QUIZ 3: 60 Minutes

Last Name: ______________________

First Name: ______________________

RIN: ______________________

Section: ______________________

Answer **ALL** questions.
NO COLLABORATION or electronic devices. Any violations result in an F.
NO questions allowed during the test. Interpret and do the best you can.

GOOD LUCK!

Circle at most one answer per question.
10 points for each correct answer

You **MUST** show **CORRECT** work to get full credit.

When in doubt, TINKER.

Total
200

1. A function f maps $\{a, b, c, d\}$ to $\{1, 2, 3\}$ as follows: $f(a) = 1, f(b) = 2, f(c) = 1, f(d) = 2$.

A f is injective (1-to-1) but not bijective.

B f is surjective (onto) but not bijective.

C f is bijective (1-to-1 and onto).

D f is neither injective nor surjective.

E f is not a valid function.

2. A set $\mathcal{S}$ contains all the distinct functions which map $\mathbb{N}$ to $\{0\}$. What is the cardinality (size) of $\mathcal{S}$?

A 0.

B 1.

C Bigger than 1 but finite.

D The same as $|\mathbb{N}|$.

E Strictly larger than $|\mathbb{N}|$.

3. A set $\mathcal{S}$ contains all the distinct functions which map $\mathbb{N}$ to $\{2, 3, 4\}$. What is the cardinality (size) of $\mathcal{S}$?

A 0.

B 1.

C Bigger than 1 but finite.

D The same as $|\mathbb{N}|$.

E Larger than $|\mathbb{N}|$.

4. What is the cardinality (size) of the set containing all distinct python programs of finite length?

A 0.

B 1.

C Bigger than 1 but finite.

D The same as $|\mathbb{N}|$.

E Larger than $|\mathbb{N}|$.

5. Which set is *not* countable, i.e., has a cardinality strictly larger than $|\mathbb{N}|$?

A $\mathbb{Q}$, the rational numbers.

B All distinct finite binary strings.

C The set of all possible Turing Machines.

D The set containing all distinct functions that map $\{0, 1\}$ to $\mathbb{N}$.

E They are all countable.

6. What is a computing problem?

$\boxed{\text{A}}$ A person who knows how to write a program in python.

$\boxed{\text{B}}$ A machine that transitions between states.

$\boxed{\text{C}}$ A rule for deciding if a string belongs to a set.

$\boxed{\text{D}}$ Any set of finite binary strings.

$\boxed{\text{E}}$ A Turing Machine.

7. $\mathcal{L}$ is a computing problem. What can we say about the cardinality (size) of $\mathcal{L}$?

$\boxed{\text{A}}$ $\mathcal{L}$ must have finite cardinality.

$\boxed{\text{B}}$ $\mathcal{L}$ must have infinite cardinality.

$\boxed{\text{C}}$ $\mathcal{L}$ must be countable.

$\boxed{\text{D}}$ $\mathcal{L}$ must be uncountable.

$\boxed{\text{E}}$ None of the above.

8. $\mathcal{L}_1$ and $\mathcal{L}_2$ are computing problems. Which of the following is *not* a computing problem?

$\boxed{\text{A}}$ $\mathcal{L}_1 \bullet \mathcal{L}_2$.

$\boxed{\text{B}}$ $\mathcal{L}_1^*$.

$\boxed{\text{C}}$ $\mathcal{L}_1 \cap \mathcal{L}_2^*$.

$\boxed{\text{D}}$ $\mathcal{L}_1^* \cup \overline{\mathcal{L}_2^*}$.

$\boxed{\text{E}}$ They are all computing problems

9. Which of the following strings is in the language described by the regular expression $\{0, 11\}^*$?

$\boxed{\text{A}}$ 011111.

$\boxed{\text{B}}$ 010010.

$\boxed{\text{C}}$ 100100.

$\boxed{\text{D}}$ 110011.

$\boxed{\text{E}}$ None of the strings above are in the language.

10. Which computing problem *cannot* be solved by a DFA (deterministic finite automata)?

$\boxed{\text{A}}$ $\mathcal{L} = \{\text{strings with at least one 1}\}$.

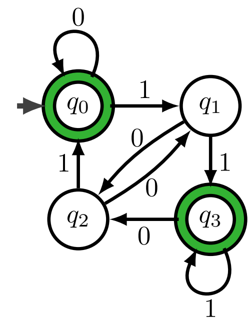
$\boxed{\text{B}}$ $\mathcal{L} = \{(01)^{\bullet n} \mid n \geq 0\}$.

$\boxed{\text{C}}$ $\mathcal{L} = \{\text{strings that end with 101}\}$.

$\boxed{\text{D}}$ $\mathcal{L} = \{1^{\bullet n} w | n \geq 1 \text{ and } w \text{ has } n \text{ or more 1s}\}$.

$\boxed{\text{E}}$ Each problem can be solved by a DFA.

11. For the DFA on the right, what sequence of states are visited for the input 0110.



A q_2 q_0 q_1 q_3 q_2.

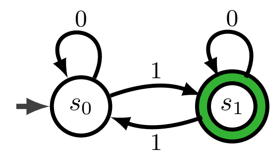
B q_0 q_0 q_0 q_0 q_0.

C q_0 q_1 q_3 q_2.

D q_0 q_0 q_1 q_2 q_3.

E q_0 q_0 q_1 q_3 q_2.

12. How many 4-bit strings are in the YES-set of the DFA on the right. Accept states are double circles (only s_1 in this case).



A 6.

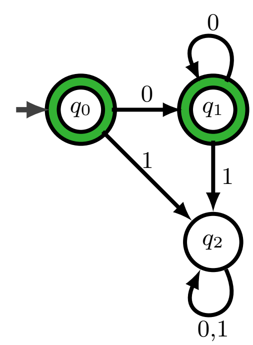
B 7.

C 8.

D 9.

E 10.

13. Which is *not* a description of the computing problem solved by the DFA on the right? Accept states are double circles (q_0, q_1 in this case).



A $\{0\}^*$.

B $\{0^{\bullet n} | n \geq 0\}$.

C All strings with no ones, including the empty string ε.

D Strings generated by the CFG: $S \to \varepsilon | 0S$.

E They all describe the computing problem solved by the DFA.

14. Which string *cannot* be generated by the CFG: $S \to 0|1|0S0|1S1$.

A 01110

B 10001

C 0110

D 11111

E They can all be generated.

15. Describe the strings generated by the CFG on the right.

$$
\begin{aligned}
1 &: S \to A | B \\
2 &: A \to 0 | 0A \\
3 &: B \to 1 | 1A
\end{aligned}
$$

A Nonempty strings containing only 0s.

B Nonempty strings containing only 1s.

C All nonempty strings.

D Nonempty strings which contain either only 0s or only 1s.

E None of the above

16. Which is a true statement about the computing problem $\mathcal{L} = \{w\#w | w \in \{0,1\}^*\}$. (# is punctuation)

$\boxed{\text{A}}$ A DFA can solve $\mathcal{L}$. A Turing Machine can solve $\mathcal{L}$.

$\boxed{\text{B}}$ A DFA cannot solve $\mathcal{L}$. A Turing Machine cannot solve $\mathcal{L}$.

$\boxed{\text{C}}$ A DFA can solve $\mathcal{L}$. A Turing Machine cannot solve $\mathcal{L}$.

$\boxed{\text{D}}$ A DFA cannot solve $\mathcal{L}$. A Turing Machine can solve $\mathcal{L}$.

$\boxed{\text{E}}$ None of the above.

17. How do we know there are computing problems which Turing Machines cannot solve?

$\boxed{\text{A}}$ Because the Turing Machines are countable and the computing problems are countable.

$\boxed{\text{B}}$ Because the Turing Machines are uncountable and the computing problems are countable.

$\boxed{\text{C}}$ Because the Turing Machines are countable and the computing problems are uncountable.

$\boxed{\text{D}}$ Because the Turing Machines are uncountable and the computing problems are uncountable.

$\boxed{\text{E}}$ None of the above proves there are computing problems which Turing Machines cannot solve.

18. What is the difference between a decider D for a language $\mathcal{L}$ and a recognizer R for $\mathcal{L}$.

$\boxed{\text{A}}$ For $w \in \mathcal{L}$, D halts with YES but R may infinite loop.

$\boxed{\text{B}}$ For $w \in \mathcal{L}$, D halts with YES and R halts but may say NO.

$\boxed{\text{C}}$ For $w \notin \mathcal{L}$, D halts with NO but R *may* sometimes go into an infinite loop.

$\boxed{\text{D}}$ For $w \notin \mathcal{L}$, D halts with NO but R *must* always go into an infinite loop.

$\boxed{\text{E}}$ For $w \notin \mathcal{L}$, D halts with NO and R halts but may say YES.

19. What is the Ultimate Debugger which we discussed in class?

$\boxed{\text{A}}$ A program that solves Goldbach's conjecture.

$\boxed{\text{B}}$ A program that solves the twin-prime conjecture.

$\boxed{\text{C}}$ A program that determines if another program will compile under a C^{++} compiler.

$\boxed{\text{D}}$ A program that translates another program into binary machine-code.

$\boxed{\text{E}}$ A program that determines if another program when run will halt.

20. Which answer is a valid conclusion about the decidability of the language $\mathcal{L}_B$?

$\boxed{\text{A}}$ $\mathcal{L}_A$ is decidable. A decider for $\mathcal{L}_B$ can be converted to a decider for $\mathcal{L}_A$. So, $\mathcal{L}_B$ is decidable.

$\boxed{\text{B}}$ $\mathcal{L}_A$ is decidable. A decider for $\mathcal{L}_A$ can be converted to a decider for $\mathcal{L}_B$. So, $\mathcal{L}_B$ is undecidable.

$\boxed{\text{C}}$ $\mathcal{L}_A$ is undecidable. A decider for $\mathcal{L}_A$ can be converted to a decider for $\mathcal{L}_B$. So, $\mathcal{L}_B$ is undecidable.

$\boxed{\text{D}}$ $\mathcal{L}_A$ is undecidable. A decider for $\mathcal{L}_B$ can be converted to a decider for $\mathcal{L}_A$. So, $\mathcal{L}_B$ is decidable.

$\boxed{\text{E}}$ $\mathcal{L}_A$ is undecidable. A decider for $\mathcal{L}_B$ can be converted to a decider for $\mathcal{L}_A$. So, $\mathcal{L}_B$ is undecidable.

SCRATCH