Show that the function \( \lambda_n (\text{\(n\)th digit of} \sqrt{2}) \) is primitive recursive.

**Solution.** The \( n \)th digit of \( \sqrt{2} \) is \( f(n) = \text{Rem}(g(n), 10) \), where \( g(n) = \) the least \( x < 4 \cdot 10^{2n} \) such that \( (x + 1)^2 > 2 \cdot 10^{2n} \).

2. Write a register machine program which computes the exponential function \( \lambda xy (x^y) \). Note that \( x^0 = 1 \) for all \( x \).

**Solution.**

3. For all \( n \) let \( A_n \) be the \( n \)th Ackermann branch, defined by

\[
A_0(x) = 2x, \\
A_{n+1}(x) = A_n \cdots A_n(x). 
\]

Thus \( A_0(x) = 2x, A_1(x) = 2^x, A_2(x) = 2^{2^x} \) (height \( x \)), etc.

Note that, for each \( n \), \( A_n \) is primitive recursive. In fact, \( A_{n+1} \) can be defined by primitive recursion using \( A_n \) as

\[
A_{n+1}(0) = 1, \\
A_{n+1}(x + 1) = A_n(A_{n+1}(x)).
\]
(a) Show that $A_n(1) = 2$, $A_n(2) = 4$, and $A_{n+1}(3) = A_n(4)$ for all $n$. Compute $A_n(x)$ for all $n, x$ with $n + x \leq 8$.

**Solution.**
For all $n$ we have $A_{n+1}(1) = A_n(1)$, hence by induction $A_n(1) = A_0(1) = 2$. Also $A_{n+1}(2) = A_n(A_n(1)) = A_n(2)$, hence by induction $A_n(2) = A_0(2) = 4$. Also, for all $n$ and $x$ we have $A_{n+1}(x+1) = A_n(A_{n+1}(x))$, in particular $A_{n+1}(3) = A_n(A_{n+1}(2)) = A_n(4)$. Table 1 shows $A_n(x)$ for small values of $n, x$.

Table 1: The Ackermann branches.

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A_0$</td>
<td>0</td>
<td>2</td>
<td>4</td>
<td>6</td>
<td>8</td>
<td>10</td>
</tr>
<tr>
<td>$A_1$</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>8</td>
<td>16</td>
<td>32</td>
</tr>
<tr>
<td>$A_2$</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>16</td>
<td>$2^{16}$</td>
<td>$2^{2^{16}}$</td>
</tr>
<tr>
<td>$A_3$</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>$2^{16}$</td>
<td>$2^2$ (height $2^{16}$)</td>
<td>$2^2$ (height $2^2$) (height $2^{16}$)</td>
</tr>
<tr>
<td>$A_4$</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>$2^2$ (height $2^{16}$)</td>
<td>$A_3(2^2$ (height $2^{16}$))</td>
<td></td>
</tr>
<tr>
<td>$A_5$</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>$A_3(2^2$ (height $2^{16}$))</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$A_6$</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(b) Prove the following:

i. $A_n(x+1) > A_n(x) > x$ for all $x \geq 1$ and all $n$.

ii. $A_{n+1}(x) \geq A_n(x+1)$ for all $x \geq 3$ and all $n$.

iii. For each primitive recursive function $f(x_1, \ldots, x_k)$ there exists $n$ such that $A_n$ covers $f$, i.e.,

$$f(x_1, \ldots, x_k) \leq A_n(\max(3, x_1, \ldots, x_k))$$

for all $x_1, \ldots, x_k$.

iv. The 1-place function $\lambda x (A_x(x))$ is not primitive recursive.

v. The 2-place function $\lambda xy (A_x(y))$ is not primitive recursive.

**Solution.** First we prove $A_n(x+1) > A_n(x) > x$ for $x \geq 1$, by induction on $n$. For $n = 0$ we have $A_0(x+1) = 2x + 2 > 2x = A_0(x)$ for all $x$, and $A_0(x) = 2x > x$ for $x \geq 1$. For $n+1$ and $x \geq 1$ we have $A_{n+1}(x+1) = A_n(A_{n+1}(x)) > A_{n+1}(x)$ by inductive hypothesis. Thus $A_{n+1}$ is strictly monotone. Since $A_{n+1}(0) > 0$, it follows that $A_{n+1}(x) > x$ for all $x$. 

2
Next we prove $A_{n+1}(x) \geq A_n(x + 1)$ for $x \geq 3$, by induction on $x$. For $x = 3$ we have $A_{n+1}(3) = A_n(4)$ as noted above, and inductively $A_{n+1}(x + 1) = A_n(A_{n+1}(x)) \geq A_n(A_n(x + 1)) \geq A_n(x + 2)$, since $A_n$ is strictly monotone and $A_n(x + 1) \geq x + 2$ by what has already been proved.

Next we prove that each primitive recursive function is covered by $A_n$ for some $n$. We prove this by induction on the class of primitive recursive functions. We begin by noting that the initial functions are covered by $A_0$.

Suppose $f$ is obtained by generalized composition, say

$$f(x_1, \ldots, x_k) = h(g_1(x_1, \ldots, x_k), \ldots, g_m(x_1, \ldots, x_k)).$$

Let $n$ be such that $A_n$ covers $h$ and $A_{n+1}$ covers $g_1, \ldots, g_m$. We then have

$$f(x_1, \ldots, x_k) = h(g_1(x_1, \ldots, x_k), \ldots, g_m(x_1, \ldots, x_k)) \leq A_n(\max(3, g_1(x_1, \ldots, x_k), \ldots, g_m(x_1, \ldots, x_k))) \leq A_n(A_{n+1}(\max(3, x_1, \ldots, x_k))) = A_{n+1}(\max(3, x_1, \ldots, x_k) + 1) \leq A_{n+2}(\max(3, x_1, \ldots, x_k)),$$

e.g., $A_{n+2}$ covers $f$.

Suppose $f$ is obtained by primitive recursion, say

$$f(0, x_1, \ldots, x_k) = g(x_1, \ldots, x_k),$$
$$f(y + 1, x_1, \ldots, x_k) = h(y, f(y, x_1, \ldots, x_k), x_1, \ldots, x_k).$$

Let $n$ be such that $A_n$ covers $h$ and $A_{n+1}$ covers $g$. We first claim that

$$f(y, x_1, \ldots, x_k) \leq A_{n+1}(y + \max(3, x_1, \ldots, x_k))$$

for all $y, x_1, \ldots, x_k$. We prove this by induction on $y$. For $y = 0$ we have $f(0, x_1, \ldots, x_k) = g(x_1, \ldots, x_k) \leq A_{n+1}(\max(3, x_1, \ldots, x_k))$.

For the inductive step we have

$$f(y + 1, x_1, \ldots, x_k) = h(y, f(y, x_1, \ldots, x_k), x_1, \ldots, x_k) \leq A_n(\max(3, y, f(y, x_1, \ldots, x_k), x_1, \ldots, x_k)) \leq A_n(\max(3, y, A_{n+1}(y + \max(3, x_1, \ldots, x_k))), x_1, \ldots, x_k) = A_n(A_{n+1}(y + \max(3, x_1, \ldots, x_k))) = A_{n+1}(y + 1 + \max(3, x_1, \ldots, x_k))$$

and this proves our claim. We then have

$$f(y, x_1, \ldots, x_k) \leq A_{n+1}(y + \max(3, x_1, \ldots, x_k)) \leq A_{n+1}(2 \max(3, y, x_1, \ldots, x_k)) \leq A_{n+1}(A_{n+2}(\max(3, y, x_1, \ldots, x_k))) = A_{n+2}(\max(3, y, x_1, \ldots, x_k) + 1) \leq A_{n+3}(\max(3, y, x_1, \ldots, x_k)),$$
i.e., $A_{n+3}$ covers $f$. This completes the proof that each primitive recursive function is covered by $A_n$ for some $n$.

Now, if $A_x(x)$ were primitive recursive, then $A_x(x) + 1$ would be primitive recursive, hence covered by $A_n$ for some $n \geq 3$. But then in particular $A_n(n) + 1 \leq A_n(\max(3, n)) = A_n(n)$, a contradiction. Thus the 1-place function $A_x(x)$ is not primitive recursive. It follows immediately that the 2-place function $A_x(y)$ is not primitive recursive.

(c) (Extra Credit) Show that the 3-place relation

$$\{ (x, y, z) \mid A_x(y) = z \}$$

is primitive recursive. Use this to prove that $\lambda x y (A_x(y))$ is recursive. Hence $\lambda x (A_x(x))$ is recursive.

**Solution.** For all $x, y > 0$ we have

$$0 < y < A_x(y) = A_{x-1}(A_x(y - 1)) = A_{x-1}(y')$$

where $y' = A_x(y - 1)$. Since $A_{x-1}(y') = A_x(y) \geq 2$, it follows that $0 < y' < A_{x-1}(y') = A_x(y)$. Repeating this step $x$ times, we obtain a finite sequence $y_0, y_1, y_2, \ldots, y_x$ starting with $y$ such that

$$A_x(y) = A_x(y_0) = A_{x-1}(y_1) = A_{x-2}(y_2) = \cdots = A_0(y_x) = 2y_x,$$

and each of $y_0, y_1, \ldots, y_x$ is $> 0$ and $< A_x(y)$. Moreover, if $y > 2$ then we also have $x < A_x(y)$. Thus the 3-place predicate $A_x(y) = z$ can be defined by course-of-values recursion on $z$ as follows:

$$A_x(y) = z \text{ if and only if }$$

$$(x = 0 \land z = 2y) \lor$$

$$(x > 0 \land y = 0 \land z = 1) \lor$$

$$(x > 0 \land y = 1 \land z = 2) \lor$$

$$(x > 0 \land y = 2 \land z = 4) \lor$$

$$(x > 0 \land y = 2 \land x < z \land \exists y_0, y_1, \ldots, y_x < z$$

$$(y_0 = y \land \forall i < x (y_{i+1} = A_{x-i}(y_i - 1)) \land z = 2y_x)).$$

Actually, the function being defined by primitive recursion is

$$a(w) = \prod \{ p_{2x,3y,5z} \mid A_x(y) = z \land x, y, z < w \}.$$

In any case, it follows that the 3-place predicate $A_x(y) = z$ is primitive recursive.

Applying the least number operator, we see that the 2-place function $A_x(y)$ is recursive. It follows immediately that the 1-place function $A_x(x)$ is recursive.
4. Given a $k$-place partial recursive function $\psi(x_1, \ldots, x_k)$, show that there is a 1-place partial recursive function $\psi^*(z)$ such that

$$\psi^*(p_1^{x_1} \cdots p_k^{x_k}) \simeq p_{k+1}^{\psi(x_1, \ldots, x_k)}$$

for all $x_1, \ldots, x_k$, and $\psi^*(z)$ is computable by a register machine using only two registers, $R_1$ and $R_2$.

**Solution.** We begin with a register machine program $P$ which computes $\psi(x_1, \ldots, x_k)$. Let $P_1, \ldots, P_k, P_{k+1}, \ldots, P_t$ be the registers used in $P$. We may safely assume that, whenever $P(x_1, \ldots, x_k)$ halts, it leaves all registers except $P_{k+1}$ empty.

We transform $P$ into a program $R$ which uses only two registers, $R_1$ and $R_2$. The idea is that, if $P_1, \ldots, P_t$ contain $z_1, \ldots, z_t$ respectively, then $R_1$ contains $z = p_1^{z_1} \cdots p_t^{z_t}$, while $R_2$ contains 0. Incrementing (decrementing) $P_i$ corresponds to multiplication (division) by $p_i$. Each instruction in $P$ is replaced by a corresponding set of instructions in $R$.

We replace $\longrightarrow P_i^- \longrightarrow$ in $P$ by Figure 1 in $R$.

![Figure 1: Incrementing $P_i$. The number of $R_2^+$ instructions is $p_i$.](image)

We replace $\longrightarrow P_i^- \longrightarrow A$ in $P$ by Figure 2 in $R$.

We replace $\longrightarrow$ stop in $P$ by Figure 3 in $R$. 

5
Figure 2: Decrementing $P_i$. The number of $R^{-}_1$ instructions is $p_i$.

Figure 3: Stopping.
5. Let \( T = \{ \text{indices of total recursive functions} \} = \{ e \mid \forall x \varphi^{(1)}_e(x) \downarrow \} \). Let \( E = \{ \text{indices of the empty function} \} = \{ e \mid \forall x \varphi^{(1)}_e(x) \uparrow \} \). Show that \( T \) and \( E \) are not recursive.

Solution. By the Enumeration and Parametrization theorems, we can find a primitive recursive function \( f \) such that \( \varphi^{(1)}_e (x) \downarrow \) if and only if \( \varphi^{(1)}_f (y) \downarrow \). Then \( x \in K \) implies \( f(x) \in T \), while \( x \notin K \) implies \( f(x) \in E \). Thus, since \( T \) and \( E \) are disjoint, \( f \) reduces \( K \) to \( T \) and to the complement of \( E \). Since \( K \) is non-recursive, it follows that \( T \) and \( E \) are non-recursive.

6. Let \( P \) be the class of 1-place partial recursive functions. For \( C \subseteq P \), define \( I_C \) to be the set of indices of functions in \( C \), i.e.,

\[
I_C = \{ e \in \mathbb{N} \mid \varphi^{(1)}_e \in C \}.
\]

Show that if \( \emptyset \neq C \neq P \) then \( I_C \) is nonrecursive. (This result is known as Rice’s Theorem.)

Solution. Let \( e_0 \) be an index of the empty function. Let \( e_1 \) be an index such that \( \varphi^{(1)}_{e_1} \in C \) if and only if \( \varphi^{(1)}_{e_0} \notin C \). By the Enumeration and Parametrization theorems, we can find a primitive recursive function \( f \) such that

\[
\varphi^{(1)}_{f(x)} (y) \simeq \begin{cases} \varphi^{(1)}_{e_1} (y) & \text{if } \varphi^{(1)}_x (x) \downarrow, \\ \uparrow & \text{otherwise,} \end{cases}
\]

for all \( x \) and \( y \). Thus \( x \in K \) implies \( \varphi^{(1)}_{f(x)} = \varphi^{(1)}_{e_1} \), while \( x \notin K \) implies \( \varphi^{(1)}_{f(x)} = \varphi^{(1)}_{e_0} \). Thus \( f \) reduces \( K \) either to \( I_C \) (if \( \varphi^{(1)}_{e_1} \in C \)) or to the complement of \( I_C \) (if \( \varphi^{(1)}_{e_1} \notin C \)). In either case it follows that \( I_C \) is not recursive.

7. Find \( m \) and \( n \) such that \( m \neq n \) and \( \varphi^{(1)}_m (0) = n \) and \( \varphi^{(1)}_n (0) = m \).

Solution.

By the Parametrization Theorem, let \( f \) be a 1-place primitive recursive function such that \( \varphi^{(1)}_{f(x)} (y) = x \) for all \( x, y \). The construction of \( f \) in the proof of the Parametrization Theorem shows that \( f(x) > x \) for all \( x \). By the Recursion Theorem, let \( e \) be such that \( \varphi^{(1)}_e (y) \simeq f(e) \) for all \( y \). In particular we have \( \varphi^{(1)}_e (0) \simeq f(e), \varphi^{(1)}_{f(e)} (0) = e, \) and \( f(e) > e \). Thus we may take \( m = e \) and \( n = f(e) \).

8. A function \( f : \mathbb{N}^k \to \mathbb{N} \) is said to be limit recursive if there exists a recursive function \( g : \mathbb{N}^{k+1} \to \mathbb{N} \) such that, for all \( x_1, \ldots, x_k, f(x_1, \ldots, x_k) = \lim_s g(x_1, \ldots, x_k, s) \).
Let $P$ be a $k$-place predicate. Show that $P$ is $\Delta^0_2$ if and only if $\chi_P$ is limit recursive.

**Solution.**

For simplicity, let $x$ be an abbreviation for $x_1, \ldots, x_k$.

First assume that $\chi_P$ is limit recursive, say

$$\chi_P(x) = \lim_{n} f(n, x)$$

for all $x$, where $f(n, x)$ is a recursive function. Then we have

$$P(x) \equiv \exists m \forall n (n \geq m \Rightarrow f(n, x) = 1)$$

and

$$\neg P(x) \equiv \exists m \forall n (n \geq m \Rightarrow f(n, x) = 0)$$

so $P$ is $\Delta^0_2$.

For the converse, assume that $P$ is $\Delta^0_2$, say

$$P(x) \equiv \exists y \forall z R_1(x, y, z)$$

and

$$\neg P(x) \equiv \exists y \forall z R_0(x, y, z)$$

where $R_1$ and $R_0$ are primitive recursive predicates. Using the bounded least number operator, define $g(n, x) = \text{the least } y < n \text{ such that either } \forall z < n R_1(x, y, z) \text{ or } \forall z < n R_0(x, y, z) \text{ or both, if such a } y \text{ exists, and }$ $g(n, x) = n$ otherwise. Thus $g(n, x)$ is a primitive recursive function, and it is easy to see that, for all $x$, $g(x) = \lim_n g(n, x)$ exists and is equal to the least $y$ such that $\forall z R_1(x, y, z)$ or $\forall z R_0(x, y, z)$. Now define $h(n, x) = 1$ if $\forall z < n R_1(x, g(n, x), z)$, and $h(n, x) = 0$ otherwise. Thus $h(n, x)$ is again a primitive recursive function, and for all $x$, $h(x) = \lim_n h(n, x)$ exists. Moreover $P(x)$ implies $h(x) = 1$, and $\neg P(x)$ implies $h(x) = 0$. Thus $\chi_P$ is limit recursive. This completes the proof.

9. (a) Show that $\text{card}(\mathbb{R}) = 2^{\aleph_0}$, where $\mathbb{R}$ is the set of real numbers.

(b) Show that $\text{card}(\mathbb{R}^\mathbb{R}) = 2^{2^{\aleph_0}}$. (Recall that $\mathbb{R}^\mathbb{R}$ is the set of functions from $\mathbb{R}$ into $\mathbb{R}$.)

(c) Show that $\text{card}(C(\mathbb{R}, \mathbb{R})) = 2^{\aleph_0}$, where $C(\mathbb{R}, \mathbb{R})$ is the set of continuous functions from $\mathbb{R}$ into $\mathbb{R}$.

10. Let $\langle \kappa_i \mid i \in I \rangle$ and $\langle \lambda_i \mid i \in I \rangle$ be indexed families of cardinals with the same index set, $I$. Show that, if $\kappa_i < \lambda_i$ for all $i \in I$, then

$$\sum_{i \in I} \kappa_i < \prod_{i \in I} \lambda_i$$

This result is known as König’s Theorem.
11. Show that the operations of ordinal arithmetic may be defined by transfinite recursion as
\[
\alpha + \beta = \{\alpha\} \cup \bigcup \{(\alpha + \gamma) + 1 \mid \gamma < \beta\},
\]
\[
\alpha \cdot \beta = \bigcup\{(\alpha \cdot \gamma) + \alpha \mid \gamma < \beta\},
\]
\[
\alpha^\beta = \{1\} \cup \bigcup\{(\alpha^\gamma) \cdot \alpha \mid \gamma < \beta\}.
\]
Alternatively, letting \(\delta\) denote a limit ordinal, we may define the operations of ordinal arithmetic by
\[
\alpha + 0 = \alpha,
\]
\[
\alpha + (\beta + 1) = (\alpha + \beta) + 1,
\]
\[
\alpha + \delta = \bigcup\{\alpha + \gamma \mid \gamma < \delta\},
\]
\[
\alpha \cdot 0 = 0,
\]
\[
\alpha \cdot (\beta + 1) = (\alpha \cdot \beta) + \alpha,
\]
\[
\alpha \cdot \delta = \bigcup\{\alpha \cdot \gamma \mid \gamma < \delta\},
\]
\[
\alpha^0 = 1,
\]
\[
\alpha^{\beta+1} = (\alpha^\beta) \cdot \alpha,
\]
\[
\alpha^\delta = \bigcup\{\alpha^\gamma \mid \gamma < \delta\}.
\]
12. Let \(\kappa\) be an uncountable cardinal. The cofinality of \(\kappa\) is defined to be the least cardinal \(\lambda\) such that \(\kappa\) can be written as the sum of \(\lambda\) cardinals each \(< \kappa\). We write \(\text{cf}(\kappa)\) = the cofinality of \(\kappa\).

(a) Show that \(\text{cf}(\kappa)\) is an infinite regular cardinal, and \(\text{cf}(\kappa) \leq \kappa\).

(b) Show that \(\kappa\) is singular if and only if \(\text{cf}(\kappa) < \kappa\).

(c) Show that \(\kappa^{\text{cf}(\kappa)} > \kappa\).

(d) Show that, for all infinite cardinals \(\lambda\), \(\text{cf}(2^\lambda) > \lambda\).

(e) In particular, show that \(\text{cf}(2^{\aleph_0}) > \aleph_0\).

13. Let \(\kappa\) be an inaccessible cardinal. Show that there exists a cardinal \(\lambda < \kappa\) such that \(R_\lambda\) is an elementary submodel of \(R_\kappa\). Show that we may obtain \(\lambda\) with the additional property \(\text{cf}(\lambda) = \aleph_0\).