

Evaluation quiz correction

Evaluation Quiz: Evaluation #2

Date: 2022-05-20

Status: Done

Duration: 15 minutes

Score: 100.0%

"I don't know": 0

Success: 16

Fail: 0

Responses

0. What are the different steps to form an executable file from C source code?

Score: 1.0

- ☐ Interpretation, compilation and assembly
- ☒ **Preprocessing, compilation, assembly, and linking**
- ☐ Interpretation, assembly and compilation
- ☐ Compilation and linking
- ☐ Preprocessing and compilation
- ☐ I don't know

*1. Without context, on Ubuntu 14.04 LTS, **write** is a(n) ...*

Score: 1.0

(please select all correct answers)

- ☒ executable
- ☒ system call
- ☐ library call
- ☐ game
- ☐ I don't know

2. What is the correct combination of `oflags` used to open a file with the mode write only, create it if it doesn't exist and append new content at the end if it already exists?

Score: 1.0

- ☐ `O_WRONLY`
- ☐ `O_WRONLY | O_CREAT | O_EXCL`
- ☒ `O_WRONLY | O_CREAT | O_APPEND`
- ☐ `O_RDWR | O_CREAT | O_APPEND`
- ☐ I don't know

3. When I am using `O_WRONLY | O_CREAT | O_APPEND` -> the `|` are bitwise operators.

Score: 1.0

- ☒ True
- ☐ False
- ☐ I don't know

4. `~ 0x98 = ?`

Score: 1.0

- ☐ `0x66`
- ☒ `0x67`
- ☐ `0x68`

- ☐ I don't know

5. What is 98 in base 16?

Score: 1.0

- ☒ 0x62
- ☐ 0x98
- ☐ 0x96
- ☐ I don't know

6. Choose the line of code to replace the comment below so the function prints a given string without a lower or uppercase c.

Score: 1.0

```
>>>def no_c_print(s):  
...     new_string = ''  
...     for character in s:  
...         # REPLACE THIS LINE  
...         new_string += character  
...     print(new_string)  
...  
>>> no_c_print("Characters")  
>>>haraters
```

- ☒ if character not in 'Cc':
- ☒ if character != 'c' and character != 'C':
- ☐ if character != "cC"
- ☐ I don't know

7. The following code gives this incorrect output.

Score: 1.0

Which of the following statements about what is causing the error is true? (select all valid answers)

[illegible]

- ☒ `j` never increments so it will always be less than 10
- ☐ `j` is always equal to `i` so the loop will never end
- ☒ `j` never increments so it is always going to print 0
- ☐ I don't know

8. `0x13 << 1 = ?`

Score: 1.0

- ☐ 0x13
- ☐ 0x12
- ☒ 0x26
- ☐ 0x4C
- ☐ I don't know

9. Choose a statement that would complete the function that returns a string made up of `+` `n` number of times (assuming `n > 0`).

Score: 1.0

```
>>> def print_plus(n):
...     # REPLACE THIS LINE
...
>>> print_plus(3)
>>> '+++'
>>> print_plus(4)
>>> '++++'
```

- ☒ `return n* '+'`
- ☒ `return '+'*n`
- ☐ `return '+'n`
- ☐ `return+n+n+n`
- ☐ I don't know

10. What does this command line print?

Score: 1.0

```
>>> a = "Hello, world!"  
>>> print(a[:5])
```

- ☒ Hello
- ☐ world!
- ☐ orld!
- ☐ I don't know

11. What is wrong with the following code?

Score: 1.0

```
int n = 0;  
int array[5];  
int i = 5;  
  
array[n] = i;
```

- ☒ Nothing is wrong
- ☐ It is impossible to declare the variable `array` this way
- ☐ The array `array` is not entirely initialized
- ☐ It is not possible to access `array[n]`
- ☐ I don't know

12. In the `main.c` file, on what line is the first error that the compiler returns?

Score: 1.0

Look at the following code.

```
carrie@ubuntu:/debugging$ cat main.c
```

```

#include <stdio.h>

/**
 * main - debugging example
 * Return: 0
 */
int main(void)
{
    char *hello = "Hello, World!";

    for (i = 0; hello[i] != '\0'; i++)
    {
        printf("%c", hello[i]);
    }

    printf("\n");

    return (0);
}
carrie@ubuntu:/debugging$
carrie@ubuntu:/debugging$ gcc -Wall -Werror -Wextra -pedantic main.
c
main.c: In function 'main':
main.c:11:7: error: 'i' undeclared (first use in this function)
    for (i = 0; hello[i] != '\0'; i++)
        ^
main.c:11:7: note: each undeclared identifier is reported only once for each function
it appears in
main.c:9:8: error: variable 'hello' set but not used [-Werror=unused-but-set-variable]
    char *hello = "Hello, World!";
        ^
cc1: all warnings being treated as errors
carrie@ubuntu:/debugging$

```

- ☐ 9
- ☒ 11
- ☐ 7
- ☐ I don't know

13. What is `0b001010010` in base10?

Score: 1.0

- ☐ 81
- ☒ 82
- ☐ 83
- ☐ 84
- ☐ I don't know

14. What does this print?

Score: 1.0

```
>>> print("My favorite line of {} is {:d}.".format("The Zen of Python", 11))
```

- ☒ My favorite line of The Zen of Python is 11.
- ☐ My favorite line of T is 1.
- ☐ My favorite line of The Zen of Python is 1.
- ☐ I don't know

15. What is the `unistd` symbolic constant for the standard input?

Score: 1.0

- ☒ `STDIN_FILENO`
- ☐ `STDOUT_FILENO` ☐ `STDERR_FILENO`
- ☐ I don't know

