This is **CS50**.
Harvard University Fall 2012

Quiz 1

out of 121 points

Do not turn this page over until told by the staff to do so.

This quiz is "closed-book." However, you may utilize during the quiz one two-sided page (8.5" × 11") of notes, typed or written, and a pen or pencil, nothing else.

Scrap paper is included at this document's end.
Unless otherwise noted, you may call any functions we've encountered this term in code that you write.
You needn't comment code that you write, but comments may help in cases of partial credit.
If running short on time, you may resort to pseudocode for potential partial credit.

Name ____________________

Harvard ID number ____________________

Circle your course.

CS50 CSCI E-52

Circle your location, if on campus.

Aldrich 210	Harvard Hall 201	Maxwell Dworkin 223	Sever Hall 113
Fong Auditorium	Harvard Hall 202	Northwest Science B103	Sever Hall 202
Geological Lecture Hall	Lowell Lecture Hall	Northwest Science B104	Sever Hall 203
Harvard Hall 104			Tsai Auditorium

Circle your teaching fellow's name.

Alex Chang	David DiCiurcio	Katryna Cadle	Nate Herman
Ali Nahm	Doug Lloyd	Kevin Mu	Ore Babarinsa
Alisa Nguyen	Elena Agapie	Kevin Schmid	Paul Bowden
Angela Li	Emmet Jao	Komal Syed	Peter Hung
Balaji Pandian	Ian Nightingale	Larry Ehrhardt	R.J. Aquino
Bannus Van der Kloot	Iva Milo	Lauren Carvalho	Яob Bowden
Ben Shryock	Jackson Steinkamp	Levi Roth	Ryan Lee
Blake Walsh	Jacob Pritt	Lexi Ross	Sebastian Pierce-Durance
Bo Han	Jelle Zijlstra	Lucas Freitas	Tim McLaughlin
Casey Fleeter	Jimmy Sun	Mark Grozen-Smith	Tommy MacWilliam
Casey Grun	Joe McCormick	Meg Quintero	Travis Downs
Chris Gerber	John Mussman	Melissa Niu	Tyler Morrison
Chris Mueller	Jonathan Miller	Michelle Luo	Vipul Shekhawat
Christopher Bartholomew	Jordan Jozwiak	Mike Tucker	Wesley Chen
	Joseph Ong	Mimi Xu	Yaniv Yacoby
Conner Dalton	Joy Ming	Mishal Rahman	Yixiao Wang
Cynthia Meng	Julia Mitelman	Nancy Chen	Yuechen Zhao
Dan Bradley	Jun S. Lee	Naomi Bolotin	Zak Burke
Daven Farnham	Karen Xiao	Nate Hardison	Zamyla Chan

for staff use only

final score out of 121

Multiple Choice.

For each of the following questions or statements, circle the letter (a, b, c, or d) of the one response that best answers the question or completes the statement; you need not explain your answers.

0. (1 point.) If `s` is a pointer to a C `struct` with a field called `id`, then `s->id` is equivalent to:
   a. `&id`
   b. `s.id`
   c. `(*s).id`
   d. `*id`

1. (1 point.) Web servers typically listen for HTTP traffic on TCP port:
   a. 21
   b. 22
   c. 25
   d. 80

2. (1 point.) jQuery is a:
   a. JavaScript library
   b. PHP library
   c. programming language
   d. SQL statement

3. (1 point.) A CSS selector that selects the HTML element whose `id` attribute has a value of `foo` is:
   a. `.foo`
   b. `#foo`
   c. `id="foo"`
   d. `foo`

True or False.

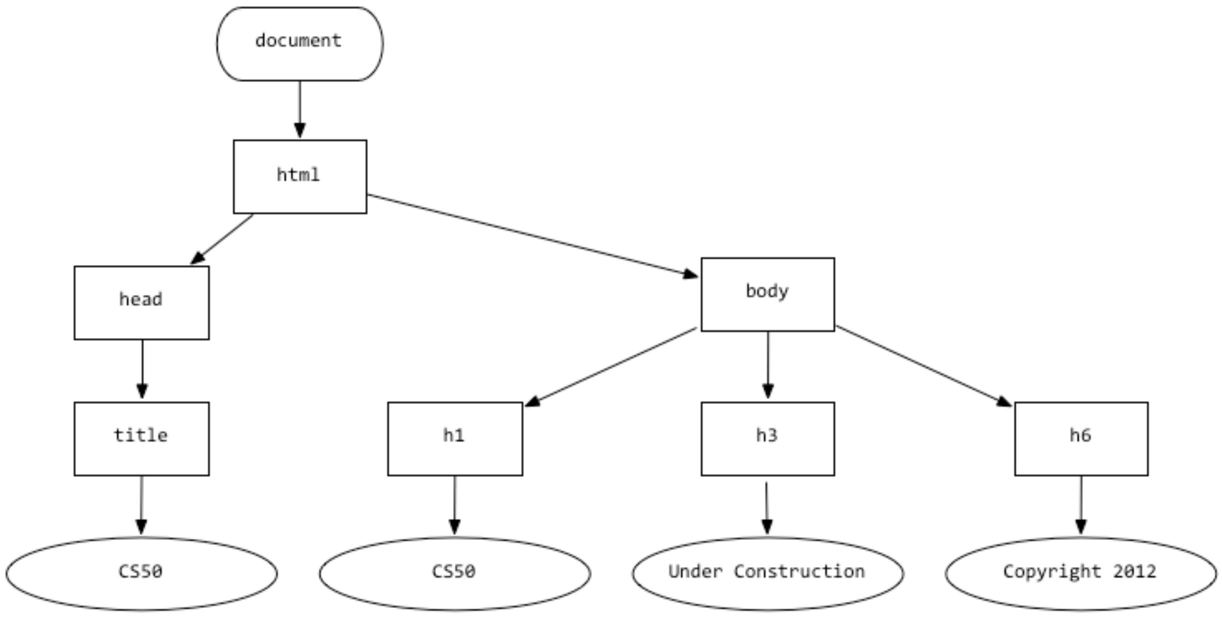
For each of the statements below, circle T if the statement is true or F if the statement is false.

4. T F (1 point.) A queue is a last-in, first-out data structure.
5. T F (1 point.) A stack is a first-in, first-out data structure.
6. T F (1 point.) Each node in a trie has 0, 1, or 2 children.

for staff use only
—

DOM, DOM DOM DOM.

7. (4 points.) Consider the DOM below.



Suppose that this DOM represents an HTML document, wherein each rectangle represents an HTML element, and each oval represents text. In the space below, complete our conversion of this DOM to valid HTML.

```
<!DOCTYPE html>

<html>
```

for staff use only
–

`GetInt` 1.0.

Consider the simplified implementation of `GetInt` below, to which line numbers have been added for the sake of discussion.

```
1   #include <stdio.h>
2
3   int GetInt()
4   {
5       int n;
6       scanf("%d", &n);
7       return n;
8   }
```

8. (2 points.) In a sentence, exactly what does `scanf` do in line `6` with respect to `n`?

9. (2 points.) In a sentence, why must `n` be passed into `scanf` by reference?

`GetString`.

10. (2 points.) Consider the simplified (and buggy) implementation of `GetString` below, to which line numbers have been added for the sake of discussion.

```
1   #include <stdio.h>
2
3   char* GetString()
4   {
5       char* s;
6       scanf("%s", s);
7       return s;
8   }
```

It turns out this implementation tends to segfault. In no more than three sentences, explain why.

for staff use only
—

`GetInt` 2.0.

Recall the actual implementation of `GetInt` below, to which line numbers have been added for the sake of discussion.

```
1   int GetInt(void)
2   {
3       // try to get an int from user
4       while (true)
5       {
6           // get line of text, returning INT_MAX on failure
7           string line = GetString();
8           if (line == NULL)
9               return INT_MAX;
10
11          // return an int if only an int (possibly with
12          // leading and/or trailing whitespace) was provided
13          int n; char c;
14          if (sscanf(line, " %d %c", &n, &c) == 1)
15          {
16              free(line);
17              return n;
18          }
19          else
20          {
21              free(line);
22              printf("Retry: ");
23          }
24      }
25  }
```

11. (2 points.) Suppose that a user, when prompted for input, inputs

    `foo`

    followed by Enter, and so `line` is assigned a value of `"foo"` in line `7`. What value will `sscanf` return in line `14`?

12. (2 points.) What role does `%c` play in line `14`?

13. (2 points.) Notice how, upon failure, `GetInt` returns `INT_MAX`, a constant defined in `limits.h` that represents the maximum value of an `int` (which happens to be `2147483647` in the CS50 Appliance). Why does `GetInt`, by design, return such a large value instead of, say, `1` or `-1` (which more commonly indicate errors) upon failure?

for staff use only
—

Register here.

14. (6 points.) Consider the web page below, in which there's a `form` that submits to `register.php`.

```
<html>
    <head>
        <title>Register</title>
    </head>
    <body>
        <form action="register.php" method="post" id="registration">
            Email: <input id="email" name="email" type="text"/>
            <br/>
            Password: <input id="password" name="password" type="password"/>
            <br/>
            Password (again): <input id="confirmation" name="confirmation" type="password"/>
            <br/><br/>
            <input type="submit" value="Register"/>
        </form>
    </body>
</html>
```

Complete the implementation of `register.php` below in such a way that the page's `body` displays, quite simply, `SUCCESS`, if the `form` is submitted with non-empty values for all three form fields and with identical values for the two password fields. Else the page's `body` should display `FAILURE`. In other words, your PHP code should `echo` or `print` precisely one of those words.

```
<html>
    <head>
        <title>Register</title>
    </head>
    <body>
        <?php









        ?>
    </body>
</html>
```

for staff use only
–

15. (6 points.) Now suppose that the registration process is enhanced with some client-side validation. Complete the implementation of the web page below in such a way that the form (toward this page's bottom) is only submitted to `register.php` if all three form fields have non-empty values and if the values of the two password fields are identical.

```
<!DOCTYPE html>

<html>
    <head>
        <script src="http://code.jquery.com/jquery-latest.js"></script>
        <script>

            // onready
            $(document).ready(function() {

                // onsubmit
                $('#registration').submit(function() {


















                });

            });

        </script>
        <title>Register</title>
    </head>
    <body>
        <form action="register.php" method="post" id="registration">
            Email: <input id="email" name="email" type="text"/>
            <br/>
            Password: <input id="password" name="password" type="password"/>
            <br/>
            Password (again): <input id="confirmation" name="confirmation" type="password"/>
            <br/><br/>
            <input type="submit" value="Register"/>
        </form>
    </body>
</html>
```

for staff use only
—

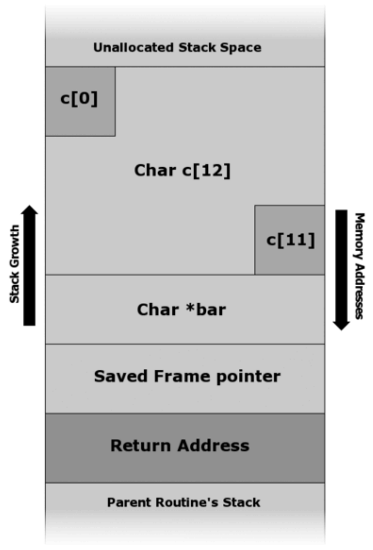
Under attack.

Recall the program below from Week 5's discussion of buffer overflows, alongside which is a depiction of the program's stack if execution is paused (as with a breakpoint) inside of `foo`.

```
#include <string.h>

void foo(char* bar)
{
    char c[12];
    memcpy(c, bar, strlen(bar));
}

int  main(int  argc,  char*  argv[])
{
    foo(argv[1]);
    return 0;
}
```



Also recall that `memcpy` "copies `n` bytes from memory area `src` to memory area `dest`", as per this prototype, wherein `void*` simply represents a pointer to any type:

```
void* memcpy(void* dest, const void* src, size_t n);
```

16. (2 points.) In no more than three sentences, what's a buffer overflow, generally speaking?

17. (2 points.) In the context of this program specifically, which buffer is at risk for overflow?

18. (3 points.) In no more than three sentences, explain how an adversary can "trick" a program into executing code via a buffer overflow.

for staff use only
–

Pointer fun with singly linked lists.

Consider the program below, whose purpose in life is to store non-negative integers in a (global) sorted linked list, ordered from smallest to largest with no duplicates, to which line numbers have been added for the sake of discussion.

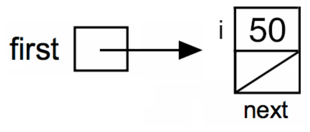
```
1   #include <stdlib.h>
2
3   typedef struct node
4   {
5       unsigned int i;
6       struct node* next;
7   }
8   node;
9
10  node* first = NULL;
11
12  void delete(unsigned int i);
13  void insert(unsigned int i);
14
15  int main(void)
16  {
17      insert(50);
18      insert(15);
19      insert(16);
20      insert(23);
21      insert(4);
22      insert(42);
23      insert(8);
24      delete(50);
25      return 0;
26  }
27
28  ...
```

19. (2 points.) In terms of *O*, what's the running time of `insert`, if *n* represents the linked list's length, assuming `insert` maintains the list's order?

20. (2 points.) In terms of Ω, what's the running time of `delete`, if *n* represents the linked list's length, assuming `delete` maintains the list's order?

for staff use only
—

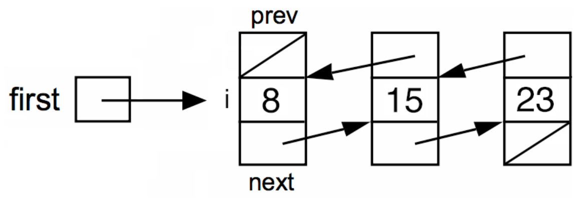
21. (2 points.) Suppose that the drawing below represents this program's linked list if execution is paused (as with a breakpoint) after line `17` executes but before line `18` executes.



Now suppose that execution is instead paused after line `24` executes but before line `25` executes. In the space below, draw this program's linked list at that moment in time. Just as we've done, you may depict any non-`NULL` pointer with an arrow and any `NULL` pointer with a slash through its box.

Pointer fun with doubly linked lists.

Suppose that the drawing below represents a sorted "doubly linked list," whose purpose in life is also to store non-negative integers, ordered from smallest to largest with no duplicates, whereby each `node` in the list has a pointer (`prev`) to the `node` before it, an `unsigned int` (`i`), and a pointer (`next`) to the `node` after it. Meanwhile, `first` is a global pointer to the first `node` in the list. Depending on the list's length, any of these pointers could be `NULL`. The list drawn below happens to be of length 3, but a doubly linked list can be of any length.



22. (3 points.) Complete the below definition of `node` for a doubly linked list.

```
typedef struct node
{






}
node;
```

for staff use only
–

23. (8 points.) Complete the implementation of `insert` below in such a way that the function inserts `i`, if not already present, into a sorted doubly linked list (of any length) to which there's a global pointer, `first`. Do not assume that `first` will be non-`NULL`. Take care to preserve the list's order.

```
void insert(unsigned int i)
{
```

for staff use only
–

Such a grind. (2 points each.)

Embedded in each of the error messages from `valgrind` below is an explanation of something gone wrong. For each message, explain, in no more than three sentences, what the programmer has done wrong and propose how to fix.

24. `Invalid write of size 4`

25. `definitely lost: 40 bytes in 1 blocks`

All your base.

26. (6 points.) Complete the table below in such a way that each row's values are equal. It's fine to omit leading zeroes.

Binary	Decimal	Hexadecimal
00000000	0	0x00
	32	
00110010		
		0xdf

for staff use only
–

Bold Claims. (2 points each.)

For each of the claims below, state whether the claim is correct or incorrect and explain, in no more than three sentences, why the claim is correct or incorrect.

27. Just the other day, Zamyla claimed that HTML is a programming language.

28. Just the other day, Lexi claimed that encoding a file with Huffman coding can sometimes increase the file's size.

29. Just the other day, Lucas claimed that GIF is a lossy format.

30. Just the other day, Tommy claimed that you can trust programs whose source code is free of backdoors.

for staff use only
–

BSTs.

Suppose that each `node` in a binary search tree is defined per the below, wherein each `node` encapsulates an `int` plus a pointer to a left child, if any, and a pointer to a right child, if any.

```
typedef struct node
{
    int i;
    struct node* left;
    struct node* right;
}
node;
```

31. (4 points.) Using recursion, complete the implementation of `find` below in such a way that the function returns `true` if `i` is present in the binary search tree rooted at `root`, else it returns `false`. Do not assume that `root` will be non-`NULL`.

```
bool find(node* root, int i)
{
```

32. (4 points.) Without using recursion, complete the implementation of `find` below in such a way that the function returns `true` if `i` is present in the binary search tree rooted at `root`, else it returns `false`. Do not assume that `root` will be non-`NULL`.

```
bool find(node* root, int i)
{
```

for staff use only
–

Nom nom nom.

33. (1 point.) Consider the (simplified) HTTP request headers below.

```
GET /quizzes/ HTTP/1.1
Host: www.cs50.net
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_8_2)
```

According to these headers, what URL was visited?

`http://`

34. (3 points.) Consider the (simplified) HTTP response headers below.

```
HTTP/1.1 200 OK
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Server: Apache
Set-Cookie: PHPSESSID=dd03b3d7p3n6ruaap1q36ag990; path=/
```

Explain, in no more than three sentences, the relationship between `PHPSESSID` and `$_SESSION`, taking care to define each.

How odd.

It turns out that you can determine whether binary number is even or odd simply by examining its least-significant (i.e., rightmost) bit. If that bit is a `0`, the number is even; if that bit is a `1`, the number is odd. For instance, `00000001` (otherwise known as 1 in decimal) is odd, and `00000010` (otherwise known as 2 in decimal) is even.

35. (4 points.) Complete the implementation of `odd` below without using `/` or `%`, instead <u>using one or more bitwise operations</u>, in such a way that the function returns `true` if `n` is odd and `false` if `n` is even.

```
bool odd(unsigned int n)
{
```

for staff use only
–

Let's talk about compilers.

Consider the source code below.

```
#include <stdio.h>

int main(void)
{
    printf("hello, world\n");
    return 0;
}
```

36. (4 points.) Explain in four sentences the process by which this source code becomes executable, beginning each sentence per the below.

    During pre-processing...

    During compiling...

    During assembling...

    During linking...

for staff use only
–

Rapid Fire. (2 points each.)

Answer each of the questions below in no more than three sentences.

37. What does it mean for a database transaction to be atomic?

38. What's an associative array?

39. What's one reason to use an external stylesheet (via a `link` tag) instead of `style` attributes within a web page?

40. Explain the V and C in MVC.

for staff use only
—

CS50 Associates.

41. (4 points.) Consider the PHP array below.

```
$tfs = [
    ["name" => "Ali", "house" => "Leverett"],
    ["name" => "Joseph", "house" => "Eliot"],
    ["name" => "Lucas", "house" => "Mather"],
    ["name" => "Ore", "house" => "Leverett"]
];
```

Complete the code fragment below in such a way that it outputs a 2-column table with TFs' names and houses. Assume that `$tfs` is in scope. Take care to close the `table` tag that we've opened for you, but no need to add elements like `html`, `head`, `title`, or `body`.

```
<table>
    <tr>
        <td>Name</td>
        <td>House</td>
    </tr>
    <?php

        foreach ($tfs as $tf)
        {
```

for staff use only
—

Sorting students.

42. (2 points.) Consider the web page below.

```
<!DOCTYPE html>

<html>
    <head>
        <title>Sorting Hat</title>
    </head>
    <body>
        <form action="http://www.hogwarts.edu/lottery.php" method="get">
            <input name="name" type="text"/>
            <br/>
            <select name="house">
                <option value=""></option>
                <option value="Gryffindor">Gryffindor</option>
                <option value="Hufflepuff">Hufflepuff</option>
                <option value="Ravenclaw">Ravenclaw</option>
                <option value="Slytherin">Slytherin</option>
            </select>
            <br/>
            <input type="submit" value="Submit"/>
        </form>
    </body>
</html>
```

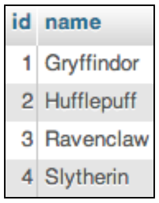
At what URL will a student named Harry find himself if he submits this page's form after inputting his name and selecting Gryffindor as his house?

for staff use only
–

Storing students.

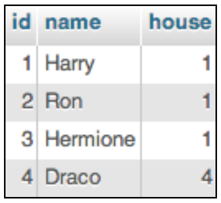
Consider the SQL tables below. At left is a table called `houses`, wherein `id` is a `PRIMARY` key. At right is a table called `students`, wherein `id` is a `PRIMARY` key (that `AUTOINCREMENT`s) and `house` is a "foreign" key (whose values correspond to values for `id` in `houses`).

`houses`



id	name
1	Gryffindor
2	Hufflepuff
3	Ravenclaw
4	Slytherin

`students`



id	name	house
1	Harry	1
2	Ron	1
3	Hermione	1
4	Draco	4

43. (2 points.) In no more than three sentences, propose a data type for `id` in `houses` and justify your choice.

44. (2 points.) In no more than three sentences, propose a data type for `name` in `houses` and justify your choice.

45. (2 points.) In no more than three sentences, propose a data type for `house` in `students` and justify your choice.

46. (1 point.) With what SQL query could we remove Draco from Slytherin (while still allowing him to remain a student)?

47. (1 point.) With what SQL query could we place a student named Luna in Ravenclaw?

48. (2 points.) With what SQL query could we retrieve a result set of rows, each of which includes the name of a student and the name of that student's house?

for staff use only
—

Scrap Paper.

Nothing on this page will be examined by the staff unless otherwise directed in the space provided for some question.