Question 1: 25% of students lost more than 2 points

Question 2: 50% of students lost 2-4 points; 50% got it entirely correct

Question 3: 95% of students got this problem entirely correct

Question 4: Only two students got this problem entirely correct. Most made serious errors in all 4 search strategies and gave either incorrect or insufficient answers (e.g. 'because it found the shortest path') in part e. The inclusion of the self loop confused some students, but many students explored nodes in really weird ways (e.g. running BFS in the iterated deepening search).

Question 5: 10% of students lost 2 or more points; most did well

Question 6 and 7: Only a handful of students lost points on these questions

Question 8: Half the class drew an incorrect state space - i.e. proposed invalid transitions or included forbidden states.

Question 9: 30% of the class confused the ordering of the steps

Question 10: 30% of the class did not understand resolution. 80% of the students incorrectly resolved part d.

Question 11: 30% of the students did not resolve correctly.

Question 12: 25% of the class made errors - usually incorrectly applying the distributive law.

The Mid-term exam is now a pedagogical device.

You can recover 50% of your missed points by showing that you have debugged and repaired your knowledge base.

For each item where points were deducted:

Write 2-4 sentences, and perhaps an equation or two. Describe:

 * What was the bug in the knowledge base leading to the error?

 * How has the knowledge base been repaired so that the error will not happen again?

Turn in, with your exam, in class on Thursday, 1 March. 50% of your missed points will be forgiven for each correct repair.

Your Mid-term Exam will be returned to you at the end of class today (Tue, 21 Feb) or upon request from the TA, Andrew Gelfand.

CS-171, Intro to A.I. — Mid-term Exam — Winter Quarter, 2012

YOUR NAME AND ID NUMBER: _____

YOUR ID: _____ ID TO RIGHT:_____ ROW:_____ NO. FROM RIGHT:_____

The exam will begin on the next page. Please, do not turn the page until told.

When you are told to begin the exam, please check first to make sure that you have all ten pages, as numbered 1-10 in the bottom-left corner of each page.

The exam is closed-notes, closed-book. No calculators, cell phones, electronics.

Please clear your desk entirely, except for pen, pencil, eraser, an optional blank piece of paper (for optional scratch pad use), and an optional water bottle.

This page summarizes the points available for each question so you can plan your time.

1. (12 pts total, 1 pt each) LOGIC CONCEPTS.

2. (5 pts total, -1 for each error, but not negative) ALPHA-BETA PRUNING.

3. (5 pts total, 1 pt each) CONSTRAINT SATISFACTION PROBLEMS.

4. (10 pts total, 2 pts each) STATE-SPACE SEARCH.

5. (15 pts total, 1 pt each) AGENT/SEARCH CONCEPTS.

6. (5 pts total, -2 pts each wrong answer, but not negative) TASK ENVIRONMENT.

7. (10 pts total, -1 each wrong answer, but not negative) SEARCH PROPERTIES.

8. (10 pts total, -1 each error, but not negative) STATE SPACE SEARCH.

9. (10 pts total, -2 each error, but not negative) OPTIMALITY OF A* SEARCH.

10. (4 pts total, 1 pt each) RESOLUTION.

**11. (10 pts total, 1 pt each) PROVE THAT THE UNICORN IS MAGICAL.**

12. (4 pts total, -1 each error, but not negative) CONJUNCTIVE NORMAL FORM (CNF).

1. (12 pts total, 1 pt each) LOGIC CONCEPTS. For each of the following terms on the left, write in the letter corresponding to the best answer or the correct definition on the right. The first one is done for you as an example.

|			
A	Agent	A	Perceives environment by sensors, acts by actuators.
C	Syntax	B	Chain of inference rule conclusions leading to a desired sentence.
I	Semantics	C	Specifies all the sentences in a language that are well formed.
L	Entailment	D	Describes a sentence that is true in all models.
J	Sound	E	Stands for a proposition that can be true or false.
K	Complete	F	Represented as a canonical conjunction of disjunctions.
E	Propositional Symbol	G	Possible world that assigns TRUE or FALSE to each proposition.
D	Valid	H	Describes a sentence that is false in all models.
M	Satisfiable	I	Defines truth of each sentence with respect to each possible world.
H	Unsatisfiable	J	An inference procedure that derives only entailed sentences.
B	Proof	K	An inference procedure that derives all entailed sentences.
G	Model	L	The idea that a sentence follows logically from other sentences.
F	Conjunctive Normal Form	M	Describes a sentence that is true in some model.

2. (5 pts total, -1 for each error, but not negative) ALPHA-BETA PRUNING. The game tree below illustrates a position reached in the game. It is **Max**'s turn to move. Inside each leaf node is the estimated score of that resulting position returned by the heuristic static evaluator.
CROSS OUT EACH LEAF NODE THAT WILL BE PRUNED BY ALPHA-BETA PRUNING.
You do not need to indicate the branch node values.

Red lines indicate where in the tree pruning occurred. You are not obliged to provide the red lines — only to cross out pruned leaf nodes.
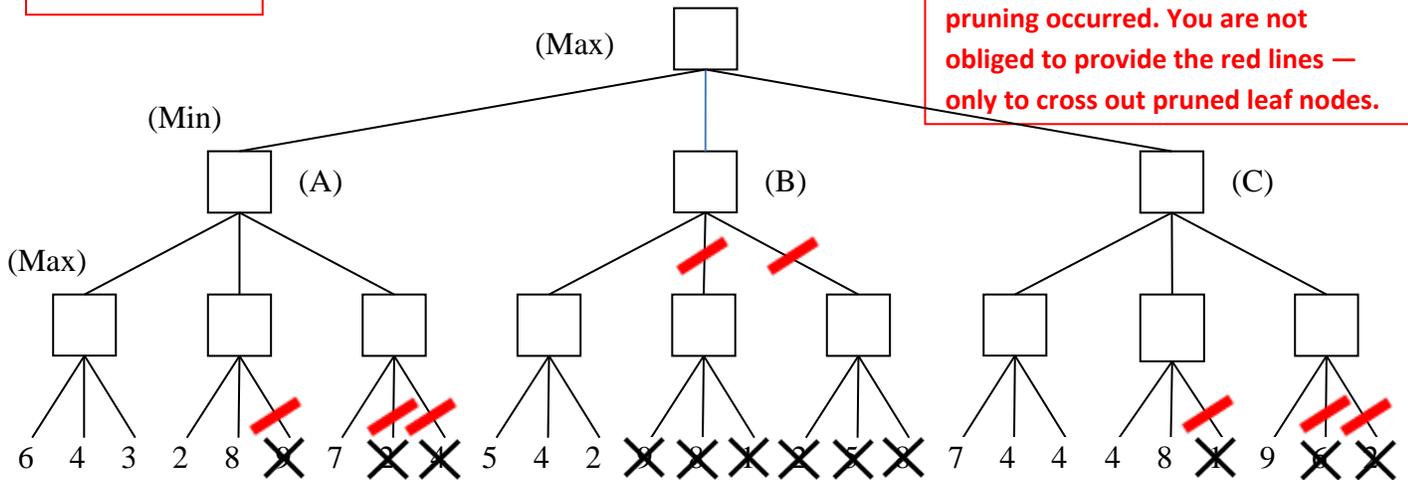
**** TURN PAGE OVER AND CONTINUE ON THE OTHER SIDE ****

3. (5 pts total, 1 pt each) CONSTRAINT SATISFACTION PROBLEMS. This problem asks about the Map Coloring Problem. Each region must be colored one of Red (R), Green (G), or Blue (B). Neighboring regions must be a different color. The map (left) and constraint graph (right) are below.
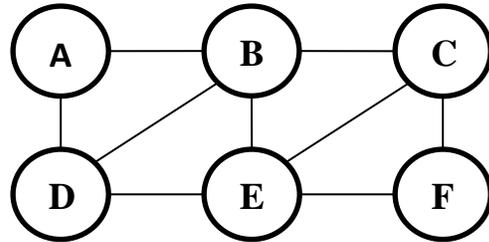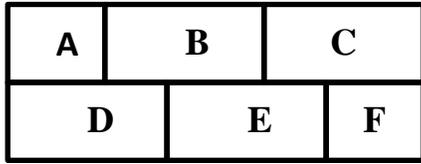
3a. (1 pt) FORWARD CHECKING. Consider the partial assignment below. Variable B has been assigned value R as shown. Cross out all values that would be eliminated by Forward Checking (FC) after the assignment to variable B.

A	B	C	D	E	F
X̶ G B	R	X̶ G B	X̶ G B	X̶ G B	R G B

3b. (1 pt) ARC CONSISTENCY. Consider the partial assignment below. Variables A and B have been assigned values as shown. Cross out all other values that would be eliminated by Arc Consistency (AC, also called AC-3 in your book).

A	B	C	D	E	F
R	G	X̶ X̶ B	X̶ X̶ B	R X̶ X̶	X̶ G X̶

3c. (1 pt) MINIMUM-REMAINING-VALUES HEURISTIC. Consider the partial assignment below. Variable A is already assigned value R, and Arc Consistency is already done. List all unassigned variables that might possibly be selected by the Minimum-Remaining-Values (MRV) Heuristic:

      B, D   .

A	B	C	D	E	F
R	G B	R G B	G B	R G B	R G B

3d. (1 pt) DEGREE HEURISTIC. Consider the partial assignment below. (It is the same assignment as in problem 3c above.) List all unassigned variables that might possibly be selected by the Degree Heuristic: **E**   .

A	B	C	D	E	F
R	G B	R G B	G B	R G B	R G B

3e. (1 pt) MIN-CONFLICTS HEURISTIC. Consider the complete but inconsistent assignment below. E has just now been selected to be assigned a new value. List all new values that might be chosen below for E by the Min-Conflicts Heuristic?   **R, B**   .

A	B	C	D	E	F
R	G	B	R	?	G

4. (10 pts total, 2 pts each) STATE-SPACE SEARCH. Execute Tree Search through this graph (do not remember visited nodes, so repeated nodes are possible). It is not a tree, but pretend you don't know that. Step costs are given next to each arc, and heuristic values are given next to each node (as h=x). The successors of each node are indicated by the arrows out of that node. **(Note: C is a successor of itself).**

For each search strategy below, indicate the order in which nodes are expanded (i.e., to expand a node means that its children are generated), ending with the goal node that is found.
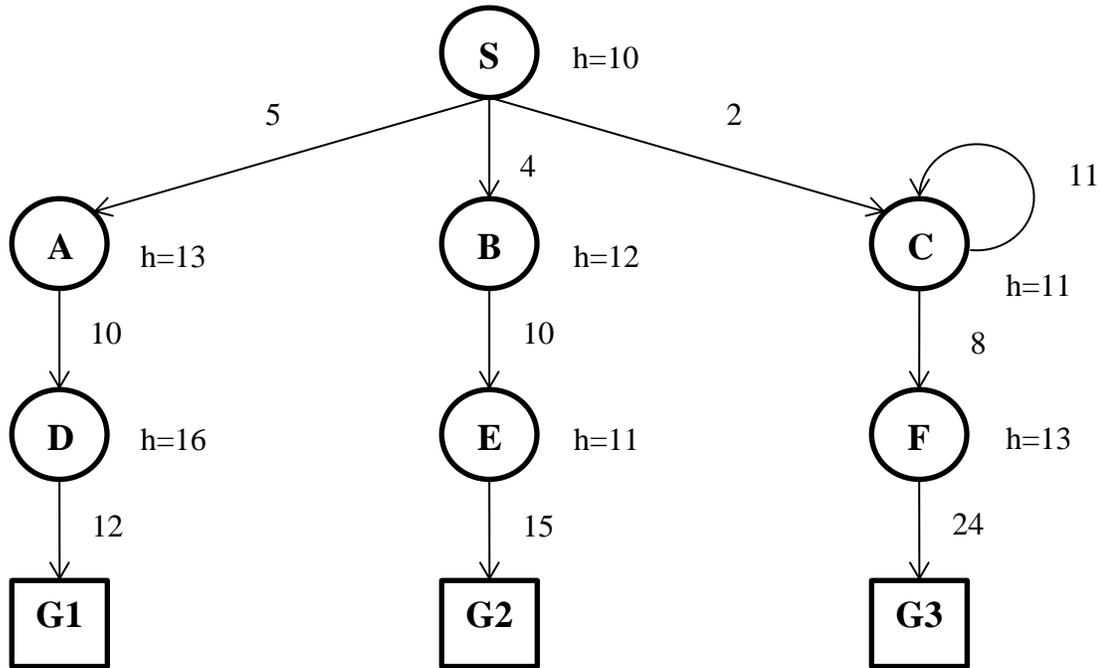
4.a. (2 pts, -1 for each wrong answer, but not negative) UNIFORM COST SEARCH.

S C B A F C E D F C G1   *See Section 3.4.2 and Fig. 3.14.*   **UCS does goaltest when node is popped off queue.**

4.b. (2 pts, -1 for each wrong answer, but not negative) GREEDY BEST-FIRST SEARCH.

S C C C C C C C C C C etc.   *See Section 3.5.1 and Fig. 3.23.*   **C always has lower h(=11) than any other node on queue.**

4.c (2 pts, -1 for each wrong answer, but not negative) ITERATIVE DEEPENING SEARCH.

S S A B C S A D G1   *See Sections 3.4.4-5 and Figs. 3.18-19.*   **IDS does the Goal-test before the child is pushed onto the queue. The goal is found when D is expanded.**

4.d. (2 pts, -1 for each wrong answer, but not negative) A* SEARCH.

S C B A F C E G2   *See Section 3.5.2 and Figs. 3.24-25.*   **A* does goaltest when node is popped off queue.**

4.e. (2 pts, -1 for each wrong answer, but not negative) OPTIMALITY.

Did Uniform Cost Search find the optimal goal? Yes   Why or why not? Step costs are ≥ ε > 0

Did A* Search find the optimal goal? No   Why or why not? heuristic is not admissible (at D)

**** TURN PAGE OVER AND CONTINUE ON THE OTHER SIDE ****

5. (15 pts total, 1 pt each) AGENT/SEARCH CONCEPTS. For each of the following terms on the left, write in the letter corresponding to the best answer or the correct definition on the right. The first one is done for you as an example.

A	Agent	A	Perceives environment by sensors, acts by actuators
K	Percept	B	All states reachable from the initial state by a sequence of actions
P	Performance Measure	C	Guaranteed to find a solution if one is accessible
L	Rational Agent	D	Process of removing detail from a representation
B	State Space	E	Maximum number of successors of any node
I	Search Node	F	Set of all leaf nodes available for expansion at any given time
N	Link between nodes	G	Estimates cost of cheapest path from current state to goal state
J	Path	H	Guaranteed to find lowest cost among all accessible solutions
D	Abstraction	I	Represents a state in the state space
H	Optimal Search	J	Sequence of states connected by a sequence of actions
C	Complete Search	K	Agent's perceptual inputs at any given instant
M	Expand a state	L	Agent that acts to maximize its expected performance measure
F	Frontier	M	Apply each legal action to a state, generating a new set of states
O	Search Strategy	N	Represents an action in the state space
E	Branching Factor	O	How a search algorithm chooses which node to expand next
G	Heuristic Function	P	Evaluates any given sequence of environment states for utility

6. (5 pts total, -2 pts each wrong answer, but not negative) TASK ENVIRONMENT. A task environment is a set of four things, with acronym PEAS. Fill in the blanks with the PEAS components.

Performance (measure)   Environment   Actuators   Sensors

See Section 2.3.1

7. (10 pts total, -1 each wrong answer, but not negative) SEARCH PROPERTIES.
Fill in the values of the four evaluation criteria for each search strategy shown. Assume a tree search where b is the finite branching factor; d is the depth to the shallowest goal node; m is the maximum depth of the search tree; C* is the cost of the optimal solution; step costs are identical and equal to some positive ε; and in Bidirectional search both directions use breadth-first search.

Note that these conditions satisfy all of the footnotes of Fig. 3.21 in your book.

See Figure 3.21

Criterion	Complete?	Time complexity	Space complexity	Optimal?
Breadth-First	Yes	$O(b^d)$	$O(b^d)$	Yes
Uniform-Cost	Yes	$O(b^{(1+floor(C*/\varepsilon))})$ $O(b^{(d+1)})$ also OK	$O(b^{(1+floor(C*/\varepsilon))})$ $O(b^{(d+1)})$ also OK	Yes
Depth-First	No	$O(b^m)$	$O(bm)$	No
Iterative Deepening	Yes	$O(b^d)$	$O(bd)$	Yes
Bidirectional (if applicable)	Yes	$O(b^{(d/2)})$	$O(b^{(d/2)})$	Yes

8. (10 pts total, -1 each error, but not negative) STATE SPACE SEARCH.

A man is traveling to market with a fox, a goose, and a bag of oats. He comes to a river. The only way across the river is a boat that can hold the man and exactly one of the fox, goose or bag of oats. The fox will eat the goose if left alone with it, and the goose will eat the oats if left alone with it. **How can the man get all his possessions safely across the river?**

We will call states where something gets eaten "forbidden" and consider them as part of the state space, but from which there is no return; i.e., there is no way to go to any other state (including back to the previous state) from such a forbidden state. All other states are "allowed."

One way to represent this problem is as a state vector with components (M,F,G,O), where M=man, F=fox, G=goose, and O=oats are binary variables that are 0 if the man/fox/goose/oats are on the **near** bank, and 1 if on the **far** bank. Note that the boat is always on the same side of the river as the man. Thus the start state is (0,0,0,0) and the single goal state is (1,1,1,1).

8.a. (1 pt) How many total states (both allowed and forbidden) in this state space?

_____ $2^4 = 16$ _____.

8.b. (1 pt) What is the maximum branching factor counting BOTH allowed and forbidden states as children?

_____ 4 _____.

8.c. (8 pts, -1 for each error, but not negative) Draw the state space considering only ALLOWED states. Represent an allowed state as a circle enclosing its binary vector (M,F,G,O). Connect states that are possible successors of each other. The start and goal states are shown. The first one is done for you as an example.
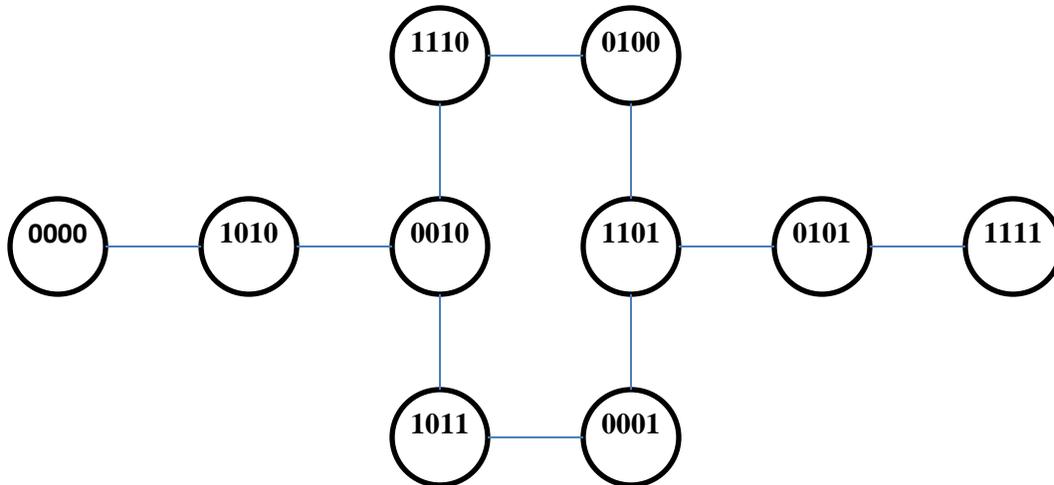
**** TURN PAGE OVER AND CONTINUE ON THE OTHER SIDE ****

9. (10 pts total, -2 each error, but not negative) OPTIMALITY OF A* SEARCH. You know that A* tree search with an admissible heuristic is an optimal search strategy. Recall that:

$g(n)$ = true path cost so far
$h(n)$ = estimated optimal cost from n to goal $\leq$ true optimal cost
$f(n) = g(n) + h(n)$ = estimated total optimal path cost $\leq$ true total optimal cost

The following is a proof that A* tree search (queue sorted by $f(n)$) is optimal if the heuristic is admissible. The lines of the proof have been labeled A through G. Unfortunately, they have been scrambled.

Let *ng* be the first goal node popped off the queue. Let *no* be any other node still on the queue. We wish to prove that the path through *no* can never be extended to a path to any goal node that costs less than the path to *ng* that we just found.

A: true total cost of path to *ng*
F: $= g(ng)$ // because *ng* represents a complete path
D: $= f(ng)$ // by definition of *f*, with $h(ng) = 0$ because *ng* is a goal node
B: $\leq f(no)$ // because queue is sorted by *f* and *ng* came off queue first
E: $= g(no) + h(no)$ // by definition of *f*
C: $\leq g(no)$ + true optimal cost to goal from *no*   // because *h* is admissible
G: = true total optimal path cost through *no*

Fill in the blanks with the letters B, C, D, E, F, and G to prove that the true total cost of the path to ng is less than or equal to the true total optimal path cost through *no*. The first and last letters, A and G, have been done for you as an example.

    A   F   D   B   E   C   G   .

10. (4 pts total, 1 pt each) RESOLUTION. Apply resolution to each of the following pairs of clauses, then simplify. Write your answer in Conjunctive Normal Form (CNF).

10.a. (1 pt) (A B ¬C D) (A ¬D E F).   (A B ¬C E F)   .

10.b. (1 pt) (A B ¬C D) (¬A).   (B ¬C D)   .

10.c. (1 pt) (¬C) (C).   ( )   **"FALSE" is OK**   .

10.d. (1 pt) (A B ¬C D) (A C ¬D E F).   (A B ¬C C E F) **also OK** (A B D ¬D E F)   .
                                                "TRUE" is OK

10

11. (10 pts total, 1 pt each) PROVE THAT THE UNICORN IS MAGICAL.

If the unicorn is mythical, then it is immortal, but if it is not mythical, then it is a mortal mammal. If the unicorn is either immortal or a mammal, then it is horned. The unicorn is magical if it is horned.

Use these propositional variables:

Y = unicorn is m**Y**thical	**R** = unicorn is mo**R**tal	**M** = unicorn is a ma**M**mal
H = unicorn is **H**orned	**G** = unicorn is ma**G**ical	

See Section 7.5.2

11.a. (5 pts total, pt each) Convert the English into propositional logic implicative form and conjunctive normal form (CNF). The first one is done for you as an example. (Note: "immortal" means "not mortal.")

11.a.1. If the unicorn is mYthical, then it is not moRtal.

S1: Implicative _____ $Y \Rightarrow \neg R$ _____ . CNF _____ $(\neg Y \lor \neg R)$ _____

It is OK not to put the parentheses around the CNF clauses.

11.a.2. (1 pt) If the unicorn is not mYthical, then it is moRtal.

S2: Implicative _____ $\neg Y \Rightarrow R$ _____ . CNF _____ $(Y \lor R)$ _____

11.a.3. (1 pt) If the unicorn is not mYthical, then it is a maMmal.

S3: Implicative _____ $\neg Y \Rightarrow M$ _____ . CNF _____ $(Y \lor M)$ _____

It is OK if you did not put in the disjunction symbol "∨".

11.a.4. (1 pt) If the unicorn is not moRtal, then it is Horned.

S4: Implicative _____ $\neg R \Rightarrow H$ _____ . CNF _____ $(R \lor H)$ _____ .

11.a.5. (1 pt) If the unicorn is a maMmal, then it is Horned.

S5: Implicative _____ $M \Rightarrow H$ _____ . CNF _____ $(\neg M \lor H)$ _____ .

11.a.6. (1 pt) The unicorn is maGical if it is Horned. (= "If the unicorn is Horned, then it is maGical.")

S6: Implicative _____ $H \Rightarrow G$ _____ . CNF _____ $(\neg H \lor G)$ _____ .

**** CONTINUE ON THE NEXT PAGE ****

11.b. (5 pts total, pt each) Resolution Theorem Proving. Use the conjunctive normal form (CNF) expressions from **11.a** above to prove that the unicorn is magical. The first and last steps are done for you. **Express your answers in CNF.** | See Section 7.5.2 |

11.b.1. The negated goal is S7.

S7: _____ (¬ G) _____ .

Here you will get full credit if you do the resolution steps correctly based on your answers for the two resolved sentences (even if those sentences were not correct). I.e., this question asks only that you do the resolution step correctly, regardless of content.

11.b.2. (1 pt) Resolve S6 and S7 to give S8.

S8: _____ (¬ H) _____ .

11.b.3. (1 pt) Resolve S5 and S8 to give S9.

S9: _____ (¬ M) _____ .

11.b.4. (1 pt) Resolve S4 and S8 to give S10.

S10: _____ (R) _____ .

It is OK not to put the parentheses around the CNF clauses.

11.b.5. (1 pt) Resolve S3 and S9 to give S11.

S11: _____ (Y) _____ .

11.b.6. (1 pt) Resolve S1 and S11 to give S12.

S12: _____ (¬ R) _____ .

11.b.7. Resolve S10 and S12 to give the empty clause, thus proving the goal sentence is true.

S13: _____ ( ) _____ .

12. (4 pts total, -1 each error, but not negative) CONJUNCTIVE NORMAL FORM (CNF). Convert the following logical sentence to Conjunctive Normal Form. **Show your work.**

$$B \Leftrightarrow (P \vee Q)$$ | See Section 7.5.2 |

1. Eliminate $\Leftrightarrow$, replacing $\alpha \Leftrightarrow \beta$ with $(\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha)$.
   $(B \Rightarrow (P \vee Q)) \wedge ((P \vee Q) \Rightarrow B)$
2. Eliminate $\Rightarrow$, replacing $\alpha \Rightarrow \beta$ with $\neg \alpha \vee \beta$.
   $(\neg B \vee P \vee Q) \wedge (\neg(P \vee Q) \vee B)$
3. Move $\neg$ inwards using de Morgan's rules:
   $(\neg B \vee P \vee Q) \wedge ((\neg P \wedge \neg Q) \vee B)$
4. Apply distributive law ($\wedge$ over $\vee$) and flatten:
   $(\neg B \vee P \vee Q) \wedge (\neg P \vee B) \wedge (\neg Q \vee B)$
5. write each clause (disjunct) as a sentence in KB:
   $(\neg B \vee P \vee Q)$
   $(\neg P \vee B)$
   $(\neg Q \vee B)$