Computer Science 50: Introduction to Computer Science I
Harvard College
Fall 2007

Quiz 1

out of 77 points

Print your name on the line below.

____

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, but nothing else.

Scrap paper is included at this document's end.
Unless otherwise noted, assume that any code herein is in C.

Please circle your section leader's name.

Abe Passaglia

Andrew Berry

Andrew Granoff

Anjuli Kannan

Charlotte Eccles

Chris Stevens

David Haley

David Ramos

Diana MacLean

Doug Lloyd

Emily Parfit

Josh Schwartz

Katie Fifer
graded by Chris Power

Katie Fifer
graded by Katie Fifer

Kelly Heffner

Kristen Lovin

Mike Tucker

Paul Govereau

Rafael Garcia

Roy Shi

Thomas Carriero

Tova Wiener

Yao Yu

for staff use only

final score out of 77

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. (0 points.) Who (should have) won Office 2007 Ultimate?

   a. David J. Malan
   b. David J. Malan
   c. David J. Malan
   d. ~~Nina Han~~ David J. Malan

1. (1 point.) If you employ binary search, about how many pages need you examine maximally in order to find a plumber in an alphabetized phonebook that (amazingly) has 1,048,576 pages?

   a. 1,048,576
   b. 524,288
   c. 1,024
   d. 20

2. (1 point.) The running time of an algorithm in $O(n^2)$ is said to be

   a. logarithmic.
   b. linear.
   c. quadratic.
   d. exponential.

3. (1 point.) Consider the function below.

```
int f(int n)
{
    if (n == 1)
        return 0;
    else
        return (1 + f(n/2));
}
```

   What value is returned by calling `f(16)`?

   a. `3`
   b. `4`
   c. `5`
   d. `6`

for staff use only
points off on this page

–

Let's see how good your memory is.

4. (8 points.) Consider the program below.

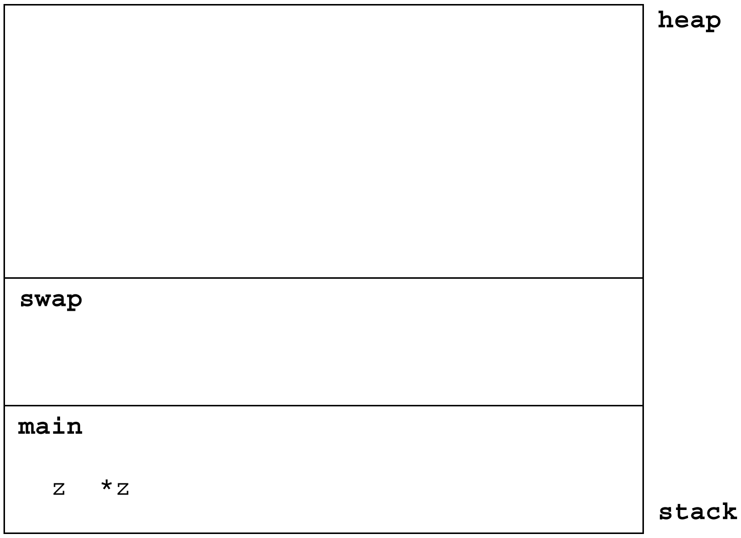
```
#include <stdlib.h>

void
swap(int *a, int *b)
{
    int tmp = *a;
    *a = *b;
    *b = tmp;

    // assume that we have paused execution here
}

int
main(int argc, char * argv[])
{
    int x = 1;
    int *y = (int *) malloc(sizeof(int));
    *y = 2;
    int *z = &x;
    swap(&x, y);
}
```

Suppose that the figure below represents this program's layout in memory. Assume that we have paused (as with `gdb`) execution of this program just before `swap` returns and that `malloc` did not return `NULL`. Without worrying about specific addresses, tell us where each of `x`, `y`, `*y`, `a`, `*a`, `b`, `*b`, and `tmp` can be found in memory, generally speaking, by jotting down each of those eight expressions in the appropriate space below. For clarity's sake, we've spoiled where `z` and `*z` can be found. For symbols that belong in the same general area, you need not worry about their relative ordering (*i.e.*, we could have written `z` to the right of `*z`).



for staff use only

points off on this page

–

O(mega).

5. (4 points.) Consider the pseudocode for Bubble Sort below.

```
Repeat n times:
    For each element i:
        If element i and its neighbor are out of order:
            Swap them.
```

Why is this implementation of Bubble Sort in $\Omega(n^2)$? In a sentence or more, explain (in English or pseudocode) how you could alter this implementation so that it is instead in $\Omega(n)$.

Should have memoized this!

Consider the memo-less implementation of the Fibonacci sequence below.

```
#include <stdio.h>

int
fs(int n)
{
    printf("*");
    if (n == 0)
        return 0;
    else if (n == 1)
        return 1;
    else
        return (fs(n-1) + fs(n-2));
}

int
main(int argc, char * argv[])
{
    int answer = fs(4);
}
```

6. (1 point.) What value is ultimately stored in `answer`?

7. (1 point.) How many asterisks does this program print?

for staff use only
points off on this page

–

This function is so foobar!

8. (6 points.) Consider the function below.

```
int
foobar(int n)
{
    if (n < 2)
        return 1;
    else
        return (n * foobar(n-1));
}
```

Assuming it is called with some non-negative `n`, what mathematical function does `foobar` compute?

In a sentence or more, explain why, for sufficiently large non-negative `n`, `foobar` actually returns negative values.

In a sentence or more, why might `foobar` sometimes segfault?

In the space below, rewrite `foobar` in such a way that it no longer employs recursion. Ensure that your version is functionally equivalent (*i.e.*, given some input, your version and ours should return the same output, our version's potential segfaults aside).

```
int
foobar(int n)
{
```

for staff use only
points off on this page

–

Fun with Tables.

9. (8 points.) Complete the table below by specifying tight bounds for each of the algorithms listed. By tight, we mean that, if some algorithm, on input of size *n*, might involve as many as n^2 operations, you should deem that algorithm in $O(n^2)$; to deem it in, for instance, $O(n^4)$ would be true, but not tight. Similarly, if some algorithm, on input of size *n*, involves at least n^2 operations, you should deem that algorithm in $\Omega(n^2)$; to deem it in, for instance, $\Omega(n)$ would be true, but not tight.

If any of your bounds depends on some assumption, note your assumption.

	O	**Ω**	**assumptions, if any**
Binary Search			
Linear Search			
Merge Sort			
Selection Sort			

10. (10 points.) Complete the table below in such a way that the values in each row are equivalent, even though written in different bases. You may omit leading 0s. For clarity's sake, we've completed the first row you.

base-2	**base-10**	**base-16**
1010	10	A
10		
		10
	128	
		F00
	255	

for staff use only
points off on this page
–

11. (6 points). Complete the table below by specifying the size (on `nice.fas.harvard.edu`), in bytes, of each of the types listed.

	`sizeof`
`char`	
`char *`	
`int`	
`int *`	
`long long`	
`long long *`	

Rapid Fire.

For each of the questions below, we expect an answer of no more than three sentences.

12. (2 points.) Why use recursion to solve some problem if an iterative solution is also possible?

13. (2 points.) What does it mean if some algorithm is in $\Theta(n)$?

for staff use only
points off on this page

–

14. (2 points.) What's the difference between the `.` (*i.e.*, dot) operator and the `->` operator, with respect to `struct`s?

15. (2 points.) Why should we start writing software that is more capable of taking advantage of parallelism?

16. (2 points.) Why is it often possible to recover a file even after it has been "deleted" from a hard drive?

17. (2 points.) What's a memory leak?

for staff use only

points off on this page

–

Makin' Copies.

Consider the lines of code below, where `string` and `GetString` are defined as in CS 50's library. (Recall that `string` and `char *` are synonymous.)

```
string s1 = GetString();
string s2 = s1;
s2[0] = toupper(s2[0]);
```

18. (2 points.) Why, in a sentence or more, does the last line above capitalize both `s1` and `s2`, even though `s2` is a copy of `s1`?

19. (8 points.) In the space below, implement a new function for CS 50's library called `CopyString` that takes as its sole argument a `string`, `s`, and returns a `string`, whereby

```
s != CopyString(s)
```

but

```
strcmp(s, CopyString(s)) == 0
```

assuming `s` is not `NULL`. In other words, informally, `CopyString` should return a true copy of `s`. If pressed for time, you may resort to pseudocode for partial credit.

```
string
CopyString(string s)
{
```

for staff use only

points off on this page

–

LAST ONE!

20. (8 points.) In the space below, write a program that converts not only the contents of an ASCII file but also the name thereof to uppercase. For instance, were you to execute

```
a.out questions.txt
```

your program would create a brand-new file called `QUESTIONS.TXT` in your current working directory, the contents of which (*i.e.*, all characters within) would be identical to those of `questions.txt`, except every alphabetical character in the latter would be capitalized in the former; all other characters would appear unchanged. You may assume that your program will be executed with exactly one command-line argument and that the argument will contain at least one lowercase letter (so that you can actually convert the file's name to uppercase).* Be sure that your program cannot segfault. If pressed for time, you may resort to pseudocode for partial credit.

```
#include <stdio.h>
#include <stdlib.h>

int
main(int argc, char * argv[])
{
```

* If wondering, you may also assume that your program's input and output will be stored on a case-sensitive file system, whereby such "identically" named files as `questions.txt` and `QUESTIONS.TXT` are indeed different files.

for staff use only

points off on this page

–

~~Extra Credit.~~ "Extra Credit."

21. (0 points.) Outline the steps involved in breeding your own super-high-yield, all-weather hybrid strain of wheat. Describe its chemical and physical properties and estimate its impact on world food supplies. Construct a model for dealing with world-wide surpluses. Write your Nobel Prize acceptance speech.

22. (0 points.) In three minutes, a stupid gorilla with a negative I.Q. will be admitted to the room. Teach him to calculate cube roots. You may not use any form of communication.

23. (0 points.) Build an exact 1:1 model of the Great Wall Of China by the end of the examination period. You have been provided under your seat with 5 bricks and a piece of Scotch tape. Extra extra credit: build a 1:1 model of the (former) Berlin Wall using 5000 pounds of sauerkraut and then build David J. Malan using 5000 lines of C.

24. (0 points.) Transform lead into gold. You will find a tripod and three logs under your seat. Show all work including Feynman diagrams and quantum functions for all steps. You have fifteen minutes.

25. (0 points.) This is a three-minute time test. Read everything before doing anything. Put your name in the upper right hand corner of this page. Circle the word "name" in sentence three. Sign your name on the bottom of this page; underneath your name, write "yes, yes, yes!" Put an 'X' in the lower left hand corner of this page. Draw a equilateral triangle around the 'X' you just put down. On the back of this exam multiply 703 and 668. Loudly call out your name when you get to this point. If you think you have followed directions carefully to this point call out "I have!" Punch three small holes in the top of this paper. If you are the first person to get this far, call out "I am the first person to this point! I am leading in following directions!" On the front of this exam now add 8950 and 9850. Put a circle around your answer and put a square around the circle. Now that you have finished reading carefully, do only sentence three.[†]

[†] Needless to say, we will make fun of you while grading if we see your name atop this page. Oh and we'll also make fun of you if we see eraser marks atop this page.

Scrap Paper.

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

Scrap Paper.

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