|  |  |  |  |
| --- | --- | --- | --- |
| Instruction | Mnemonic | Code | What it does |
| LOAD | LDA | 5 | Load the contents of the given mailbox onto the accumulator (calculator).  Note: the contents of the mailbox are not changed. |
| STORE | STA | 3 | Store the contents of the accumulator (calculator) to the mailbox of the given address.  Note: the contents of the accumulator are not changed. |
| ADD | ADD | 1 | Add the contents of the given mailbox onto the accumulator (calculator).  Note: the contents of the mailbox are not changed, and the actions of the accumulator are not defined for add instructions that cause sums larger than 3 digits. |
| SUBTRACT | SUB | 2 | Subtract the contents of the given mailbox from the accumulator (calculator).  Note: the contents of the mailbox are not changed, and the actions of the accumulator are not defined for subtract instructions that cause negative results -- however, a negative flag will be set so that BRP can be used properly (see below). |
| INPUT | INP | 901 | Copy the value from the "in box" onto the accumulator (calculator). |
| OUTPUT | OUT | 902 | Copy the value from the accumulator (calculator) to the "out box".  Note: the contents of the accumulator are not changed. |
| END | HLT | 000 | Causes the Little Man Computer to stop executing your program. |
| BRANCH IF ZERO | BRZ | 7 | If the contents of the accumulator (calculator) are 000, the PC (program counter) will be set to the given address.  Note: since the program is stored in memory, data and program instructions all have the same address/location format. |
| BRANCH IF ZERO OR POSITIVE | BRP | 8 | If the contents of the accumulator (calculator) are 000 or positive (i.e. the negative flag is not set), the PC (program counter) will be set to the given address.  Note: since the program is stored in memory, data and program instructions all have the same address/location format. |
| BRANCH ALWAYS | BRA | 6 | Set the contents of the accumulator (calculator) to the given address.  Note: since the program is stored in memory, data and program instructions all have the same address/location format. |
| DATA LOCATION | DAT | - | When compiled, a program converts each instruction into a three-digit code.  These codes are placed in sequential mailboxes.  Instead of a program component, this instruction will reserve the next mailbox for data storage.  |