**Computer Science 162**
**Sam Kumar**
**University of California, Berkeley**
**Quiz 0**
**June 24, 2020**

| Name | |
|---|---|
| Student ID | |

This is an optional assignment; it does not count for a grade. It is a set of questions, compiled in the format of a quiz, covering the prerequisite material for CS 162. We do not expect most students to know the answer to all of the questions. However, if concepts covered in this exam seem completely foreign to you, you should quickly learn the corresponding material or take one of the prerequisite classes, like CS 61C or CS 70, instead.

This practice quiz is not intended to be comprehensive; it is merely a useful resource for you to identify gaps in your knowledge coming into the class. The length and difficulty of this practice quiz is not indicative of the length and difficulty of the other quizzes in CS 162.

As an opportunity to become familiar with the online exam platform we will be using, this quiz will be given in the online exam format during the first week of class.

**Grade Table (for instructor use only)**

| Question: | 1 | 2 | 3 | 4 | 5 | 6 | Total |
|---|---|---|---|---|---|---|---|
| Points: | 2 | 9 | 10 | 15 | 5 | 0 | 41 |
| Score: | | | | | | | |

1. (2 points) **Course Policies**

   Please reference the course info page if your are uncertain about these questions.

   (a) (1 point)  Are you allowed to look at online resources during this quiz?

   ○ Yes, because all exams are open book

   ○ No

   (b) (1 point)  Are you allowed to message a friend or post on StackOverflow during this quiz?

   ○ Yes

   ○ No, because collaboration with other people is forbidden during exams

2. (9 points) **General Questions**

(a) (1 point) Write the number `1024` in hexadecimal.

(a) _____

(b) (1 point) Write the hexadecimal value `0xdec0de` in decimal.

(b) _____

(c) (1 point) What is the largest unsigned value that can be represented in bits 2 to 7, inclusive, of a memory address?

(c) _____

(d) (1 point) You want to write a function `sum_array` that accepts an array of integers as input and returns the sum of all elements in the array. Which of the following is the most appropriate signature for the the `sum_array` function?

- ○ `int sum_array(int[] array);`
- ○ `int sum_array(int* array);`
- ○ `int sum_array(int* array, size_t array_length);`
- ○ `int sum_array(int** array);`
- ○ `int sum_array(int** array, size_t array_length);`

(e) (1 point) You want to write a function `range` that an integer `n` as input and returns an array containing the integers `0` to `n`. Which of the following is the most appropriate signature for the the `range` function?

- ○ `int range(unsigned int n);`
- ○ `int* range(unsigned int n);`
- ○ `int** range(unsigned int n);`

(f) (1 point) When executing an instruction to retrieve a value from memory, which of the following determines whether the CPU retrieves the value from the cache instead of physical memory?

- ○ A special, predetermined range of memory addresses correspond to the cache instead of physical memory.
- ○ A special type of load instruction is used to retrieve data from the cache instead of physical memory.
- ○ The CPU loads a value from the cache if it finds a cache entry whose tag matches that of the accessed memory address.
- ○ The CPU loads a value from the cache if it finds a cache entry whose tag and index match those of the accessed memory address.
- ○ The CPU loads a value from the cache if it finds a cache entry whose tag, index, and offset match those of the accessed memory address.

(g) (1 point) Suppose that `uint32_t* x` is a variable in a C program that currently contains the value `0x4000`. What will `&x[3]` evaluate to? (Hint: `&x[3]` is equivalent to `&(x[3])`.)

○ `0x4000`

○ `0x4003`

○ `0x400c`

○ `0x4012`

○ Not enough information is given to determine the answer.

(h) (1 point) Suppose that `uint32_t* x` is a variable in a C program that currently contains the value `0x4000`. What will `&x` evaluate to?

○ `0x4000`

○ `0x4003`

○ `0x400c`

○ `0x4012`

○ Not enough information is given to determine the answer.

(i) (1 point) Consider a computer with 4,096 byte pages and 32-bit memory addresses. Global variables start at the address `0x08048000`. Say there is an integer that sits `0x4814` bytes above this address. What is the virtual page number, in hexadecimal, of the page containing this address?

(i) _____

3. (10 points) **Caches**

   Consider a computer with a 32 KiB eight-way set associative cache with 64 B cache blocks.

   (a) (3 points) Let bit 0 of a memory address represent the least significant bit, and let bit 31 represent the most significant bit. Which bits correspond to the tag? Which bits correspond to the index? Which bits correspond to the offset?

   bits [    31    :  ____  ]   are the tag

   bits [  ____  :  ____  ]   the index

   bits [  ____  :   0   ]   are the offset

   (b) (2 points) Before adding a cache to the computer, you ran measurements and found that it takes 100 ns to perform a memory access. With the cache, you can service cache hits in just 2 ns. What cache hit rate is required for the average memory access time to be below 10 ns?

   (b) _____

   (c) (1 point) Suppose that we increase the associativity of the cache while keeping the cache size the same. For typical workloads (e.g., those exhibiting spatial and temporal locality), how would this affect the cache's hit rate?
   - ○ It would increase
   - ○ It would decrease
   - ○ It would stay about the same

   (d) (1 point) Why do most processors use a set-associative cache instead of a fully-associative cache?

   Now suppose that our computer has two CPU cores, each with its own 32 KiB eight-way set associative cache.

   (e) (2 points) Suppose that we wish to compute the sum of the elements of a very long array containing 64-bit integers. Here are two strategies to parallelize this task:

   1. In parallel, the first core sums the first half of the array and the second core sums the second half of the array. Then, the partial sums are added.
   2. In parallel, the first core sums the even-indexed elements of the array and the second core sums the odd-indexed elements of the array. Then, the partial sums are added.

   Which approach is faster?
   - ○ Strategy #1 will compute the total sum faster than Strategy #2.
   - ○ Strategy #2 will compute the total sum faster than Strategy #1.
   - ○ Both strategies will perform about the same.

Explain your answer:

<br>

(f) (1 point) Suppose that, in a C program, the one-byte variable x is a global variable. Is there ever a situation where the cache block containing x is stored in both cores' caches? Explain.

4. (15 points) **C Programming and Data Structures**

   (a) (3 points) Implement `char* strcpy(char* dest, const char* src)` from the C standard library. You may assume that `src` and `dest` point to allocated memory buffers large enough to store the string in `src`, including its null terminator (but not necessarily any larger), and that the two buffers do not overlap. It should return the value of `dest`. Do not use any functions from the C standard library.

<br><br><br><br><br><br><br><br><br>

Consider a map data structure (like a Python dictionary) implemented using a singly linked list in C. Keys and values both have the type `uint64_t`. The linked list is implemented using the following data structure for a node of the linked list.

```
typedef struct node {
    uint64_t key;
    uint64_t value;
    node_t* next;
} node_t;
```

The overall map data structure is declared as follows:

```
typedef struct map {
    node_t* head;
} map_t;
```

The `next` field of the last node in the linked list is set to `NULL`. An empty linked list is represented by a `NULL` head pointer.

   (b) (4 points) Suppose that `map_t* x;` is a variable whose value is currently stored in a register. Assume that `x->head != NULL`.

      i. (1 point) How many memory accesses must be performed to evaluate the expression `x->head->key`?

                                                           i. _____

      ii. (1 point) How many memory accesses must be performed to evaluate the expression `x->head->next`?

                                                           ii. _____

      iii. (1 point) How many memory accesses must be performed to evaluate the expression `&x->head->next`?

                                                         iii. _____

iv. (1 point) Suppose we want to assign the variable `y` as follows: `y = &x->head->next;`. As what type should `y` be declared?

iv. _____

(c) (2 points) Suppose we run our code on a 32-bit system where the pointer size is 4 bytes.
   i. (1 point) What is `sizeof(map_t)`?

i. _____

   ii. (1 point) What is `sizeof(node_t)`?

ii. _____

(d) (3 points) Implement `node_t* map_find(map_t* map, uint64_t key)`, which returns a pointer to the node corresponding to the provided key, or `NULL` if the key is not in the map.

(e) (3 points) Implement `void map_insert(map_t* map, uint64_t key, uint64_t value)`. If the `key` is already in the map, its value should be updated. Otherwise, a new key-value pair should be inserted. You may call `map_find` in your answer.

5. (5 points) **Probability**

   (a) (2 points) Let $X$ be a random variable that is normally distributed with a mean of 10 and a standard deviation of 1. What is $\Pr[X > 3]$? Write your answer in terms of $\Phi$, the CDF of the standard normal distribution.

   ┌─────────────────────────────────────────────────────────────────┐
   │                                                                   │
   │                                                                   │
   │                                                                   │
   │                                                                   │
   │                                                                   │
   └─────────────────────────────────────────────────────────────────┘

   (b) (2 points) Let $Y$ be a random variable that is exponentially distributed with parameter $\frac{1}{2}$.
       i. (1 point) What is the mean (expectation) of $Y$?

                                                              i. _____

       ii. (1 point) What is the variance of $Y$?

                                                              ii. _____

   (c) (1 point) The exponential distribution is said to be *memoryless*. What does this mean?

   ┌─────────────────────────────────────────────────────────────────┐
   │                                                                   │
   │                                                                   │
   │                                                                   │
   │                                                                   │
   │                                                                   │
   └─────────────────────────────────────────────────────────────────┘

6. (0 points) **Optional Questions**

    (a) (0 points) Having finished the exam, how do you feel about it? Check **all** that apply:

      ☐ 🙂    ☐ ☹️    ☐ 😐    ☐ 😕    ☐ 😑    ☐ 😵    ☐ 😎    ☐ 🤯

      ☐ 😴    ☐ 🐱    ☐ Other (please draw):

    (b) (0 points) If there's anything you'd like to tell the course staff (e.g., feedback about the class or exam, suspicious activity during the exam, new logo suggestions, etc.) you can write it on this page.