1. (35 pts total, -5 pts for each error, but not negative) MINI-MAX SEARCH IN GAME TREES.
The game tree below illustrates a position reached in the game. Process the tree left-to-right. It is Max's turn to move. At each leaf node is the estimated score returned by the heuristic static evaluator.

1.a. Fill in each blank square with the proper mini-max search value.

1.b. What is the best move for Max? (write A, B, or C) ________

1.c. What score does Max expect to achieve? ____________

2. (35 pts total, -5 for each error, but not negative) ALPHA-BETA PRUNING. Process the tree left-to-right. This is the same tree as above (1.a). You do not need to indicate the branch node values again.

Cross out each leaf node that will be pruned by Alpha-Beta Pruning. Do not just draw pruning lines.

**** TURN PAGE OVER AND CONTINUE ON THE OTHER SIDE ****
3. (30 pts total, -5 for each error, but not negative) RESOLUTION THEOREM PROVING. You are engaged in Knowledge Engineering for the Wumpus Cave. You have interviewed an expert on the Wumpus Cave who told you, among other things, “A stench in square (1,2) is equivalent to a wumpus in square (1,1) or (2,2) or (1,3). A stench in square (2,1) is equivalent to a wumpus in square (1,1) or (2,2) or (3,1).” You translated this into propositional logic as

\[(S_{12} \Leftrightarrow W_{11} \lor W_{22} \lor W_{13})\]  \[(S_{21} \Leftrightarrow W_{11} \lor W_{22} \lor W_{31})\]

and then into Conjunctive Normal Form (CNF) as

\[\neg S_{12} \lor W_{11} \lor W_{22} \lor W_{13}\land (S_{12} \lor \neg W_{11}) \land (S_{12} \lor \neg W_{22}) \land (S_{12} \lor \neg W_{13})\]
\[\neg S_{21} \lor W_{11} \lor W_{22} \lor W_{31}\land (S_{21} \lor \neg W_{11}) \land (S_{21} \lor \neg W_{22}) \land (S_{21} \lor \neg W_{31})\]

Now it is time for the first “live” test of your system. An agent has been lowered down into the Wumpus cave, and reports back by radio, “Square (1,1) has no wumpus and no stench. Square (1,2) has a stench. Square (2,1) has no stench.” You translate this knowledge into CNF as “(\neg W_{11}) \land (\neg S_{11}) \land (S_{12})” and add it to your knowledge base.

Next the agent asks by radio, “Is it true that square (1,3) has a wumpus?” You translate this query into propositional logic as the goal sentence “(W_{13}).” You form the negated goal as “(\neg W_{13}).” Now your knowledge base plus the negated goal, expressed in clausal form, is:

\[\neg S_{12} W_{11} W_{22} W_{13}\]
\[\neg S_{21} W_{11} W_{22} W_{31}\]
\[(S_{12} \neg W_{11})\]
\[(S_{12} \neg W_{22})\]
\[(S_{12} \neg W_{13})\]
\[(S_{21} \neg W_{11})\]
\[(S_{21} \neg W_{22})\]
\[(S_{21} \neg W_{31})\]
\[\neg W_{11}\]
\[\neg S_{11}\]
\[(S_{12})\]
\[\neg S_{21}\]
\[\neg W_{13}\]

Run resolution on this knowledge base until you produce the null clause, “( )”, thereby proving that the goal sentence is true. The shortest proof I know of is only five lines long. It is OK to use more lines, if your proof is correct.

Repeatedly choose two clauses, write one clause in the first blank space on a line, and the other clause in the second. Apply resolution to them. Write the resulting clause in the third blank space, and insert it into the knowledge base.

Think about what you are trying to prove, and find a proof that mirrors how you think. You know S12 and (S12 \Rightarrow W_{11} \lor W_{22} \lor W_{13}). You know (\neg W_{11}). It is easy to prove (\neg W_{22}), so (W_{13}) is the only possibility left. Your negated goal is (\neg W_{13}). You seek ( ). Think about it.

Resolve \_______________ and \_______________ to give \_______________

Resolve \_______________ and \_______________ to give \_______________

Resolve \_______________ and \_______________ to give \_______________

Resolve \_______________ and \_______________ to give \_______________

Resolve \_______________ and \_______________ to give \_______________

Resolve \_______________ and \_______________ to give \_______________

Resolve \_______________ and \_______________ to give \_______________

Resolve \_______________ and \_______________ to give \_______________

Resolve \_______________ and \_______________ to give \_______________

Resolve \_______________ and \_______________ to give \_______________

Resolve \_______________ and \_______________ to give \_______________

Resolve \_______________ and \_______________ to give \_______________

Resolve \_______________ and \_______________ to give \_______________

Resolve \_______________ and \_______________ to give \_______________

Resolve \_______________ and \_______________ to give \_______________