CIS 21JA - Lab 7

Write a program that accepts a user input decimal value and prints it as a binary string.

**Details**

Follow the 5 steps below to write your program.

1. (3pts) Write a printString macro that will:

* print a text string of 8 characters, where each output character is separated by a space.
* receive the address of the string.
* Hint: don't use writeString since it won't let you insert a space between characters.

2. (3pts) Write a procedure readInput that will:

* use THE STACK for parameter passing and return value.
* accept an address of a prompt string as input argument
* read in a user input number
* return it on the stack if it's a valid value. Otherwise keep re-prompting if you don't get a positive value that's within one byte.

3. (4pts) Write a procedure toBinary that will:

* use REGISTERS for parameter passing.
* accept the user input number and the address of a binary string variable as input arguments.
* convert the user input number into the appropriate binary characters and store them in the binary string memory space.
* The procedure should not modify the input number, and to do this, it should not have to make a copy of the input number.

4. (3pts) Write a procedure called printOutput that will:

* use the STACK for parameter passing.
* accept the user input number, the address of a text string, and the address of the binary string.
* print the input number, the string "is", and invoke the macro to print the binary string.

5. (2pts) Write a main procedure that calls the readInput, toBinary, and printOutput procedures, passing input arguments and receiving return values as appropriate.

**Additional requirements**

* Define text string variables as needed, and the program should additionally define a binary string which is an array of characters, enough to store the bits of one byte of data.
* Other than the memory variables described above, don't create any other memory variable and instead use the available registers: eax, ebx, ecx, edx, esi, edi, and ebp
* Make sure each procedure uses the correct way to pass data: either through the stack or through registers.  
  1pt of each procedure is to pass data the correct way.
* Make sure each procedure (except for main) uses only the input arguments that are passed to it. These procedures should not access variable names directly.

1/2pt deduction in each procedure for each variable name that is used directly.

**Sample output**:

Enter an 8-bit positive number: -2

Enter an 8-bit positive number: 7

7 is 0 0 0 0 0 1 1 1

Enter an 8-bit positive number: 128

128 is 1 0 0 0 0 0 0 0

Enter an 8-bit positive number: 250

250 is 1 1 1 1 1 0 1 0