**University of Nevada, Las Vegas Computer Science 456/656 Fall 2026**

**Assignment 3: Due Saturday October 19, 2026, midnight**

**Name:**_____

You are permitted to work in groups, get help from others, read books, and use the internet. Turn in the assignment as instruced by the Graduate Assistant, Sabrina Wellace, `wallace4@unlv.nevada.edu`

Officially, the Dyck language is over the alphabet $\{(,)\}$, the left and right parentheses, and consists of all "balanced" strings of parentheses, such as "()" "(())()" However, when written without sufficient caution, such as by rushed students taking an exam, the two symbols look so similar that the grader (me, or whoever) is easily confused. Thus, this semester, I will use the alphabet $\{a, b\}$ instead, so that the Dyck language contains, for example, the strings "$ab$" and "$aabbab$."

1. True **T**/False **F**/Open **O**, meaning that the answer is not known to science at this time.

   (i) _____ $\mathcal{P} = \mathcal{NP}$.

   (ii) _____ If $L$ is $\mathcal{P}$, it must be $\mathcal{NP}$ and also co-$\mathcal{NP}$.

   (iii) _____ The factoring problem for integers, given as binary numerals, is $\mathcal{NP}$-complete.

   (iv) _____ Every language is countable.

   (v) _____ There is an algorithm which decides whether any given context-free grammar generates the Dyck language.

   (vi) _____ For any rational number $q$, there is an algortihm which computes the $n^{\text{th}}$ decimal digit of $q$ for any $n$.

   (vii) _____ There is an algorithm which computes the $n^{\text{th}}$ decimal digit of $\pi$ for any given $n$.

   (viii) _____ For any real number $x$, there is an algortihm which computes the $n^{\text{th}}$ decimal digit of $x$ for any given $n$.

   (ix) _____ There is a program which computes $B(n)$, for any integer $n$, where $B$ is the busy beaver function. (Look it up!)

   (x) _____ The problem of whether two C++ programs are equivalent is decidable. (Teo programs are defined to be *equivalent* if, given the same input they give the same output, regardless of how long they may run.)

2. Label each set listed below as **C**, countable, or **U**, uncountable. Recall that every C++ program is a string over the Ascii alphabet.

   (i) _____ $\mathbb{Z}$, the integers.

   (ii) _____ $\mathbb{Q}$, the rational numbers.

   (iii) _____ $\mathbb{R}$, the real numbers.

   (iv) _____ Complex numbers.

   (v) _____ Albebraic real numbers. (Look it up.)

   (vi) _____ Recursive real numbers.

(vii) _____ Gaussian numbers.

(viii) _____ Binary languges, *i.e.* languages over the binary alphabet $\{0, 1\}$.

(ix) _____ Recursively enumerable binary languages.

(x) _____ C++ programs. (*Not* all programs which have ever been written.)

(xi) _____ Sets of integers.

(xii) _____ Finite sets of integers.

(xiii) _____ Finite sets of real numbers..

3. Give a polynomial time reduction of 3-SAT to the independent set problem.

4. Prove that every decidable language can be enumerated in canonical order by some machine.

5. Suppose there is a machine which enumerates a language $L$ in canonical order. Prove that $L$ is decidable.

6. Let $L$ be a language accepted some machine. Prove that $L$ is recursively enumerable.

7. Suppose there is a machine which enumerates a language $L$. Prove that there is a machine which accepts $L$.