CS536, Spring 2022
Midterm Exam #1
SOLUTIONS: DO NOT DISTRIBUTE

Name _____

IIT Email _____

Important notes:

- This exam has 14 pages. Make sure you have them all.
- You have 75 minutes to complete the exam. We suggest looking through the questions first to see where to focus your time.
- Use only **blue or black pen** to complete this exam. If you don't have one, ask.
- Write your answers in the space provided. If you need more space for answers or scrap, you can use the back of the page, but clearly mark where your answers are.
- The last 3 pages of this exam are reference material. You may (carefully) tear them off and use them durng the exam. We do not need to collect these pages.
- You are permitted to refer to one double-sided 8.5" × 11" sheet of notes. We will collect your sheet of notes at the end of the exam, so if you want it back, please make sure your name is on it. **No other outside aids (including electronics) or notes are permitted.**
- Sign the statement below:
  I have not used any unauthorized resources or received or given help during this exam.

Signed_____ Date_____

Question	1	2	3	4	5	**Total**
Points	20	21	12	22	25	100
Score						
Grader						

1 True and False (20 points)

1. T Verification can find bugs that testing might never find.

2. F $(\{x = 1\}[y \mapsto 2])(y) = \bot_e$.

3. F P is a contradiction if and only if $F \Rightarrow P$.

4. T $\{F\}\ s\ \{F\}$ is valid for any s.

5. T If $[\ T\ ]\ s\ [\ T\ ]$ is valid, then s is guaranteed to terminate and not have a runtime error.

6. F If $\{p\}\ s\ \{q\}$ is not valid, then the program s must have a bug that needs to be fixed.

7. T If $\sigma \not\Vdash \{p\}\ s\ \{q\}$, then $\bot \notin M(s, \sigma)$

8. F $\bot_d \in M(\mathsf{while}\ x \neq y\ \{x := x + \overline{1}\}, \{x = 0, y = 5\})$

9. F $[\ y = k^2\ ]\ x := sqrt(y)\ [\ x = k\ ]$ is invalid because k doesn't appear in the program.

10. F $\forall x.\forall y.x > y$ is a contingency.

2 Proving predicates (21 points)

(a) (9 points) For each of the following predicates, circle the answer that describes how we could prove that the predicate is valid.

i) $\vDash \forall x \in \mathbb{Z}.\exists y \in \mathbb{Z}.x > y$

A) Give an x and y such that $x > y$.

B) Show that, for any given x, we can give a y such that $x > y$.

C) Show that, for any given y, we can give an x such that $x > y$.

D) Give an x and show that, for any y, $x > y$.

B

ii) $\vDash \exists x \in \mathbb{Z}.\forall y \in \mathbb{Z}.y \times x = 0$

A) Give an x and y such that $y \times x = 0$.

B) Show that, for any given x, we can give a y such that $y \times x = 0$.

C) Give an x and show that, for any y, $y \times x = 0$.

C

iii) $\nvDash \exists x \in \mathbb{Z}.\forall y \in \mathbb{Z}.x = y$

A) Give an x and y such that $x \neq y$.

B) Show that, for any given x, we can give a y such that $x \neq y$.

C) Show that, for any given y, we can give an x such that $x \neq y$.

D) Give an x and show that, for any y, $x \neq y$.

B

(b) (12 points) For each of the following predicates, give a state σ such that the predicate is satisfied or explain briefly (1-3 sentences) why there is no state σ such that the predicate is satisfied.

i) $\sigma \vDash \forall y \in \mathbb{Z}.y < z$ There is none. No matter what z is, it is not true that all $y \in \mathbb{Z}$ are less than z.

ii) $\sigma \vDash \exists x \in \mathbb{Z}.x^2 = y.$ $\{y = 9\}$ (or any perfect square)

iii) $\sigma \vDash \forall x \in \mathbb{Z}.(x = -3 \wedge (\forall y \in \mathbb{Z}.\exists x \in \mathbb{Z}.y^2 = x))$ There is no state, as the predicate is unsatisfied for any $x \neq 3$, and we quantify over all x.

3 Truth Table (12 points)

(a) (10 points) Complete the truth table for the proposition

$$(P \to Q) \leftrightarrow (P \wedge \neg Q)$$

P	Q	$\neg Q$	$(P \to Q)$	$(P \wedge \neg Q)$	$(P \to Q) \leftrightarrow (P \wedge \neg Q)$
T	T	F	T	F	F
T	F	T	F	T	F
F	T	F	T	F	F
F	F	T	T	F	F

(b) (2 points) Is $(P \to Q) \leftrightarrow (P \wedge \neg Q)$ a tautology, contradiction or contingency?

Contradiction

4 Proof (22 points)

Prove the following logical implication, known as *Modus Tollens*.

$$(P \to Q) \wedge \neg Q \Rightarrow \neg P$$

Write a statement and a justification on each line, as you did on HW1. The justifications should be logical laws (See Appendix A) with a reference to the lines you're using. For example, if line 3 has the statement P and line 4 has the statement Q, you could justify $P \wedge Q$ with "Conjunction(3, 4)". If a justification uses only one line and it's the line immediately before, you can leave it out. You can leave lines left over; there may be more than you need.

1	$(P \to Q) \wedge \neg Q$	Assumption
2	$(\neg P \vee Q) \wedge \neg Q$	Definition of Conditional
3	$(\neg P \wedge \neg Q) \vee (Q \wedge \neg Q)$	Distributivity
4	$(\neg P \wedge \neg Q) \vee F$	Contradiction
5	$(\neg P \wedge \neg Q)$	Identity
6	$\neg P$	Simplify

5 Multiplying Numbers (25 points)

Consider the following statement, which we'll call s. The goal of this code is to multiply x and y by performing repeated additions.

$$
\begin{array}{l}
i := \overline{0}; \\
n := \overline{0}; \\
\mathsf{while}(i < y)\{ \\
\quad n := n + x; \\
\quad i := i + \overline{1} \\
\}
\end{array}
$$

(a) (4 points) For each of the following configurations we might come across while evaluating s, fill in the blanks in the configuration we would step to *in one step*. When writing states, write them as sets of variables and values, e.g. $\{x = 1, y = 2\}$, not as state updates (e.g. $\{x = 2, y = 1\}[x \mapsto 1]$).

i)

$$\langle s, \{x = 1, y = 1\}\rangle$$

$$\rightarrow \quad \langle \mathsf{skip}; n := \overline{0}; \mathsf{while}\ i < y\ \{n := n + x; i := i + \overline{1}\}, \{x = 1, y = 1, i = 0\}\rangle$$

ii)

$$\langle n := n + x; i := i + \overline{1}, \{x = 3, y = 3, i = 2, n = 6\}\rangle$$

$$\rightarrow \quad \langle \mathsf{skip}; i := i + \overline{1}; \mathsf{while}\ i < y\ \{n := n + x; i := i + \overline{1}\}, \{x = 3, y = 3, i = 2, n = 9\}\rangle$$

Note: $\mathsf{skip}; i := i + \overline{1}$ also OK because of mistake in the exam

[QUESTION CONTINUES ON NEXT PAGE]

(b) (6 points) Use the big-step semantics to figure out the final state we'll reach by running s in each of the initial states below. For example, $M(s, \{x = 1, y = 0\}) = \{\{x = 1, y = 0, n = 0, i = 0\}\}$ and $M(s, \{x = 1, y = 1\}) = \{\{x = 1, y = 1, n = 1, i = 1\}\}$.

Use $\bot_d$ for divergence and $\bot_e$ for errors; do not use just $\bot$ (with no subscript). You don't need to show work.

i) $M(s, \{x = 3, y = 2, i = 0, n = 0\})$ {{x = 3, y = 2, i = 2, n = 6}}

ii) $M(s, \{x = 3, y = 2, i = 0, n = 1\})$ {{x = 3, y = 2, i = 2, n = 6}}

iii) $M(s, \{x = 3, y = -1, i = 0, n = 0\})$ {{x = 3, y = -1, i = 0, n = 0}}

(c) (4 points) Fill in the precondition below so the triple is satisfied. Your precondition should be satisfied for infinitely many values of y, e.g., don't simply write "F" (which would be satisfied in 0 states) or $y = 1$ (which would be satisfied by only one value of y).

$$\vDash \{y \geq 0\}\ s\ \{n = x * y \wedge i = y\}$$

[QUESTION CONTINUES ON NEXT PAGE]

(d) (4 points) Fix the program (you can rewrite the whole program or just the changed part, but make it clear which lines you're changing) so that the following triple is satisfied **or** explain, in 2-5 sentences, why this is not possible.

$$\vDash [\ T\ ]\ s\ [\ n = x * y \wedge i = y\ ]$$

Some possibilities:

$y0 := y;$
$y := y < 0\ ?\ -y : y;$
$i := \overline{0};$
$n := \overline{0};$
$\mathsf{while}(i < y)\{$
  $n := n + x;$
  $i := i + \overline{1}$
$\};$
$\mathsf{if}(y \neq y0)\{$
  $n := -n;$
  $y := -y$
$\}$

$y := y < 0\ ?\ -y : y;$
$i := \overline{0};$
$n := \overline{0};$
$\mathsf{while}(i < y)\{$
  $n := n + x;$
  $i := i + \overline{1}$
$\}$

$n := 0;$
$x := 0;$
$y := 0;$
$i := 0$

(e) (7 points) Suppose we now want to multiply not just two numbers, but we want to calculate $a[j] \times b[j]$ for each element of a and b and put the result in $c[j]$. We'll use the code below, where s refers to the code at the beginning of this question. The code loads x from a and y from b, to set things up to run s, and then writes n (the result from running s) into c. We'll refer to the code below as t.

$$
\begin{array}{l}
j := \overline{0}; \\
\mathsf{while}(j < \mathit{size}(a))\{ \\
\quad x := a[j]; \\
\quad y := b[j]; \\
\quad s; \\
\quad c[j] := n; \\
\}
\end{array}
$$

i) (2 points) What is $M(t, \{a = [1; 2; 3], b = [4; -1]\})$? As before, use $\perp_d$ for divergence and $\perp_e$ for errors; do not use just $\perp$ (with no subscript). You don't need to show work.

$\{\perp_e\}$

One point for $\{\perp_d\}$ because increment wasn't included

[QUESTION CONTINUES ON NEXT PAGE]

ii) (5 points) Fill in the precondition below so the triple is satisfied. Similar to the previous part, your precondition should be satisfied for infinitely many arrays a and b. Recall that $|a|$ is the size of a and $\forall i \in [m, n]$ quantifies over all integer values of i between m and n (inclusive).

$$\vDash \{\forall i \in [0, |a| - 1].b[i] \geq 0\}\ t\ \{\forall i \in [0, |a| - 1].c[i] = a[i] * b[i]\}$$

A Logic Laws

Name	Description
Simplify	$p \wedge q \Rightarrow p, q$
Modus Ponens	$(p \rightarrow q), p \Rightarrow q$
Conjunction	$p, q \Rightarrow p \wedge q$
Disjunction	$p \Rightarrow p \vee q, q \vee p$
Definition of Conditional	$p \rightarrow q \Leftrightarrow \neg p \vee q$
Definition of Biconditional	$p \leftrightarrow q \Leftrightarrow (p \rightarrow q) \wedge (q \rightarrow p)$
Law of the Excluded Middle (LEM)	$p \vee \neg p \Leftrightarrow T$
Double Negation Elimination (DNE)	$p \Leftrightarrow \neg\neg p$
Contradiction	$p \wedge \neg p \Leftrightarrow F$
Identity	$p \wedge T \Rightarrow p$, $p \vee F \Rightarrow p$
DeMorgan's Laws	$\neg(p \wedge q) \Leftrightarrow \neg p \vee \neg q$ $\neg(p \vee q) \Leftrightarrow \neg p \wedge \neg q$ $\neg(\forall x.p(x)) \Leftrightarrow \exists x.\neg p(x)$ $\neg(\exists x.p(x)) \Leftrightarrow \forall x.\neg p(x)$
Distributivity	$(p \wedge q) \vee r \Leftrightarrow (p \vee r) \wedge (q \vee r)$ $(p \vee q) \wedge r \Leftrightarrow (p \wedge r) \vee (q \wedge r)$
Commutativity	$p \wedge q \Leftrightarrow q \wedge p$, $p \vee q \Leftrightarrow q \vee p$
Associativity	$(p \wedge q) \wedge r \Leftrightarrow p \wedge (q \wedge r)$, $(p \vee q) \vee r \Leftrightarrow p \vee (q \vee r)$
Idempotency	$p \wedge p \Leftrightarrow p$, $p \vee p \Leftrightarrow p$
Domination	$p \vee T \Leftrightarrow T$, $p \wedge F \Leftrightarrow F$

B Language Syntax and Semantics

Expression and Statement Syntax

$$
\begin{array}{rcl}
e & ::= & \overline{n} \mid \mathsf{true} \mid \mathsf{false} \mid x \mid a[e] \mid e \; op \; e \mid e \; ? \; e : e \mid size(a) \\
s & ::= & \mathsf{skip} \mid s; s \mid x := e \mid a[e] := e \mid \mathsf{if} \; e \; \mathsf{then} \; \{s\} \; \mathsf{else} \; \{s\} \mid \mathsf{while} \; e \; \{s\}
\end{array}
$$

Expression Semantics

$$
\begin{array}{rcll}
\sigma(\overline{n}) & = & n & \\
\sigma(\mathsf{true}) & = & T & \\
\sigma(\mathsf{false}) & = & F & \\
\sigma(x) & = & \sigma(x) & \\
\sigma(a[e]) & = & (\sigma(a))[\sigma(e)] & \sigma(e) \neq \bot_e \wedge 0 \leq \sigma(e) < |\sigma(a)| \\
\sigma(a[e]) & = & \bot_e & \text{otherwise} \\
\sigma(e_1 \; op \; e_2) & = & \sigma(e_1) \; op \; \sigma(e_2) & \sigma(e_1) \neq \bot_e \neq \sigma(e_2) \\
\sigma(e_1 \; op \; e_2) & = & \bot_e & \sigma(e_1) = \bot_e \vee \sigma(e_2) = \bot_e \\
\sigma(e_1 \; ? \; e_2 : e_3) & = & \sigma(e_2) & \sigma(e_1) = T \\
\sigma(e_1 \; ? \; e_2 : e_3) & = & \sigma(e_3) & \sigma(e_1) = F \\
\sigma(e_1 \; ? \; e_2 : e_3) & = & \bot_e & \sigma(e_1) = \bot_e \\
\sigma(size(a)) & = & |\sigma(a)| &
\end{array}
$$

Statement Semantics - Small-step

$$\frac{\langle s_1, \sigma\rangle \to \langle s_1', \sigma\rangle}{\langle s_1; s_2, \sigma\rangle \to \langle s_1'; s_2, \sigma\rangle} \qquad \frac{\langle s_1, \sigma\rangle \to \langle \mathsf{skip}, \bot_e\rangle}{\langle s_1; s_2, \sigma\rangle \to \langle \mathsf{skip}, \bot_e\rangle} \qquad \frac{}{\langle \mathsf{skip}; s, \sigma\rangle \to \langle s, \sigma\rangle}$$

$$\frac{\sigma(e) \neq \bot_e}{\langle x := e, \sigma\rangle \to \langle \mathsf{skip}, \sigma[x \mapsto \sigma(e)]\rangle} \qquad \frac{\sigma(e) = \bot_e}{\langle x := e, \sigma\rangle \to \langle \mathsf{skip}, \bot_e\rangle}$$

$$\frac{\sigma(e_1) \neq \bot_e \qquad \sigma(e_2) \neq \bot_e \qquad 0 \leq \sigma(e_1) < |\sigma(a)|}{\langle a[e_1] := e_2, \sigma\rangle \to \langle \mathsf{skip}, \sigma[a[\sigma(e_1)] \mapsto \sigma(e_2)]\rangle} \qquad \frac{\sigma(e_1) = \bot_e \vee \sigma(e_2) = \bot_e}{\langle a[e_1] := e_2, \sigma\rangle \to \langle \mathsf{skip}, \bot_e\rangle}$$

$$\frac{\sigma(e_1) \geq |\sigma(a)| \vee \sigma(e_1) < 0}{\langle a[e_1] := e_2, \sigma\rangle \to \langle \mathsf{skip}, \bot_e\rangle} \qquad \frac{\sigma(e) = T}{\langle \mathsf{if} \; e \; \mathsf{then} \; \{s_1\} \; \mathsf{else} \; \{s_2\}, \sigma\rangle \to \langle s_1, \sigma\rangle}$$

$$\frac{\sigma(e) = F}{\langle \mathsf{if} \; e \; \mathsf{then} \; \{s_1\} \; \mathsf{else} \; \{s_2\}, \sigma\rangle \to \langle s_2, \sigma\rangle} \qquad \frac{\sigma(e) = \bot_e}{\langle \mathsf{if} \; e \; \mathsf{then} \; \{s_1\} \; \mathsf{else} \; \{s_2\}, \sigma\rangle \to \langle \mathsf{skip}, \bot_e\rangle}$$

$$\frac{}{\langle \mathsf{while} \; e \; \{s\}, \sigma\rangle \to \langle \mathsf{if} \; e \; \mathsf{then} \; \{s; \mathsf{while} \; e \; \{s\}\} \; \mathsf{else} \; \{\mathsf{skip}\}, \sigma\rangle}$$

Statement Semantics - Big-step

$$
\begin{array}{rcll}
M(\mathsf{skip}, \sigma) & = & \{\sigma\} & \\
M(s_1; s_2, \sigma) & = & \bigcup_{\sigma' \in M(s_1, \sigma)} M(s_2, \sigma') & \\
M(x := e, \sigma) & = & \{\sigma[x \mapsto \sigma(e)]\} & \sigma(e) \neq \bot_e \\
M(x := e, \sigma) & = & \{\bot_e\} & \sigma(e) = \bot_e \\
M(a[e_1] := e_2, \sigma) & = & \{\sigma[a[\sigma(e_1)] \mapsto \sigma(e_2)]\} & \sigma(e_1) \neq \bot_e \wedge \sigma(e_2) \neq \bot_e \wedge 0 \leq \sigma(e_1) < |\sigma(a)| \\
M(a[e_1] := e_2, \sigma) & = & \{\bot_e\} & \text{otherwise} \\
M(\mathsf{if}\ e\ \mathsf{then}\ \{s_1\}\ \mathsf{else}\ \{s_2\}, \sigma) & = & M(s_1, \sigma) & \sigma(e) = T \\
M(\mathsf{if}\ e\ \mathsf{then}\ \{s_1\}\ \mathsf{else}\ \{s_2\}, \sigma) & = & M(s_2, \sigma) & \sigma(e) = F \\
M(\mathsf{if}\ e\ \mathsf{then}\ \{s_1\}\ \mathsf{else}\ \{s_2\}, \sigma) & = & \{\bot_e\} & \sigma(e) = \bot_e \\
M(\mathsf{while}\ e\ \{s\}, \sigma) & = & \Sigma_k & \Sigma_k \text{ is the lowest } k \text{ such that if} \\
 & & & \sigma \in \Sigma_k \text{ , then } \sigma(e) = F \\
M(\mathsf{while}\ e\ \{s\}, \sigma) & = & \{\bot_d\} & \text{no such } k \text{ exists}
\end{array}
$$

where

$$
\begin{array}{rcl}
\Sigma_0 & = & \{\sigma\} \\
\Sigma_k + 1 & = & \bigcup_{\sigma \in \Sigma_k} M(s, \sigma)
\end{array}
$$