The UNIVERSITY *of* NORTH CAROLINA *at* CHAPEL HILL

Comp 411 Computer Organization
Spring 2007

Problem Set #4
Issued Thursday, 2/08/07; Due Thursday, 2/22/07

Homework Information: Some of the problems are probably too time consuming to be done the night before the due date, so plan accordingly. Late homework will not be accepted. Feel free to get help from others, but the work you hand in should be your own.

Problem 1. ASCII decimal numbers to binary numbers

Write a function `atoi` that converts a string of up to 8 ASCII digits into a 32-bit integer.

```
int atoi(char str[]);
```

The string ends with a null character (i.e., its ASCII code is 0), which could be the 9th character. You don't need to check for errors in the string, i.e., you may assume the string contains only characters '0' through '9'.

Then write the main program, which calls the atoi function by placing the starting address of the string in $a0 and displays the returned integer value in the console window. You may use syscall 1 to display the integer value, as described in the textbook pages A-43 to A-45 (on the CD).

Your program must run in SPIM without error.

Problem 2. Binary numbers to ASCII decimal numbers

Write a function `num2str` that converts a 32 bit integer into a null-terminated ascii string of characters.

```
void num2str(int num, char str[]);
```

Inside your function you may assume that there is sufficient space allocated in the output argument string to store all the characters of num.

Then write the main program, which calls the num2str function and displays the generated string in the console window. You may use syscall 4 to display the string, as described in the textbook pages A-43 to A-45. You'll want to make sure that the pointer you pass to your function points to a sufficiently large area to contain the resulting string. 9 or 10 bytes should be plenty.

Your program must run in SPIM without error.

Problem 3. Putting it all together

The goal of these assignments is to add two numbers to produce a third number. The complication is that the two numbers to be added are each represented by a string of up to 8 ASCII characters for the decimal digits and the result must be represented as a string of up to 8 decimal digits as well. This will allow you to get some experience with the representation of numbers.

Write a main program which uses atoi to convert each input string into a number, add them using MIPS add instruction and then convert the result back to a string using num2str. Display the result using the syscall 4 described in the textbook pages A-42 to A-45. Declare the two input

strings as num1 and num2 exactly as shown in the .data section to help the grading. (Hint: Use .asciiz instead of .ascii will null-terminate the string automatically.)

Your program should work like this:

```
char num1[] = "1234";
char num2[] = "87654321";
char numres[10];

main() {
  int n1 = atoi(num1);
  int n2 = atoi(num2);
  int nr = n1 + n2;
  num2str(nr, numres);
  syscall(4, numres);
}
```

Your program must include the atoi function, the num2str function, and the main program. Your program must run in SPIM without error.