CISS360: Computer Systems and Assembly Language
Quiz q1302

Name: aoro1@cougars.ccis.edu Score:

Open `main.tex` and enter answers (look for `answercode`, `answerbox`, `answerlong`). Turn the page for detailed instructions. To rebuild and view pdf, in bash shell execute `make`. To build a gzip-tar file, in bash shell execute `make s` and you'll get `submit.tar.gz`.

Q1. The `sum(a, b)` returns the sum of integers from `a` to `b - 1`. For instance `sum(3, 5)` is `3 + 4 + 5`, i.e. `12`. Of course you can implement this easily with a loop:

```
int sum(int a, int b)
{
    int s = 0;
    for (int i = a; i < b; i++)
    {
        s += i;
    }
    return s;
}
```

However this can also be implemented recursively:

```
int sum(int a, int b)
{
    if (a >= b)
    {
        return 0;
    }
    else
    {
        return a + sum(a + 1, b);
    }
}
```

Write a MIPS program implementing the recursive version of the sum function.
Answer:

```
        .text
        .globl main
sum_exclude_b:
        addi    $sp, $sp, -8
        sw      $ra, 4($sp)
        sw      $a0, 0($sp)
```

```
        # Base case: if a >= b, return 0
        bge     $a0, $a1, base_case

        # Recursive case:
        addi    $a0, $a0, 1               # a = a + 1
        move        $a1, $a1

        jal     sum_exclude_b               # call sum recursively

        lw      $t0, 0($sp)               # retrieve a from stack
        add        $v0, $t0, $v0                # v0 = a + result

        # function epilogue
        lw      $ra, 4($sp)
            addi        $sp, $sp, 8
        jr        $ra

base_case:
        li        $v0, 0
        lw        $ra, 4($sp)
        addi        $sp, $sp, 8
        jr      $ra

main:   li      $v0, 5     # read int and store in a0
        syscall
        move    $a0, $v0
        li      $v0, 5     # read int and store in a1
        syscall
        move    $a1, $v0
        jal     sum_exclude_b      # v0 = sum(a0, a1)
        move    $a0, $v0     # print int v0
        li      $v0, 1
        syscall
        li      $v0, 10     # exit
        syscall
```

Instructions

In `main.tex` change the email address in

```
\renewcommand\AUTHOR{jdoe5@cougars.ccis.edu}
```

to yours. In the bash shell, execute "`make`" to recompile `main.pdf`. Execute "`make v`" to view `main.pdf`. Execute "`make s`" to create `submit.tar.gz` for submission.

For each question, you'll see boxes for you to fill. You write your answers in `main.tex` file. For small boxes, if you see

```
1 + 1 = \answerbox{}.
```

you do this:

```
1 + 1 = \answerbox{2}.
```

`answerbox` will also appear in "true/false" and "multiple-choice" questions.

For longer answers that needs typewriter font, if you see

```
Write a C++ statement that declares an integer variable name x.
\begin{answercode}
\end{answercode}
```

you do this:

```
Write a C++ statement that declares an integer variable name x.
\begin{answercode}
int x;
\end{answercode}
```

`answercode` will appear in questions asking for code, algorithm, and program output. In this case, indentation and spacing is significant. For program output, I do look at spaces and newlines.

For long answers (not in typewriter font) if you see

```
What is the color of the sky?
\begin{answerlong}
\end{answerlong}
```

you can write

```
What is the color of the sky?
\begin{answerlong}
The color of the sky is blue.
\end{answerlong}
```

For students beyond 245: You can put LaTeX commands in `answerbox` and `answerlong`.

A question that begins with "T or F or M" requires you to identify whether it is true or false, or meaningless. "Meaningless" means something's wrong with the statement and it is not well-defined. Something like "$1+_2$" or "$\{2\}^{\{3\}}$" is not well-defined. Therefore a question such as "Is $42 = 1+_2$ true or false?" or "Is $42 = \{2\}^{\{3\}}$ true or false?" does not make sense. "Is $P(42) = \{42\}$ true or false?" is meaningless because $P(X)$ is only defined if X is a set. For "Is $1 + 2 + 3$ true or false?", "$1 + 2 + 3$" is well–defined but as a "numerical expression", not as a "proposition", i.e., it cannot be true or false. Therefore "Is $1 + 2 + 3$ true or false?" is also not a well-defined question.

When writing results of computations, make sure it's simplified. For instance write 2 instead of $1 + 1$. When you write down sets, if the answer is $\{1\}$, I do not want to see $\{1, 1\}$.

When writing a counterexample, always write the simplest.

Here are some examples (see `instructions.tex` for details):

1. T or F or M: $1 + 1 = 2$ ..... T
2. T or F or M: $1 + 1 = 3$ ..... F
3. T or F or M: $1+^2 =$ ..... M
4. $1 + 2 =$ 3
5. Write a C++ statement to declare an integer variable named `x`.

   ```
   int x;
   ```

6. Solve $x^2 - 1 = 0$.

   Since $x^2 - 1 = (x - 1)(x + 1)$, $x^2 - 1 = 0$ implies $(x - 1)(x + 1) = 0$. Therefore $x - 1 = 0$ or $x = -1$. Hence $x = 1$ or $x = -1$.

7. Which is true? C
   (A) $1 + 1 = 0$
   (B) $1 + 1 = 1$
   (C) $1 + 1 = 2$
   (D) $1 + 1 = 3$
   (E) $1 + 1 = 4$