

COS 335 Spring 2009 Assignment 10 Due: Thursday April 29

1.(10 points) Fibonacci Numbers in BCD

The Fibonacci series is defined as $F(n) = F(n-1) + F(n-2)$, where $F(1) = 1$ and $F(2) = 1$. So we have 1,1,2,3,5,8,13,21, ...

Using BCD arithmetic, write a BCD addition subroutine that will handle any size operand (see lab notes 04), then using appropriately sized variables in your program, write program that will display all Fibonacci numbers of 2000 digits or fewer on stdout, and then display the value of N where Fib(N) first exceeds 2000 digits (it is surprisingly small). (Hint: develop, test and debug the program using 10 or 12 digits).

Note that to add BCD numbers it may be easiest to “right-justify” numbers in a maximum length field of containing binary 0s (00h). For example, in 10 digits, $43+69 = 112$ (Note that sum has one extra digit.)

```
00 00 00 00 00 00 00 00 04 03
00 00 00 00 00 00 00 00 06 09
00 00 00 00 00 00 00 00 01 01 02
```

This way you can just loop through all digits without concern for the length of the number. When displaying a sum, however, you should not display leading '0's.

You will need routines to convert unpacked BCD to ASCII and to display numbers without leading 0s.

Here is a routine that can be used to display a string of a known length, without need for a \$ delimiter:

```
OutString:
; display a string to std output device
; parameters:
; dx address of string
; cx number of bytes to display
; returns with all registers unmodified
pusha ; save all regs
sub bx, bx ; handle std output in bx
mov ax, 4000h ; write to file function
int 21h
popa
ret
```

For this assignment, please submit both printed source code and submit source code by email. Include in the printed source your answer to the value of N where the length of Fib(N) > 2000. You do not need submit the output that displays Fibonacci Numbers.