Minor programming

Programming 1 (C)

March 26th, 2014

You can earn up to 80 points for this exam.
To pass the exam, you need to earn at least 41 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 (30 points)
The first 15 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. How many bits do you require to represent the decimal number 9 in binary?

   A 4
   B 5
   C 6
   D 8

2. Which header should you include before you can use the `printf` function?

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

3. 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);`

4. Consider a running program that encounters the following line of code.
   `int tmp = 7 / 2;`
   What is the resulting value stored in `tmp` when done executing this line?

   A 3
   B 3,5
   C 4
   D 7

5. How many different encryptions are possible using Caesar's cipher?

   A 1
   B 13
   C 26
   D 26!

6. What is the running time of selection sort?

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

7. Consider the following bit sequence: 0110 0111.
If this sequence represented an `int`, what would that `int`'s decimal value be?

A 5
B 39
C 71
D 103

8. Convert the decimal number 41 to a hexadecimal number.

A 1A
B 27
C 29
D 2F

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

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

10. What is the difference between `char*` and `string`?

A a `char*` is a pointer and a `string` isn't
B a `string` leaks memory and a `char*` doesn't
C a `char*` can point to only one character, whereas `string` can point to multiple characters
D none of the above

11. Which program compiles your C code?

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

12. Consider the following two statements. Are they true or false?
1: "`main` must always return an `int`."
2: "`argc` 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

13. How many values are present in the non-extended ASCII table?

A 26
B 62
C 128
D 256

14. Say that we successfully compile and run a program that, at one point, includes a call to the function `fopen`. Upon inspection, that function then appears to return `NULL`, rather than a file pointer. What does that indicate?

A the required header file (`stdio.h`) was not included in our source code
B the file we tried to open is in another directory
C the file we tried to open does not exist
D the file we tried to open could not be opened

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

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <cs50.h>

int main(void)
{
      string a = "world";
      string b = "world";

      printf("hello ");

      if (a == b)
            printf("%s ", a);
            printf("friend ");

      printf("goodbye ");
      return 0;
}
```

A `hello world`
B `hello world friend goodbye`
C `hello goodbye`
D `hello friend goodbye`

Open questions (50 points)
The remaining questions are open questions. Their point values are printed alongside them. Answering these incorrectly does not deduct any points, so try to answer every one of them, even if (partly) unsure!

16. (2 points.) What does it mean when a function is recursive?

17. (2 points.) Suppose you've written a program that uses an integer called `n`, and that your compilers throws the following error at you:
`"Use of undeclared identifier 'n'"`
What is the problem and how can you fix it?

18. (4 points.) Suppose you have a program in which you want to swap two `int`'s. Finish the following function, using only pointers, that performs the swap.

```
void swap(int *a, int *b)
{




}
```

19. (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?

20. (5 points.) Recall that `strlen` is a function that takes a `string` (or, `char*`) as input and returns the length of that `string`. 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)
{









}
```

21. (8 points.) Recall that `atoi` is a function that takes a `char*` as input and returns an `int`. Without using any function other than `strlen`, implement `atoi` below. If `s` is `NULL` or contains any symbol other than `0` through `9`, your function should return `0`. This means that your `atoi` function only needs to work for positive `int`s, as the – symbol is not within the range of `0` through `9`.

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

```
int atoi(char* s)
{









}
```

22. (2 points.) What is a `continue` statement? Be precise in your explanation, but do not use more than 50-or-so words.

23. (2 points.) What is a `break` statement? Be precise in your explanation, but do not use more than 50-or-so words.

24. (2 points.) What is a `return` statement? Be precise in your explanation, but do not use more than 50-or-so words.

25. (6 points.) Suppose we have the following program. (Note that, for simplicity, we omitted the necessary `#include`s.)

```
int main(void)
{
      int n = GetInt();

      if (n == 1)
            printf("cat\n");
      else if (n >= 2 && n < 5)
            printf("dog\n");
      else if (n == 7)
            printf("giraffe\n");
      else
            printf("elephant\n");

      return 0;
}
```

Rewrite this program using a `switch` statement. You're not allowed to use the `if` and `else` keywords. Make sure your program behaves exactly the same as ours. You need not `#include` the necessary header files.

```
int main(void)
{
      int n = GetInt();









      return 0;
}
```

26. (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.

27. (2 points.) What are the only two correct ways, according to our style guide, to define `main`?

For the remaining questions, consider the following program below:

```
int main(void)
{
      int x = GetInt();
      int y = GetInt();
      int m[x][y];
      srand(time(NULL));









      return 0;
}
```

28. (2 points.) What does that call to `srand` do? Be sure to also explain why we would want to call this function with an argument of `time(NULL)`.

29. (2 points.) It seems we forgot to `#include` some necessary header files for this program to compile. Write lines above the `main` function to fix this. You may not `#include` headers you do not actually need for this program to compile.

30. (4 points.) Write code to fill the matrix (which we called `m`) with random `int` values between `0` and `10`, exclusive. Be sure to fill the entire matrix without threading out of your matrix's boundaries! In addition, you are only allowed to use exactly one `for` loop and no `while` loops, and you may not `#include` any more headers than were required to answer question 29.