1*) Let $A(n, x)$ be the Ackermann function. Consider the graph $G \subseteq \mathbb{N}^3$ of $A$, i.e., the ternary relation

$$G(n, x, z) \iff A(n, x) = z.$$ 

Show that $G$ is primitive recursive.

2) Consider the following “stack algorithm” for computing the Ackermann function $A(n, x)$:

**Input**: A pair $(n, x) \in \mathbb{N}^2$.

**Algorithm**: Given a sequence $s = (n_1, \ldots, n_k)$ of natural numbers, do the following:

- If $k = 1$, i.e., $s = (n_1)$, Stop: $n_1$ is the output.
- If $k \geq 2$, write $s = t^* (n', x')$ (where $^*$ denotes here concatenation), with $t$ a finite sequence of numbers, perhaps empty. Then do the following:

  - If $n' = 0$, replace $s$ by $s' = t^* (x' + 1)$.
  - If $n' > 0, x' = 0$, replace $s$ by $s' = t^* (n' - 1, 1)$.
  - If $n' > 0, x' > 0$, replace $s$ by $s' = t^* (n' - 1, n', x' - 1)$.

Show that for each input $(n, x)$ this algorithm terminates after finitely many steps (i.e., we reach a sequence of length 1) and the output is the value $A(n, x)$.

3) Show that the following partial function $f : \mathbb{N} \to \mathbb{N}$ is recursive: $f(n) = 1$, if there is a sequence of $n$ consecutive 7’s in the decimal expansion of the square root of 2; $f(n)$ is undefined otherwise.

4) Show that every program $P$ for the RM on an alphabet $A$ can be replaced by an equivalent program $P'$ on the same alphabet which uses only instructions of types I, II, VI, VII. (*Equivalent here means that on any input $(w_1, \ldots, w_n)$ either both $P, P'$ do not halt, or else they both halt and produce
the same output. The intermediate steps of the computation however might be different. Also the program \( P' \) can use more registers than \( P \).)

Note: Homework rules are posted in the Ma/CS 117a web page. Each problem is worth 25 points.