Minor programming

Programming 1 (C)

May 28th, 2014

You can earn up to 65 points for this exam.
To pass the exam, you need to earn at least 33 points.

The time allowed is 120 minutes. Once started, you are not allowed to leave the room. Please turn off your cell phone. Besides a pen or pencil, nothing else is allowed to be on your table during the exam.

Good luck!

Name:

Student ID:

Autograph:

Multiple-choice questions (26 points)
The first 13 questions are multiple-choice. For each of these questions, there is exactly one correct answer. Circle the answer you think is correct. Each correct answer rewards you with 2 points; each wrong answer deducts 1 point. Questions left unanswered do not give or take points. (In other words, it might not be a good idea to guess answers.)

1. Which of the following is a pointer?

   A `char`
   B `int`
   C `string`
   D `float`

2. How many bits do you minimally require to represent the decimal number 7 in binary?

   A 3
   B 4
   C 7
   D 8

3. What is the running time of selection sort?

   A $\Omega(n)$
   B $\Omega(n^2)$
   C $\Omega(1)$
   D $\Omega(\log n)$

4. What is the decimal representation of the hexadecimal number 20?

   A 16
   B 31
   C 32
   D 34

5. Which program compiles your C code?

   A `gdb`
   B `std=c99`
   C `make`
   D `clang`

6. While all four of the main definitions below correctly compile, we recommend only one of them in our style guide. Which one?

   A
   ```
   int main(int argc)
   ```
   B
   ```
   main(void)
   ```
   C
   ```
   int main(void)
   ```
   D
   ```
   main(int argc, char *argv[])
   ```

7. Which header should you include before you can use the `fopen` function?

A `stdlib.h`
B `stdio.h`
C `cs50.h`
D `string.h`

8. What will the following code, when compiled and run, print to the screen?

```
#include <stdio.h>

int main(void)
{
      int a = 7;
      int b = 7;

      if (a == b)
            a = a + b;
      else
            b = a + b;
            a = 0;

      printf("%d, %d", a, b);

      return 0;
}
```

A `7, 7`
B `7, 14`
C `14, 7`
D `0, 7`

9. Consider the following two statements. Are they true or false?
1: "Each C file must contain a `main` function."
2: "`argv` contains the arguments the user typed when invoking a program."

A both 1 and 2 are true
B both 1 and 2 are false
C 1 is true and 2 is false
D 1 is false and 2 is true

10. Assume that `var` is a `float` variable. How can I correctly tell `printf` to print that variable with exactly four decimals?

A `printf("%4.d", var);`
B `printf("%4.f", var);`
C `printf("%.4f", var);`
D `printf("%.4d", var);`

11. What is the worst-case performance of bubble sort?

A $O(1)$
B $O(n!)$
C $O(n^2)$
D $O(n)$

12. Consider the following bit sequence: 0010 1001.
If this sequence represented an `int`, what would that `int`'s decimal value be?

A 3
B 39
C 41
D 47

13. Consider a running program that encounters the following line of code.

```
int z = 3 + 7 / 2;
```

What is the resulting value stored in `z` when done executing this line?

A 5
B 6
C 6.5
D 7

Short-answer questions (27 points)
Please answer the following open questions with exactly one sentence or one line of code. Answering these incorrectly does not deduct any points, so try to answer every one of them, even if (in part) unsure. No need to explain your answers.

14. (2 points.) Write a line of code that takes the remainder of the division of integers `a` by integer `b`, and stores it in integer `c`. You may assume that both `a` and `b` are greater than zero.

```
int c =                ;
```

15. (2 points.) What is a `continue` statement? What does it do?

16. (2 points.) What is a `return` statement? What does it do?

17. (1 point.) Suppose we're trying to order an array of integers. We've decided to go with bubble sort. Could you name a faster alternative to bubble sort?

18. (3 points.) Suppose we declare and initialize a float such as the following:
`float f = 0.41;`
However, when we print the first ten decimals of `f`, we get the following:
`0.4099999964`
Why does this happen?

19. (2 points.) Which header do I need to `#include` before I can use the `round` function?

20. (1 point.) Does the following program compile without errors? "Yes" or "no" will suffice, no additional explanation needed.

```
#include <stdio.h>

int main(void) {
     printf("hello world!");
}
```

21. (4 points.) Creating an executable program out of your source code actually involves a number of steps. (Four to be precise.) Give the names of these steps in the order of their execution.

22. (2 points.) Assume we want to generate a random number. Write a line of code that seeds our random generator with the current time. Use only one line of code. You may assume that all required headers are `#include`d.

23. (3 points.) Having previously seeded the random generator, now declare a new `int` and assign a random integer value between 1 and 10 exclusive to that integer. Use only one line of code.

24. (4 points.) Suppose I have the following two binary values that represent integers: 01010011 and 00011101. When added together (e.g. +), what is the result in decimal?

25. (1 point.) Which function from the standard C libraries could I use to convert a `string` into a `float`?

Programming questions (12 points)
The remaining questions are programming questions. Answering these incorrectly does not deduct any points, so try to answer every one of them, even if (in part) unsure!

26. (7 points.) Finish the following program so that it sorts, from lowest to highest value, the contents of array `a`. Write your code directly in `main`; do not declare a separate function. You do not need to `#include` any necessary header files. Your implementation should still work even if I later decide to change the order of the contents in `array`, but you may assume that `array` will always contain the same elements (e.g. numbers 0 through 9) and never any duplicates.

```
int main(void) {
      // the array to sort!
      int a[10] = { 7, 4, 1, 3, 6, 9, 0, 2, 5, 8 };

      // insert your code here!









      // all done!
      return 0;
}
```

27. (5 points.) Recall that `strlen` is a function that takes a `string` as input and returns the length of that `string` in the form of an `int`. Without calling any function, implement `strlen` yourself. If `s` is `NULL`, you should return `0`.

You do not need to `#include` any headers, even if you decide to use something that would require one (or more) of them.

```
int strlen(char* s)
{









}
```