

Unit 07_010 – C Functions

1. Why is C a "low level, high level language?"
2. What does the following code mean in C? How many bytes would be needed (remember to include the null character)

```
char * s = "Hello"
```

3. Write the assembler code in the .data section that would create the equivalent of `char * s = "Hello"`. It was not covered explicitly in the video, but you should be able to figure out the answer.
4. What does the puts command do?
5. What is %d in the printf function?
6. What is %x in the printf statement?
7. How can the %x format specifier be used so that it is clear the output is in hex?

Unit 07_020 – Farewell, start

8. What command may be used for linking to replace the ld command?
9. What label needs to be changed in order to use gcc as a linker?
10. What else can gcc with assembler in addition to linking?

11. What is the command you would use to assemble and link your program?
 12. What is `gasm`?
 13. In software development, do you need to dig through documentation yourself, or can you depend on Google and AI to do the searching for you?
-

Unit 07_030 – puts

Video Length 8:00

14. What does the `puts` function do in C? How many arguments does `puts` take?
 15. How are parameters passed to functions in x86-64 assembler?
 16. What is the order of register use for sending arguments to function calls?
 17. Assume the a variable called `sentence` is defined as a null terminated string. Write the code needed to call `puts` from assembler. Do not worry about the return value.
 18. What does the `$` mean in `movq message, rdi`?
-
-

If you have any lingering questions or problems, please write them here or see me.



Draft