**Quiz - Paper 2**

**Instructions**

**- This quiz contains 25 multiple-choice questions.**

**- Select the best answer for each question.**

**- Time allowed: 30 minutes.**

**bleedecode bleedecode bleedecode bleedecode bleedecode bleedecode bleedecode bleedecode bleedecode bleedecode bleedecode bleedecode bleedecode bleedecode bleedecode bleedecode bleedecode bleedecode bleedecode bleedecode bleedecode bleedecode bleedecode bleedecode bleedecode bleedecode bleedecode bleedecode**

1. What is the difference between malloc() and calloc() in C?
   * A) malloc() allocates uninitialized memory while calloc() initializes memory to zero
   * B) malloc() allocates for one block while calloc() allocates for arrays
   * C) malloc() is faster than calloc() but uses more memory
   * D) There is no difference except parameter format
2. In C, what does the volatile keyword indicate?
   * A) The variable can be changed by external factors outside program control
   * B) The variable should be stored in fast access memory
   * C) The variable cannot be optimized away by the compiler
   * D) Both A and C
3. What does this C code snippet do?

int x = 5;

int y = x++ + ++x;

* + A) Sets y to 11
  + B) Sets y to 12
  + C) Causes undefined behavior
  + D) Compilation error

1. What's the output of: printf("%d", sizeof(int[10])/sizeof(int))?
   * A) 10
   * B) 40
   * C) Depends on the compiler
   * D) Memory address
2. What problem does this C macro solve?

#define max(a, b) \

({ typeof (a) \_a = (a); \

typeof (b) \_b = (b); \

\_a > \_b ? \_a : \_b; })

* + A) Avoids double evaluation side effects
  + B) Makes max work with any data type
  + C) Creates thread-safe maximum function
  + D) Both A and B

1. When your IDE autocompletes your curly braces, it's secretly thinking:
   * A) "I've prevented 17 syntax errors you would have made in the next hour"
   * B) "I'll still let you stare at this code for 20 minutes before you notice the semicolon is missing"
   * C) "Look at me doing the bare minimum and expecting gratitude"
   * D) "My human thinks they're coding but I'm doing all the real work here"
2. What's the real meaning of "RTFM" in technical support?
   * A) "I've never read the manual either but want to sound superior"
   * B) "This question is asked so often there's a shrine dedicated to it in the documentation"
   * C) "The answer is on page 573 of a 1200-page PDF I haven't opened since 2007"
   * D) "Let me Google that and pretend I knew all along"
3. When a developer says they'll fix a bug "when I have time," they actually mean:
   * A) "When the heat death of the universe makes this bug irrelevant"
   * B) "When the bug becomes sentient and fixes itself"
   * C) "When I've exhausted all possible excuses not to fix it"
   * D) "When three more customers report it and my manager questions my existence"
4. What happens when the product manager says "just one more quick feature before release"?
   * A) Time itself warps to create a reality distortion field where "quick" means "three more sprints"
   * B) Developers spontaneously develop the ability to age a decade in a week
   * C) The CI/CD pipeline achieves sentience and files for emotional distress
   * D) The feature is actually implemented quickly, and everyone is suspicious
5. What does "I've optimized our codebase" usually translate to?
   * A) "I've replaced readable code with arcane incantations that will summon debugging demons"
   * B) "It's 0.02% faster but completely unmaintainable"
   * C) "I've deleted all the comments because real programmers don't need them"
   * D) "I've rewritten someone else's perfectly functional code to match my preferences"
6. If Professor [X] says "Pointers are the foundation of C," the correct interpretation is:
   * A) "Your dreams will be haunted by memory leaks for the next semester"
   * B) "80% of you will submit segmentation faults instead of programs"
   * C) "I enjoy watching the exact moment your confidence leaves your body during labs"
   * D) "All of the above, and I'm not even sorry about it"
7. The half-life of free pizza in a programming lab is measured in:
   * A) Nanoseconds multiplied by the distance to the announcement email server
   * B) The time it takes for the last person wearing headphones to notice
   * C) Planck time units — it essentially disappears instantly
   * D) The compile time of your largest project divided by the number of hungry interns
8. Complete the equation: Debug time = (Hours spent writing the code) × (Hubris level) ÷ ?
   * A) Number of rubber ducks on your desk
   * B) Cups of coffee consumed while writing said code
   * C) Characters in your error message that actually help
   * D) Useful comments you didn't bother to write
9. What's the most effective way to name C pointers?
   * A) p\_thing, pp\_thing, ppp\_thing until your keyboard breaks from overuse of the p key
   * B) thing\_ptr, except when you forget and use ptr\_thing, creating naming conventions chaos
   * C) Whatever cryptic abbreviation will confuse your future self the most
   * D) Single letters that give absolutely no hint about what they point to
10. Which debugging technique do C programmers fear admitting they use most?
    * A) Adding 500 printf statements and slowly removing them one by one
    * B) Staring intensely at the screen hoping the bug reveals itself out of intimidation
    * C) Explaining the problem to an inanimate object while colleagues silently judge
    * D) Giving up, rewriting the entire function, and creating three new bugs
11. True or False: Adding more comments to your C code makes the program run faster.
    * A) True, the compiler feels appreciated and optimizes out of gratitude
    * B) False, but it helps you remember what your code does five minutes after writing it
    * C) True, but only if your comments include compliments to the CPU
    * D) False, excessive comments create a text burden that weighs down execution speed
12. What's the technical term for C code that compiles with zero warnings?
    * A) Mythological artifact
    * B) Statistical impossibility
    * C) Compiler Stockholm Syndrome
    * D) Suspiciously broken but in ways we don't understand yet
13. If you dream about buffer overflows, does your brain crash?
    * A) Yes, that's literally what waking up disoriented is
    * B) No, the human brain implements dynamic memory allocation
    * C) Only if you've exceeded your daily stack of pancakes
    * D) It depends on whether your subconscious has garbage collection
14. When is it appropriate to use goto in modern C code?
    * A) Only when you want to make senior developers cry
    * B) When you're writing a compiler or operating system kernel
    * C) In very specific error handling cases that you'll have to defend in every code review
    * D) When you want to ensure job security through code no one else will touch
15. What happens if you run classic C code on a quantum computer?
    * A) All your uninitialized variables simultaneously contain every possible value
    * B) Pointers become so uncertain that even observed memory can't be trusted
    * C) Schrödinger's buffer: it's both overflowed and secure until measured
    * D) NULL becomes a philosophical concept rather than a memory address
16. If you code C with your eyes closed, why do segmentation faults still occur?
    * A) Because pointers can sense fear even without visual confirmation
    * B) Memory allocation failure is a state of mind, not a visual experience
    * C) The bugs are coming from inside your muscle memory
    * D) Your keyboard has learned your most common mistakes and makes them automatically
17. What's the correct ritual before running a C program for the first time?
    * A) Preparing a sacrificial backup of all your important files
    * B) Writing your will and testament in case of catastrophic memory corruption
    * C) Placing valgrind in a circle of salt around your computer
    * D) Repeatedly muttering "it's just a segfault, it's just a segfault" to build emotional resilience
18. How does commenting your C code in Klingon affect maintainability?
    * A) Increases it, as only the worthy will have the courage to modify your battle-tested code
    * B) Decreases it, unless your team consists entirely of Star Trek convention attendees
    * C) Creates a temporary causality loop where you become your own legacy code maintainer
    * D) Results in mandatory team-building exercises centered around Klingon language classes
19. What actually causes most "array index out of bounds" errors?
    * A) Off-by-one errors caused by the universal confusion about whether arrays start at 0 or 1
    * B) The compiler secretly adding or removing elements when you're not looking
    * C) Your absolute conviction that you "definitely allocated enough space for everything"
    * D) Cosmic rays flipping bits in your loop counters, obviously
20. How many C header files can a programmer include before their code achieves consciousness?
    * A) One more than whatever your current project has
    * B) #include <limits.h> has the answer, appropriately
    * C) It's inversely proportional to the programmer's understanding of each header
    * D) None, it's .c files all the way down