CPE 301: Homework 1

Terence Henriod

Section 1101

Instructor: Dr. Egbert

TA: Andy Olson

September 3, 2013

1. **Evaluate Algebraic Expressions**
   1. **Y = X – 5**

|  |  |  |
| --- | --- | --- |
| **The characteristics of the program in mnemonics and plain English (cells in this table correspond to memory locations in the simulated computer)** | | |
| INP12 – Prompts user for a value and places in in location 12 |  | The storage location reserved for user input (X) |
| LDA12 – Loads the accumulator with the stored user input |  |  |
| SUB11 – Performs the subtraction operation using the user input value (in the accumulator) and the constant |  |  |
| STA17 – Stores the result held in the accumulator in the appropriate location to prepare for output |  |  |
| OUT17 – Output the result of the expression to the console |  |  |
| STP – Ends program execution | The memory location reserved for the constant (5) | The memory location reserved for the result of the expression (Y) |

|  |  |  |
| --- | --- | --- |
| **Input** | **Expected Output** | **Screenshot of Result** |
| 7 | 2 |  |
| 5 | 0 |  |
| -77 | -82 |  |

* 1. **Y = 7X + 8**

|  |  |  |
| --- | --- | --- |
| **The characteristics of the program in mnemonics and plain English (cells in this table correspond to memory locations in the simulated computer)** | | |
| INP12 – Prompts user for a value and places in in location 12 | STP – Ends program execution | The storage location reserved for user input (X) |
| LDA12 – Loads the accumulator with the stored user input |  |  |
| MUL10 - Performs the multiplication operation using the current accumulator value and the multipl. constant |  |  |
| SUB11 – Performs the subtraction operation using the current accumulator value and the addition constant |  |  |
| STA17 – Stores the result held in the accumulator in the appropriate location to prepare for output | The memory location reserved for the multipl. constant (7) |  |
| OUT17 – Output the result of the expression to the console | The memory location reserved for the addition constant (8) | The memory location reserved for the result of the expression (Y) |

|  |  |  |
| --- | --- | --- |
| **Input** | **Expected Output** | **Screenshot of Result** |
| 3 | 13 |  |
| 43 | 293 |  |

* 1. **Y = 3A + 5B**

|  |  |  |
| --- | --- | --- |
| **The characteristics of the program in mnemonics and plain English (cells in this table correspond to memory locations in the simulated computer)** | | |
| INP15 – Prompts user for A value and places in in location 15 | MUL13 – Performs the multiplication operation with the current accumulator value and the B multipl. constant | The memory location reserved for the A multipl. constant (3) |
| INP16 – Prompts user for B value and places in in location 16 | ADD14 – Adds the current accumulator value and the intermediate 3 \* A result | The memory location reserved for the B addition constant (5) |
| LDA15 – Loads the accumulator with the stored A input value | STA17 – Stores the result held in the accumulator in the appropriate location to prepare for output | The memory location reserved for the intermediate result of 3 \* A |
| MUL12 - Performs the multiplication operation using the current accumulator value and the A multipl. constant | OUT17 – Output the result of the expression to the console | The storage location reserved for user input (A) |
| STA14 – Stores the intermediate result of 3 \* A in its designated memory location | STP – Ends program execution | The storage location reserved for user input (B) |
| LDA16 – Loads the accumulator with the stored B input value |  | The memory location reserved for the result of the expression (Y) |

|  |  |  |
| --- | --- | --- |
| **Input** | **Expected Output** | **Screenshot of Result** |
| A = 2, B = 5 | 31 |  |
| A = -5, B = 3 | 0 |  |

* 1. **Y = X^2 + 2X – 1**

|  |  |  |  |
| --- | --- | --- | --- |
| **The characteristics of the program in mnemonics and plain English (cells in this table correspond to memory locations in the simulated computer)** | | | |
| INP18 – Prompts user for X value and places in in location 18 | ADD17 – Add the intermediate X^2 result to the accumulator (X^2 + 2X) |  | The memory location reserved for the user input value (X) |
| LDA18 – Loads the stored user input value X in the accumulator | SUB16 – Subtract the constant (1) from the accumulator, resulting in the final evaluation of the expression |  | The memory location reserved for the final result (Y) |
| MUL18 – Multiplies the value in the accumulator with the stored X value (result = X^2) | STA19 – Store the final result from the accumulator into the designated location |  |  |
| STA17 – Store the intermediate result of X^2 in the designated location | OUT19- Output the final result to the console | The memory location reserved for the multipl. constant (2) |  |
| LDA18 – Load the accumulator with the user inputted X value again | STP – End program execution | The memory location reserved for the subtraction constant (1) |  |
| MUL15 – Multiply the current accumulator value with the constant (result = 2X) |  | The memory location reserved for the intermediate result of X^2 |  |

|  |  |  |
| --- | --- | --- |
| **Input** | **Expected Output** | **Screenshot of Result** |
| 3 | 14 |  |
| 0 | -1 |  |
| -3 | 2 |  |

1. **Number Sequences**
   1. **Start at X; multiply by 2 continuously**

|  |  |  |  |
| --- | --- | --- | --- |
| **The characteristics of the program in mnemonics and plain English (cells in this table correspond to memory locations in the simulated computer)** | | | |
| INP17 – Prompts user for initial value X and places it in designated memory location | OUT17 – Output the most recent sequence term to the screen | STP – Ends program execution | The memory location reserved for the decrement constant (1) |
| INP19 – Prompts user for count value C and places it in designated location (does not include first term) | LDA19 – Load the number of remaining terms to generate in the accumulator |  | The memory location reserved for the count of remaining sequence terms to output (second user input) |
| OUT17 – Output the first number of the sequence to the console | SUB18 – Decrement the counter number stored in the accumulator |  |  |
| LDA17 – Load the most recent sequence term into the accumulator | STA19 – Store the updated counter value in the appropriate location |  |  |
| MUL16 – Performs the multiplication operation to produce the next sequence term | SKP04 – Skips the re-loop command if the counter has reached 0 (or less) | The memory location reserved for the multiplication constant (2) |  |
| STA17 – Store the most recent sequence term in the designated location | JMP03 – creates a loop by jumping back to the start of the sequence generation process | The memory location reserved for the result of successive sequence steps (starts with X) |  |

|  |  |  |
| --- | --- | --- |
| **Input** | **Expected Output** | **Screenshot of Result** |
| X = 3, C = 5 | 3, 6, 12, 24, 48, 96… |  |

* 1. **Start at X, multiply by 2, add 1; multiply by 2, add 1…**

|  |  |  |  |
| --- | --- | --- | --- |
| **The characteristics of the program in mnemonics and plain English (cells in this table correspond to memory locations in the simulated computer)** | | | |
| INP17 – Prompts user for initial value X and places it in designated memory location | STA17 – Store the most recent sequence term in the designated location | JMP03 – creates a loop by jumping back to the start of the sequence generation process | The memory location reserved for the decrement constant (1) |
| INP19 – Prompts user for desired term count value C and places it in designated location (does not include first term) | OUT17 – Output the most recent sequence term to the screen | STP – Ends program execution | The memory location reserved for the count of remaining sequence terms to output (second user input) |
| OUT17 – Output the first number of the sequence to the console | LDA19 – Load the number of remaining terms to generate in the accumulator |  |  |
| LDA17 – Load the most recent sequence term into the accumulator | SUB18 – Decrement the counter number stored in the accumulator | The memory location reserved for the addition constant (1) |  |
| MUL16 – Performs the multiplication operation to produce the next sequence term | STA19 – Store the updated counter value in the appropriate location | The memory location reserved for the multiplication constant (2) |  |
| ADD15 – Performs the addition operation in term generation using the appropriate constant | SKP04 – Skips the re-loop command if the counter has reached 0 (or less) | The memory location reserved for the result of successive sequence steps (starts with X) |  |

|  |  |  |
| --- | --- | --- |
| **Input** | **Expected Output** | **Screenshot of Result** |
| X = 2, C = 5 | 2, 5, 11, 23, 47, 95… |  |

* 1. **Start at X; multiply by 3, subtract 1; multiply by 3, subtract 1…**

|  |  |  |  |
| --- | --- | --- | --- |
| **The characteristics of the program in mnemonics and plain English (cells in this table correspond to memory locations in the simulated computer)** | | | |
| INP17 – Prompts user for initial value X and places it in designated memory location | STA17 – Store the most recent sequence term in the designated location | JMP03 – creates a loop by jumping back to the start of the sequence generation process | The memory location reserved for the decrement constant (1) |
| INP19 – Prompts user for desired term count value C and places it in designated location (does not include first term) | OUT17 – Output the most recent sequence term to the screen | STP – Ends program execution | The memory location reserved for the count of remaining sequence terms to output (second user input) |
| OUT17 – Output the first number of the sequence to the console | LDA19 – Load the number of remaining terms to generate in the accumulator |  |  |
| LDA17 – Load the most recent sequence term into the accumulator | SUB18 – Decrement the counter number stored in the accumulator | The memory location reserved for the subtraction constant (1) |  |
| MUL16 – Performs the multiplication operation to produce the next sequence term | STA19 – Store the updated counter value in the appropriate location | The memory location reserved for the multiplication constant (3) |  |
| SUB15 – Performs the subtraction operation in term generation using the appropriate constant | SKP04 – Skips the re-loop command if the counter has reached 0 (or less) | The memory location reserved for the result of successive sequence steps (starts with X) |  |

|  |  |  |
| --- | --- | --- |
| **Input** | **Expected Output** | **Screenshot of Result** |
| X = 2, C = 4 | 2, 5, 14, 41, 122… |  |

* 1. **Fibonacci Sequence: Start with 1 (and an implied 0), sum the two most recent results, store the result, sum the two most recent results, store the result…**

|  |  |  |  |
| --- | --- | --- | --- |
| **The characteristics of the program in mnemonics and plain English (cells in this table correspond to memory locations in the simulated computer)** | | | |
| INP19 – Prompts for desired term count C, stores it in designated location | STA15 – Stores the accumulator value (the old recent term) as the new “previous” value | SKP04 – Skips the re-loop command if counter reaches 0 (or less) | The memory location reserved for the decrement constant (1) |
| LDA16 – Loads the accumulator with the most recently generated term | LDA17 – Loads the newly generated term in order to copy it | JMP03 – creates a loop by jumping to start of sequence generation process | The memory location reserved for the count of remaining sequence terms to output (user input) |
| OUT16 – Outputs the most recently generated term to the console (starts at 1 for first run) | STA16 – Stores current accumulator value (new term) as most recently generated term | STP – Ends program execution |  |
| ADD15 – Adds the “previous” term to the accumulator | LDA19 – Load the number of remaining terms to generate in accumulator | The memory location reserved for the “previous” term (starts at 0 for first run) |  |
| STA17 – Stores the current accumulator value (the resulting sum) as the most recently generated term | SUB18 – Decrement the counter number stored in the accumulator | The memory location reserved for the most recently generated term (starts at 1 for first run) |  |
| LDA16 – Loads what was the most recent term into the accumulator in order to copy it | STA19 – Store the updated counter value in the appropriate location | The memory location reserved for the newly created sum term, really just a temporary storage |  |

|  |  |  |
| --- | --- | --- |
| **Input** | **Expected Output** | **Screenshot of Result** |
| C = 9 | 1, 1, 2, 3, 5, 8, 13, 21, 34… |  |

1. **A Decision Maker**
   1. **Write a program that uses an INP instruction. If the number that you INPut is positive or zero, have your program OUTput the number 1. If the number that you INPut is negative, have the program OUTput the number zero.**

|  |  |  |  |
| --- | --- | --- | --- |
| **The characteristics of the program in mnemonics and plain English (cells in this table correspond to memory locations in the simulated computer)** | | | |
| INP24 – Prompts user for a number X and stores it in the accumulator | OUT18 – Outputs truth value 1 indicating that a number >= 0 was entered | OUT19 – Outputs truth value 0 indicating number < 0 was entered | The memory location reserved for the constant truth value 1 |
| SKP01 – Skips the following instruction the accumulator value is < 0 | STP – Ends program execution | STP – Ends program execution | The memory location reserved for the constant truth value 0 |
| JMP06 – Jumps to the code segment for the case of user input >= 0 |  |  |  |
| JMP12 – Jumps to the code segment for the case of negative user input |  |  |  |
|  |  |  |  |
|  |  |  |  |

|  |  |  |
| --- | --- | --- |
| **Input** | **Expected Output** | **Screenshot of Result** |
| X = 80 | 1 |  |
| X = 0 | 1 |  |
| X = -34 | 0 |  |

* 1. **This exercise is an extension of the above. If the user INPuts a positive number or zero, have your turtle turn right and draw a line 10 units long. If a negative number is INPut, the turtle must turn left and draw the line.**

|  |  |  |  |
| --- | --- | --- | --- |
| **The characteristics of the program in mnemonics and plain English (cells in this table correspond to memory locations in the simulated computer)** | | | |
| INP24 – Prompts user for a number X and stores it in the accumulator | LDA06 – Loads the angle the turtle should turn if a number >= 0 was entered | LDA06 – Loads the angle the turtle should turn if a number < 0 was entered | The memory location reserved for the length of line turtle will draw (10) |
| SKP01 – Skips the following instruction the accumulator value is < 0 | STA22 – Stores the angle value in the designated location | STA22 – Stores the angle value in the designated location |  |
| JMP06 – Jumps to the code segment for the case of user input >= 0 | LDA18 – Loads the line length value in the accumulator | LDA18 – Loads the line length value in the accumulator |  |
| JMP12 – Jumps to the code segment for the case of negative user input | STA23 – Stores the accumulator value in the designated area and activates turtle | STA23 – Stores the accumulator value in the designated area and activates turtle | Memory location reserved for color of line turtle draws, this program uses turquoise (160) by default |
|  | STP – Ends program execution | STP – Ends program execution | Memory location reserved for the angle by which turtle turns before drawing |
|  | The memory location reserved for the direction value the turtle should turn if a number >= 0 was entered (90) | The memory location reserved for the direction value the turtle should turn if a number < 0 was entered (-90) | Memory location reserved for length of line turtle draws |

|  |  |  |  |
| --- | --- | --- | --- |
| **Input** | **Expected Output** | **Screenshot(s) of Result** | |
| X = 53 | Turtle line to the right |  |  |
| X = 0 | Turtle line to the right |  |  |
| X = -2 | Turtle line to the left |  |  |

* 1. **Write a program which determines whether the value the user INPuts is less than or equal to 7. OUT-put a 1 if it is, and a 0 if it is not.**

|  |  |  |  |
| --- | --- | --- | --- |
| **The characteristics of the program in mnemonics and plain English (cells in this table correspond to memory locations in the simulated computer)** | | | |
| INP24 – Prompts user for a number X and stores it in the accumulator | OUT19 – Outputs truth value 0 indicating that a number > 7 was entered | OUT18 – Outputs truth value 1 indicating a number <= 7 was entered | The memory location reserved for the constant truth value 1 |
| SUB05 – Subtracts the constant comparison value from the input value | STP – Ends program execution | STP – Ends program execution | The memory location reserved for the constant truth value 0 |
| SKP04 – Skips the following instruction the accumulator value is <= 0 |  |  |  |
| JMP06 – Jumps to the code segment for the case of user input > 7 |  |  |  |
| JMP12 – Jumps to the code segment for the case of user input <= 7 |  |  |  |
| Memory location designated to hold the constant comparison value (7) |  |  |  |

|  |  |  |
| --- | --- | --- |
| **Input** | **Expected Output** | **Screenshot of Result** |
| X = 8 | 0 |  |
| X = 7 | 0 |  |
| X = 6 | 1 |  |

* 1. **In exercise 1 you were asked to write computer programs which evaluate expressions like X— 5 or 7X. Now write a program which can tell you if: X^2 - 3 = 2X. If it is true for the user's INPut, then OUTput a 1. Output a 0 if it is not true.**

|  |  |  |  |
| --- | --- | --- | --- |
| **The characteristics of the program in mnemonics and plain English (cells in this table correspond to memory locations in the simulated computer)** | | | |
| INP17 – Prompts user for a number X and stores it in the accumulator | MUL16 – Multiply the accumulator value by the multiplier constant to produce 2X | OUT18 – Output constant true 1 to the console indicating equation holds | The memory location reserved for the constant truth value 1 |
| LDA17 – Load the X value into the accumulator | SUB14 – Subtract the other side of the equation to test for equality | STP – End program execution | The memory location reserved for the constant truth value 0 |
| MUL17 – Multiply accumulator value and stored user input (X^2) | SKP06 – Skip if the acc != 0, i.e., skip if equation doesn’t hold | Memory designated for X^2 - 3 intermediate result |  |
| SUB15 – Subtract the subtraction constant from accumulator (X^2 – 3) | JMP12 – Jump to another code segment,  case: the equation holds (acc = 0) | Memory designated for subtraction constant (3) |  |
| STA14 – Store the intermediate result in the designated area | OUT19 – Output constant false 0 indicating equation doesn’t hold | Memory designated for multiplication constant (2) |  |
| LDA17 – Load X value to accumulator to process 2X | STP – End program execution | Memory designated for user input X |  |

|  |  |  |
| --- | --- | --- |
| **Input** | **Expected Output** | **Screenshot of Result** |
| X = 2 | False (0) |  |
| X = 3 | True (1) |  |