Exam 1 Information

The exam will be in class on Thursday, 8 October. It will cover all material up through Thursday, 24 September. The main topics we have covered so far are:

- C programming (K& R chapters 1–3).
- Background; performance (P&H chapter 1)
- MIPS and assembly language (P&H chapter 2, sections 2.1–2.7)

Question types will include:

- Short answer (“define …”, “what is the value of …”, etc.)
- Write portions of C code
- Explain portions of C code
- Write portions of MIPS code
- Explain portions of MIPS code
- Calculation problems (performance, binary/decimal/hexadecimal computations, etc.)

You will be given the first page of the MIPS reference card and a list of ASCII codes.

Here are examples of the kinds of questions that might be asked. This is not intended to be a sample exam; the topics covered below are not intended to be an exhaustive review. In particular, knowing the answers to all the questions below will not guarantee a good grade on the exam!

1. The following MIPS program, which is supposed to store the value “10” in memory location $x$, produced an execution error that said: “store address not on word boundary”:

```assembly
.data
msg: .asciiz "Hi"
x: .space 4
.text
li $t0,10
sw $t0,x
... rest of program ...
```

How would you repair the program? (Be specific—what goes where?)
For the next 4 problems, assume we are using 16-bit two's-complement representation for integers.

2. What is the decimal value of the number represented by:

```
1111111111111111 (sixteen 1s)
```

3. What is the binary representation of the following hexadecimal value:

```
7af3
```

4. What is the hexadecimal representation of the decimal number 43?

5. What is the hexadecimal representation of the decimal number -43?

6. The following C program reads in a single line of text, a character at a time, and prints it out, replacing each lowercase letter with the corresponding uppercase letter. (For instance, the input “Hello!” would be printed as “HELLO!”.) One or more statements need to be added. Provide them. (You should not need to declare any additional variables; however, if you think you need them you should declare them within the if statement.)

```c
#include <stdio.h>
int main()
{
    int c;
    while ((c = getchar()) != 'n') {
        if (c >= 'a' && c <= 'z') {

```

7. What is the “performance equation”? Identify what each variable stands for and state the units for each variable.

8. Processor A executes a sequence of instructions in $3 \times 10^{-6}$ seconds; processor B executes the same number of instructions in $0.5 \times 10^{-6}$ seconds. Processor B is ______ times faster/slower than A.

9. Processor A has a clock speed of 2 GHz, processor B has a clock speed of 1.5 GHz. The average number of cycles per instruction for processor A is 4. How many cycles per instruction would processor B need to achieve in order to have the same performance as processor A?
For the next 2 questions, assume that the MIPS data setup looks like this:

...  
x: .word  ... some initial value ...  
y: .word  ... some initial value ...  
z: .space 4  
...

10. Write the MIPS statements needed to load and add the values stored in \textit{x} and \textit{y} and store the result in location \textit{z}.

11. Write the MIPS statements needed to store the value 30 in location \textit{z} if \textit{x} == \textit{y}, and store 0 in \textit{z} otherwise.

12. What does the following C program print?

```c
#include <stdio.h>
int main() {
  char s[] = "this is a string";
  s[6] = '\0';
  printf("%s",s);
}
```

13. An R-type MIPS instruction looks like:

\begin{center}
\begin{tabular}{|c|c|c|c|c|c|}
\hline
op & rs & rt & rd & shamt & funct \\
\hline
6 bits & 5 bits & 5 bits & 5 bits & 5 bits & 6 bits \\
\hline
\end{tabular}
\end{center}

\textbf{Instruction fields}

- \textit{op}: operation code (opcode)  
- \textit{rs}: first source register number  
- \textit{rt}: second source register number  
- \textit{rd}: destination register number  
- \textit{shamt}: shift amount (00000 for now)  
- \textit{ funct}: function code (extends opcode)

Here is the hexadecimal representation of an R-format instruction to perform an \textbf{add}:

\begin{center}
00432020
\end{center}

In this instruction, what are the numeric values of the three register numbers \textit{rs}, \textit{rt}, and \textit{rd}?

14. What does the following C code print?

```c
int i = 5;
while(i) {
  printf("%d\n",--i);
}
```

\textbf{Handed out on 1 October 2015}
15. Show the precise output of the following C `printf` statement; use the grid provided to indicate the exact spacing.

```c
printf("%d,%3d,%c\n%5.2f%d\n1234567890\n", 100, 1, 65, 3.333, 5);
```

16. For a certain processor and a certain program consisting of 1000 instructions, half the instructions require 3 clock cycles and the other half require 2 clock cycles. The cycle time is $1 \times 10^{-9}$ sec. How long will it take to execute the program, expressed in microseconds?

17. Write a loop in C that reads in every character in an input file and prints it back out. You may assume that an `int` variable named `c` has already been declared—just write the loop.

18. What value is stored in location `result` by the following MIPS code? (The full program is not shown.)

```mips
li $s0,40  # s0 = 40
li $s1,30  # s1 = 30
slt $t0,$s0,$s1
bne $t0,$zero,label
li $t0,15
sw $t0,result
j label1
label: li $t0,45
sw $t0,result
label1: ... etc. ...
```

For each of the following 3 questions, assume that initially register `$t0` contains the hexadecimal value `0xFFFFFFFF` and register `$t1` contains the hexadecimal value `0x0000000F`.

19. What is the value (in hexadecimal) of register `$t2` after executing the statement:

```mips
and $t2,$t0,$t1
```

20. What is the value (in hexadecimal) of register `$t2` after executing the statement:

```mips
or $t2,$t0,$t1
```

21. What is the value (in hexadecimal) of register `$t2` after executing the statement:

```mips
sra $t2,$t0,1
```

The above is not a full review!

Handout 12  
Handed out on 1 October 2015