Problem 1

In the last homework you proved that the language $L = \{a^n b^{n+1} c^{n+2}\}$ is not context-free. Prove that $L$ is decidable by giving a Turing Machine that decides $L$. Note: in this problem you are required to define the Turing machine formally (you have to do this at least once in your life), i.e., define the set of states, input and tape alphabets etc. You can describe the transition function by drawing the state transition diagram. In order to facilitate grading, please give also an English explanation of the meaning of each state in the machine.

Problem 2

Let $P$ be the language over the alphabet $\{a\}$ of all words $a^p$ such that $p$ is a prime number. (Remember, an integer $p \geq 2$ is prime if it is only divisible by 1 and $p$ itself.) Prove that the language $P$ is decidable, e.g., give a Turing Machine that decides $P$. You can use any of the Turing machine variant from Chapter 3 in the book. (In this problem an informal description of the Turing Machine is enough. However, your description should be sufficiently detailed to make it clear that it can be easily transformed into a formal description.)

Problem 3

Push down automata are very simple computational models that cannot recognize even relatively simple languages like $L = \{w\#w | w \in \{0, 1\}^*\}$. To overcome this limitations we defined a more powerful model of computation: the Turing machine. In this problem we explore a different way to enhance the computational power of a push down automaton. The idea is to enhance the push down automaton with an extra stack, i.e. we consider push down automata with two stacks. In this problem you are required to formalize this model of computation, show that it can recognize the language $L$, and finally prove that push down automata are indeed equivalent to turing machines.

(a) Give a formal definition of 2-stack push down automaton, configurations, computations and the language accepted by such an automaton
(b) Prove that any enumerable language can also be recognized by a 2-stack push down automaton, i.e., show that any turing machine can be transformed into an equivalent 2-stack push down automaton. This proves that 2-stack PDAs are at least as powerful as Turing machines.

(c) Now prove that 2-stack PDAs are not more powerful than Turing machines: show that if a language is accepted by a 2-stack PDA, then the language is enumerable, i.e. show that any 2-stack PDA can be transformed into an equivalent Turing machine.

Problem 4

In previous homeworks you proved that the intersection of regular languages is regular, but the intersection of context free languages is not necessarily context free (i.e. context free languages are not closed under intersection). In this problem you are asked to study how enumerable and decidable languages behave with respect to the intersection operation.

(a) Prove that the intersection of decidable languages is decidable, i.e., given two Turing machines $M_1$ and $M_2$ deciding languages $L_1$ and $L_2$, show that there exists a Turing machine $M$ that decides the language $L = L_1 \cap L_2$.

(b) Prove that the intersection of enumerable languages is enumerable, i.e., given two Turing machines $M_1$ and $M_2$ accepting languages $L_1$ and $L_2$, show that there exists a Turing machine $M$ that accepts the language $L = L_1 \cap L_2$. 