Lab 6: myatoi

Due date: By the end of Thursday, **10/17/2019**.

Lab worth **10 points**. **This lab should be submitted individually.**

Description

In this lab, you implement the `myatoi` function with MIPS assembly code which converts a decimal number stored as an ASCIIZ string to an unsigned integer, and complete the main function to test it. The skills you practice in this lab includes MIPS coding, multiplications in MIPS, and dealing with strings. The lab will also help you to understand the representations of numbers.

Implement the `myatoi` function. The `myatoi` function converts an ASCII string to an unsigned 32-bit integer. The function takes as input the address of a string that has the decimal representation of a non-negative number and returns a 32-bit value. The interface of the function looks like:

```
unsigned int myatoi(char   s[]);
```

The function uses a 32-bit unsigned integer, initialized to 0, to keep track of the value to be returned. The function keeps updating the integer for each decimal digit in the string. It returns on one of the following conditions.

1. The function finds a character that is not a decimal digit. In this case, the function returns `v`, where `v` is the value stored in the 32-bit integer.
2. An overflow has happened when the function updates the value. In this case, the function returns `0xFFFF FFFF`. Since `0xFFFF FFFF` indicates error, the largest number the function can convert is `0xFFFF FFFE` (`4294967294`).

The C-like pseudocode is provided on the next page.

The main function in the skeleton code repeats the following steps until the user enters an empty line. Steps 1 and 5 are provided in the skeleton code. **You need to complete Steps 2 to 4.**

1. Use a system call to read a line of user input and place it into buffer `str`. `str` is now a NUL terminated ASCII string. Assume the line is of 126 characters or shorter.
2. Exit from the loop if the first character of `str` is `'\n'` (ASCII value 10) or a NUL.
3. Call `myatoi` to convert `str` to an integer,
4. If `myatoi` returns the error code, print an error message. Otherwise, print the returned integer in three formats, hexadecimal, decimal as unsigned, and decimal as signed. You will use different syscalls to print the same 32 bits.
5. Go to Step 1.

Please follow the MIPS calling conventions. Add brief comments to explain your code.

Deliverables

Submit revised lab6.s, which has your code and comments, in HuskyCT.

To receive full credits, your code should use proper MIPS instructions/pseudoinstructions.

```
/* The function converts an ASCII string to an unsigned integer.
   s is the starting address of a string that has the decimal representation of a non-
   negative number.
*/
unsigned int atoi(char s[])
{
       /* define local variables. */
       char c;
       unsigned int v;
       int    i;

       v = 0;
       i = 0;
       c = s[i];
       while (c >= '0' and c <= '9') {
              /* add the value of the decimal digits*/
              v = v * 10 + (c – '0');    /* Think about why this step works */
              if (there is an overflow) /* when doing * or + */
                     return (unsigned int) -1; /* -1 is represented with 0xFFFFFFFF*/
              i += 1;
              c = s[i];
       }
       return v;
}
```

Below is a sample session of running the code. The blue lines are user input. Note that the program terminates on an empty line. You should use more numbers to test your code (and think about how to test)

```
CSE3666: Lab 6: YOURNAME (YOUR NetID)

4294967296
The number is too large.
4294967294
0xfffffffe 4294967294 -2
10000000000
The number is too large.
```