

CS107 Quiz 1 Solution

This is a closed book, closed note, closed computer quiz. You have 7 minutes to complete it.

SUNet ID (@**stanford.edu**): _____

Last Name: _____

First Name: _____

I accept the letter and spirit of Stanford's Honor Code.

[signed] _____

**Do not turn over until
instructed to do so.**

Problem 1: Binary, Hexadecimal, Decimal, and Other Bases

- a. [2 points] Consider the hexadecimal constant 0x47. Express the same constant in binary.

$$47_{16} = 0100\ 0111_2$$

- b. [2 points] Consider the decimal constant 8195, which happens to equal $8192 + 3$, or rather $2^{13} + 3$. Express the same constant in hexadecimal.

$$8192 = 10\ 0000\ 0000\ 0000_2 + 011_2 = 2003_{16}$$

- c. [2 points] What is the largest positive constant that can be represented by a two-digit number in base 5? Assume both digits contribute to magnitude and nothing else. Express your answer in either base 5 or decimal—it's your choice.

$$\text{largest digit in base-5 is 4, largest two-digit constant} = 44_5 = 24_{10}$$

Problem 2: Command-Line Arguments

Consider the following **main** function, which relies on **atoi**. **atoi** accepts a numeric C string (e.g., "46") and returns the corresponding **int** (e.g., 46).

```
// File: sum.c compiles to an executable called sum
int main(int argc, char *argv[]) {
    int sum = atoi(argv[1]) + atoi(argv[2]);
    printf("The sum is %d.\n", sum);
    return 0;
}
```

A quick test suggests that the program logic is correct:

```
myth@stanford.edu$ ./sum 10000 200
The sum is 10200.
```

- a. [2 points] Why does the implementation ignore **argv[0]**? Be brief.

argv[0] is the name of the executable (here: "**./sum**"), so it shouldn't be referenced by the addition.

- b. [2 points] Consider what happens when the program is invoked as shown below.

```
myth@stanford.edu$ ./sum 30000 500 70 9
```

Does the program crash? If not, what gets printed? Be brief.

Doesn't crash at all. **argc** is 5, implementation just ignores **argv[3]** and **argv[4]**
Program prints out "**The sum is 30500.**"