1. Consider the language $L_1$ defined by the grammar $G_1$:

1. $S ::= A \, S$
2. $A ::= B \, s$
3. $A ::= P \, s$
4. $A ::= s$
5. $B ::= [ \, A$
6. $P ::= < \, A >$

1(a) (10 points) Write a recursive descent parser for the language. You can assume the existence of a `nexttoken` and `match` function. Your parser need not check for errors, need not take any semantic actions, and need not produce any output.

1(b) (5 points) Based on your answer to 1(a), is the grammar LL(1)?

1(b) (13 points) Show the LR(0) characteristic finite state machine (CFSM) for the grammar. A partial CFSM is provided, you only need to fill in states 9 and 10, and the edges entering and leaving those states. Circles contain the number of a state that the edge is connect to.

1 (c) (5 points) Based on your answer to 1(b), is the grammar an LR(0) grammar? Why or why not (a brief answer of a sentence or two is sufficient).
2. Given the loop nests

\textbf{Nest 1}
\begin{verbatim}
for (i=0; i < n; i++) {
    for (j=0; j < n; j++) {
        a[i+1, j] = a[i,j]
        b[i-1, j] = b[i,j]
    }
}
\end{verbatim}

\textbf{Nest 2}
\begin{verbatim}
for (j=0; j < n; j++) {
    for (i=0; i < n; i++) {
        a[i+1, j] = a[i,j]
        b[i-1, j] = b[i,j]
    }
}
\end{verbatim}

2(a) (10 points) Describe the dependences on the "a" array in loop Nest 1. Either say "no dependence", or, if a dependence exists, give the type (flow or true, output or anti), the direction and the distance.

2(b) (10 points) Which loop(s), if any, can be parallelized in loop Nest 1?

2(c) (5 points) Both loops execute the same number of operations and require the same number of registers. What can cause their performance to be significantly different?

3. Given the loop nest:

\begin{verbatim}
k = 0
for (i=0; i < n; i++) {
    for (j=0; j < n; nj++) {
        t1 = 2 * i;
        t2 = k + 1;
        t3 = t1 + t2;
    }
}
\end{verbatim}

3(a) (5 points) Give the USE and DEF sets for the i loop.

3(b) (5 points) Give the USE and DEF sets for the j loop.

3(c) (5 points) Given your answer to 3(a), what computations in the three assignment statements in the inner loop can be moved out of the i loop?

3(d) (5 points) Given your answer to 3(b), what computations in the three assignment statements in the inner loop can be moved out of the j loop?
4. Given the two code fragments:

```
Nest 1
for (i=0; i < n; i++) {
    a[i] = b[i] + c[i]
    d[i] = a[i] + e[i]
    f[i] = d[i] + a[i]
}

Nest 2
for (i=0; i < n; i++) {
    a[i] = b[i] + c[i]
    for (i=0; i < n; i++) {
        d[i] = a[i] + e[i]
    }
    for (i=0; i < n; i++) {
        f[i] = d[i] + a[i]
    }
```

4(a) (4 points) Which loop nest potentially requires more cache misses? Why? (A short answer is sufficient.)

4(b) (4 points) Which loop nest requires more registers to perform a register allocations without spills? Assume that variable values must be accessed from a register.

4(c) (4 points) What is the name of the transformation going from Nest 1 to Nest 2?

5. Given the node representing block b, taken from a dataflow graph, as shown below, with the given In(b) and Out(b) sets:

```
5(a) (5 points) What is a possible definition of the KILL set in terms of variables used and defined in the block b below if the analysis is a backwards analysis? As an example, one possible answer might be "All variables used in the block".

5(b) (5 points) What is a possible definition of the GEN set in terms of variables used and defined in the block b below if the analysis is a backwards analysis?
```

```
In(b)  c, d, e, f

Block b
a = b + c
b = a + d
a, c, e

Out(b)  f
```