C1 **Theory** (25 points)
Let N denote the sets of non-negative integers.

Let $\sum_1 = \{a, b\}$.

Let $\sum_2 = \{a, b, c\}$.

In this problem (C1) we will make reference to the languages L_1, L_2, and L_3 defined just below.

Definition

(i) $L_1 \stackrel{\text{def}}{=} \{S(i) \mid i \in N\}$ where S is defined recursively as follows:

$$\begin{aligned} S(0) &= b \\ S(i+1) &= S(i)a^{i+1}b \end{aligned}$$

In other words, L_1 consists of the strings b, bab, $babaab$, $babaabaaab$, etc.

(ii) $L_2 \stackrel{\text{def}}{=} \{w \in \sum_1^* \mid$ number of occurrences of substring ab in $w >$ number of occurrences of substring ba in $w\}$.

(iii) $L_3 \stackrel{\text{def}}{=} \{w \in \sum_2^* \mid$ number of occurrences of substring ab in $w >$ number of occurrences of substring ba in $w\}$.

a. (8 points)
Use the Myhill-Nerode theorem to show that L_1 is *not* regular. Recall that the crucial equivalence relation for Myhill-Nerode applied to the language L_1 is $\equiv_{L_1}$, where, for words $x, y \in \sum_1^*$,

$$x \equiv_{L_1} y \stackrel{\text{def}}{\Leftrightarrow} (\forall z \in \textstyle\sum_1^*)[xz \in L_1 \Leftrightarrow yz \in L_1]$$

b. (8 points)
Prove that L_2 is a regular language.

c. (9 points)
Use a pumping lemma to prove that L_3 is *not* a regular language.

C2 **Theory** (25 points)
Let $\sum_1$ and $\sum_2$ be alphabets, and let f be a function from $\sum_1^*$ to subsets of $\sum_2^*$ such that

(i) $f(\lambda) = \{\lambda\}$. (The symbol λ denotes the empty string.)

(ii) For all $n \geq 1$, if $a_1, \ldots, a_n \in \sum_1$, $f(a_1 \cdots a_n) = f(a_1) \cdots f(a_n) = \{w_1 \cdots w_n \mid w_i \in f(a_i), 1 \leq i \leq n\}$. (To clarify notation, $a_1 \cdots a_n$ is the string composed from the letters $a_1, \ldots, a_n$ and $w_1 \cdots w_n$ is the concatenation of strings $w_1, \ldots, w_n$.)

(iii) For each $a \in \sum_1$, $f(a)$ is a regular language.

Now suppose that $L \subseteq \sum_1^*$. Define $f(L) \stackrel{\text{def}}{=} \bigcup_{w \in L} f(w)$. Prove that if L is a regular language, then $f(L)$ is a regular language.

Hint: Make a finite state automaton that accepts $f(L)$.

C3 **Theory** (25 points)
Let N denote the set of non-negative integers. Fix a standard programming formalism ψ for computing all the *one-argument* partial computable functions which map N into N. Code (Godël) number the ψ-programs *onto* N. Let ψ_p denote the partial function computed by program (number) p in the ψ-system. You may assume without proof that, in the ψ-system, Universality, S-m-n, and the Kleene Recursion Theorem (KRT) hold.

Let $K \stackrel{\text{def}}{=} \{p \in N \mid \psi_p(p) \text{ is defined}\}$.

Let $W_p \stackrel{\text{def}}{=}$ the domain of ψ_p. Then $W_0, W_1, W_2, \ldots$ provides a standard listing of *all* the r.e. sets (of non-negative integers).

a. (12 points)
Let $A = \{p \mid W_p \subseteq \overline{K}\}$. Prove or disprove that A is recursively enumerable. (Remember that $\overline{K}$ is the complement of K.)

b. (13 points)
Let p be the encoding of a program such that

$\psi_p(x) = 1 + \psi_x(x)$ if $\psi_x(x)$ is defined and $\psi_p(x)$ is undefined otherwise.

Show that there is no computable function f such that $\psi_p \subseteq f$ (i.e., $f(x) = \psi_p(x)$ whenever $\psi_p(x)$ is defined).

C4 **Theory** (25 points)
a. (20 points)
(Course-of-Values Recursion) Let f be a function defined as

$$\begin{aligned} f(x_1,\ldots,x_n,y) &= h_1(x_1,\ldots,x_n) \text{ if } y=0 \\ &= h_2(x_1,\ldots,x_n,y,[f(x_1,\ldots,x_n,0),\ldots,f(x_1,\ldots,x_n,y-1)]) \text{ if } y>0 \end{aligned}$$

Note that $[a_1,\ldots,a_n]$ is defined as

$$\prod_{i=1}^{n} p_i^{a_i}$$

where p_i is the ith prime number.

Show that if h_1 and h_2 are primitive recursive then so is f.

You may use without proof the primitive recursiveness of standard text book primitive recursive functions and any standard text book closure properties of the class of primitive recursive functions *provided you clearly say which ones you are using when.*

Hint: You might find it useful to first define a function
$g(x_1,\ldots,x_n,y) = [f(x_1,\ldots,x_n,0),\ldots,f(x_1,\ldots,x_n,y)]$ and show that it is primitive recursive.

b. (5 points) Let $f(x_1,\ldots,x_n,t)$ be a computable predicate. show that $\exists t f(x_1,\ldots,x_n,t)$ is recursively enumerable.