CSE220 Spring 2014 – Practice Questions for Midterm #2

*****These are only samples of the types of questions, it does not cover everything you need to know.**
But it does touch on as many topics as possible.
This is not an indicator of the number of questions on the midterm.***

1. Consider the following C program.

```
#include <stdio.h>
int mystery (char *c) {
        int n = 0;

        while ( *c >= '0' && *c <= '9') {
                n = n+1;
                *c = *c - '0' + 'A';
                c++;}
        return n;
}

int main(void){
        char str[200];
        printf ("Enter a string, end with return\n");
        scanf ("%s", str);
        printf ("%d\n", mystery(str));      <-- 2nd printf
        printf ("%s\n", str);           <-- 3rd printf
        return 0;
}
```

   (a) What does this program print for input string 123-123 ?
   Second printf __________________ Third printf _________________
   (b) In one or two simple sentences, state what does function mystery do?
   (c) If we change declaration of parameter to function mystery to int mystery (char c[ ]), and keep everything else as it is, what would be its effect on the entire program?
   Syntax Error   Execution error   Work as before
   (d) Explain your answer in (c).

2. Consider the fetch-decode-execute cycle for the John Von Neumann machine that we studied in class. Note that it does not have a cache memory.
   (a) Give an example of an instruction that requires accessing memory during execution of an instruction.
   (b) Do we always have to access memory in fetch part of the cycle? Why or why not?

3. Consider an array A of integers in memory of MIPS. The array is indexed by 0, 1, 2, ... etc. Also assume that address of A is in register \$s1. Suppose we want to copy and store the second element of this array (indexed by 2) to variable i. Address of variable i is in register \$s2. Write the exact sequence of instructions MIPS that will accomplish this. You may use a temporary register. (In other words, write MIPS code for statement i = A[2];)

4. Translate the following code into MIPS assembly:
```
A = B – C – 30 + D – E + F
```

Assume that variables A, B, C, D, E, and F are in registers $s0, $s1, $s2, $s3, $s4, $s5. Also, assume that you cannot overwrite any variables (except A) since they are used later in the program.

Write the code such that it uses the minimal number of registers. How many additional registers (other than $s0-$s5) does your code require?

5. MIPS registers
    (a) The MIPS processor has 32 "general purpose" registers. How many bits are needed to refer to them?
    (b) Where do registers reside, inside or outside of the processor? Why?
    (c) Usually, more registers imply faster overall operation (throughput). If that is the case, why do modern machines not have Gigabytes of registers?
    (d) Floating point registers of MIPS are used for storing single precision numbers and/or double precision numbers. How many floating point registers does MIPS have? How do we reference them in MIPS (naming convention)?
    (e) If we store 7 single precision numbers, how many double precision numbers can be stored in the remaining floating point registers?

6. Write the function declaration for `fopen`. Explain each of the parameters and return type. Give an example of its use.

7. Consider the following lines of C code:

```
#define PHRASE_LENGTH 50
struct AcronymNode{
      struct AcronymNode* next;
      char acronym[5];
      double num_phrases;
      struct The_Phrase* phrase_list;
} Dictionary;

struct The_Phrase {
      char phrase[PHRASE_LENGTH];
      int frequency;
      struct The_Phrase* next;
};
```

(a) How many bytes does `Dictionary` take in memory?
(b) What is the size of an instance of `The_Phrase` structure?
(c) Can either structure be optimized to reduce its space in memory?
(d) Given `Dictionary`, assign the frequency of the head of the `phrase_list` to integer i.
    int i =

(e) Write the following function `The_Phrase* get(char* A)` return a pointer to the linked list of all phrases in the `phrase_list` for the `acronymNode` for A

(f) Write the following function `int add(struct ArconymNode* Head, char* newAcronym, char* newPhrase)`

Insert `newAcronym` in the linked list of Acronyms at the tail of the list if the acronym is not in the list already. If the `newAcronym` already exists, check for `newPhrase`. Insert the newPhrase if it doesn't exist, or increment the frequency if it does exists. Remember to create/allocate the linked list elements if they do not already exist. Return 1 if the phrase was inserted into the table, 0 if frequency was incremented.

(g) Declare one instance of the variables of the structure `AcronymNode` (the member variables) as variables in the .data section of a MIPS program.

8. Write a function in C to determine if a string is a palindrome. A string is a palindrome if its reverse is the same as the original string (EX: '12ABCB21' and '229922' are palindromes). Assume the function palin has 2 input parameters, a pointer to the first character of the string and a pointer to the last character of the string. The function returns 1 if it is a palindrome, 0 otherwise.

9. Translate the following code into MIPS assembly: `A[i] = B[i-1] + B[i] + B[i+1]`
Assume that A and B are byte arrays. Both arrays are stored in memory. Register $s1 contains the initial address for array A; register $s2 contains the initial address for array B; and register $s3 contains the value of i. (Hint: use lb & sb. Also, this is not a loop, only a statement).

10. Add a "for" loop around the statement in Question 9. Iterate on variable i for values 1 to 8.

11. Assume the arrays in Question 9 are integer (word) arrays, modify the code.

12. Complete the following C framework using only pointers! Make sure your code would not cause any warnings or errors when compiled with gcc on sparky.

```
int i;
double A[10];
int B[10;]
double * pA = A;
int *pB = B;

for (i = 1; i < 9, i++)
{
     //Implement A[i] = B[i-1] + B[i] + B[i+1] here
}
```

13. Write the MIPS code to exit a program

14. Write the MIPS code to print the integer value in $t0 to the screen

15. Write the MIPS code to read in a string of length 30 characters from the user.