Computer Science 50 **Quiz 0 Practice**
TF: Samuel Green **October 2015**
samuelgreen@college.harvard.edu

N.B. This practice quiz is my best guess at what a quiz might look like. I haven't seen Quiz 0, so no guarantees! I'd recommend taking this quiz in the same ∼90 minute controlled setting as the real quiz, but also be warned that it may be shorter or longer than the real quiz. It would also be most helpful after you've prepared your one page front-and-back "cheatsheet".

1 Lightning Round

1.1 Short Answer

(A) What's decimal 51 in binary? How about hexadecimal?

(B) How many bytes does an `int` occupy? How many bits?

(C) What does it mean to compile code? What are the steps?

(D) Who's the best TF in all the land?

1.2 0 or 1

Please answer the following questions, justifying your answer in the space provided.

(A) A string is really an array.

`TRUE / FALSE`

(B) The `float` datatype can store infinitely precise decimal numbers.

`TRUE / FALSE`

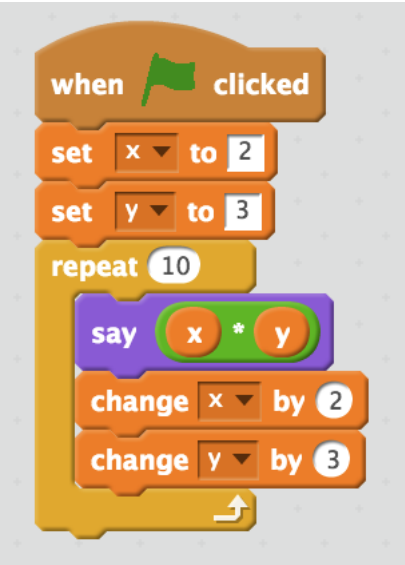
(C) Binary search is always better than linear search.

`TRUE / FALSE`

(D) Running time is all that matters when considering the performance of an algorithm.
`TRUE / FALSE`

2 Fun with Scratch

Take a look at the Scratch script below:



In the space below, write a C program that creates "equivalent" output, where `say` is equivalent to `printf` .

```
#include <stdio.h>

int main(void){
```

3 Things that Divide Us

I really enjoy random mathematical facts. As part of my fascination, I have a C program that really, really needs to know some stuff about multiples. Can you help me out?

3.1 Three's Plenty

Complete the function below so that it returns `true` if `n` is a multiple of three, `false` otherwise.

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

3.2 More's a Crowd

Turns out, I need a more general function after all.

Complete the function below so that it returns true if `n` is a multiple of `f`.

```
bool isMult(unsigned int n, unsigned int f){
```

4 Uppers and Lowers

You're working on a super cool program called initials, which you hope will let users easily find out what their initials are. For some reason, though, you can't remember the header file in which some handy character functions live, so you've decided to write them again.

Fill in the functions below, maximizing code reusing whenever possible. You may assume that all necessary headers have been included (except the one containing these functions!).

```
bool isupper(char c){









}

bool islower(char c){









}

bool isalpha(char c){









}
```

5 What you doin' in the club on a Thursday?

(A) I can't remember the upper and lower bounds on the running times of my favorite algorithms. Remind me by filling in the following table based on algorithms we have seen in class. Be careful not to reuse any algorithm!

Algorithm	Ω	O
Linear Search		
Merge Sort		
	n	n^2
	n^2	n^2
Insertion Sort		

(B) What're the lower and upper bound on the running time of binary search? Why does it have those bounds?

(C) Would we ever choose an algorithm with a higher upper-bound on its running time over a similarly effective algorithm with a lower upper-bound? Why or why not?

6 Stop Copying Me!

Consider the program below:

```
#include <stdio.h>
#include <string.h>
#include <ctype.h>

#define ALPHA 26

void change(char* str){
    for (int i = 0, n = strlen(str); i < n; i++){
        if (islower(str[i]))
            str[i] = (str[i] % 'a' + 1) % ALPHA + 'a';
        else if (isupper(str[i]))
            str[i] = (str[i] % 'A' + 1) % ALPHA + 'A';
    }
    printf("%s\n", str);
}

int main(void){
    char myString[] = "Sam's section rockz!";

    change(myString);
    printf("%s\n", myString);
}
```

(A) What does the program above output?

(B) What does `#define` do, and why does it appear in the program above?

(C) What does it mean for a function to have a "side effect"?

7 Pointing is Rude

In the code below, you may assume the following values:

&w	0x123
&x	0x4ff
&y	0xf77
&z	0x288

```
#include <stdio.h>

int main(void){
  // comment 1
  int w = 2, x = 3, y = 4, z = 5;
  int* wp = &w;
  int* xp = &x;
  int* yp = &y;
  int* zp = &z;

  // comment 2
  wp = yp;
  yp = zp;
  zp = xp;

  // comment 3
  x = x*2;
  y = x-*yp;
  z = x + y;
  w = *wp + *zp;
  printf("%p\n", yp);
  printf("%i\n", w);
}
```

(A) In the space to the right of the code, please draw two box-and-arrow diagrams (like we did in section!): one showing the program's variables as it "passes through" comment 2, and one showing the program's variables as it "passes through" comment 3.

(B) What does the program output?

8 Up Periscope!

(A) Define scope.

(B) What does the code below output?

```
void func(void){
  int x = 100;
  printf("%i\n", x);
}
int main(void){
  int x = 10;
  func();
  printf("%i\n", x);
}
```

(C) Does it matter what names variables are given? Why and why not?

9 Inbox

Respond to the following emails:

(A) Dear TF:

I think something is going wrong with my compiler! I'm certain that my code is correct, but I keep seeing `Segmentation Fault` every time I run my program. What's going on?!?@@@? Please help <3333333

Sincerely, certain this is not an error.

(B) Dear TF:

why do you keep on taking style points off on my problem sets? If code all looks the same to the computer, why does style even matter?

Sincerely, unstylish

(C) Dear TF:
I really don't get the point of writing functions. Our programs are short, so I might as well just put the code in again! Can you help me understand the benefits?

Thx, functional