
   d & e \\
   8 & 56 & 64 \\
\end{array} \]

length: 128  alignment: 64

\[ \begin{array}{cccc}
   a & b & c \\
   40 & 24 & 64 & 128 \\
\end{array} \]

length: 256  alignment: 64

**case 3**

\[ \begin{array}{cc}
   s & t \\
   360 & 24 & 64 \\
\end{array} \]

length: 448  alignment: 64

Case 3 is the largest so it determines the size of the union. Finally map the entire union

\[ \begin{array}{c}
   u\text{Tag} \\
   \text{case 3} \\
   16 & 48 & 448 \\
\end{array} \]

length: 512  alignment: 64

The most common problems were failing to map the entire union and laying out the union cases sequentially rather than overlapping them.
4. [20 marks]

Fragment A

1. Check that method with this signature not already declared
2. Check that I is not already declared in this scope.
   Check that 0 can be assigned to I
3. Check that I is declared
   Check the I < 10 is a legal comparison
4. Check that J is not already declared in this scope.
   Check that - can be assigned to J
5. Check that I is declared
   Check that ++ can be applied to I
6. Check that System is defined and is a class
   Check that System has a field out
   Check that System.out.println is defined
   Check that a System.out.println that takes an int argument exists
   Check that I is declared

Fragment B

1. Check that K is not already declared in this scope
   Check that 0 can be assigned to K
   Check that K is declared
   Check that data is declared
   Check that data has a field length
   Check that K < data.length is a legal comparison
   Check that K is declared
   Check that ++ can be applied to K
2. Check that data is declared
   Check that data is a 1-dimensional array
   Check that K is declared
   Check that K is a valid subscript for data [ ]
   Check that target is declared
   Check that data[K] == target is a legal comparison
3. Check that index is declared
   Check that K is declared
   Check that K can be assigned to index
4. Check that break is inside a loop

Curiously, many people missed checking the variable target on line 2
5. [20 marks]

(anon)

uTag

x

y

z

a

b

c

d

e

s

t

union 3

subRange 1 .. 3

uCase 1

uCase 2

uCase 3

string 3

array

subrange 1 .. 5

record 2

string 1

array

subrange 1 .. 9

string 5

int builtin

real builtin

boolean builtin