c. Problem: Powers Program (x^y)

Write a program that calculates the powers of unsigned integers. Using the

.DB directive, create two 8-bit values, one called (labeled) Base, and the

other called Expon. Load these two values from program memory and

compute the result of Base raised to the Expon power, with the result

saved as a 16-bit value (e.g. if Base = 5 and Expon = 4, 54 = 625).

Try your program with various values in Base and Expon to ensure that it

works as expected.

Include “m32def.inc”

Start:

Ldi zl,LOW(2\*base)

Ldi zh, HIGH(2\*base)

lpm r16, z ; base r16

ldi zl, LOW(2\*expon)

ldi zh, HIGH(2\*expon)

lpm r17, z ; expon r17

clr r0 ; result lowbyte

clr r1; result highbyte

clr r2; non-empty adder

clr r3; empty adder high-byte

clr r18; outer counter

cp r18, r16

breq end;

loop1: ; exponent loop

ldi r19, 1; inner counter

clr r2;

add r2, r0

inc r18

loop2: ; multiplier loop

inc r19

add r0, r2

adc r1, r3

cp r19, r16

brne loop2

cp r18, r17 ; see if we’ve gone through expon times

brne loop1

end: rjmp end

; Declarations

base: .DB 2; the base

expon: .DB 10; the exponent

d. Problem: String Compare Program

Write a program that compares two strings to see if they are the same.

Using the .DB directive, create two strings, labeled String1 and String2,

both of which are appropriately terminated (see Part 1). Compare the two

strings, byte by byte, to determine whether the two strings are identical. If

the two strings are identical, end the program with the hexadecimal value

of 0x01 in register R0, otherwise have register R0 be 0x00.

Note: The two strings are not necessarily of the same length, so ensure that

your program works even if the two strings are of different lengths.

Try your program with various ASCII strings in String1 and String2 to

ensure that it works in all cases.