**Quiz - Paper 4**

**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 happens when you run this C code? printf("VTU" + 1);**
   * A) Prints "TU" (string pointer arithmetic)
   * B) Segmentation fault (core dumped)
   * C) Prints "VTU1"
   * D) Compiler error: cannot add integer to string literal
2. **What will this loop print?**

int i = 0;

for(; i < 5; i++) {

if(i == 2)

continue;

printf("%d ", i);

}

* + A) 0 1 3 4
  + B) 0 1 2 3 4
  + C) 1 2 3 4 5
  + D) Infinite loop (obviously)

1. **What's the value of x after this code?**

int x = 5;

int y = x++;

int z = ++x;

x += y + z;

* + A) 19
  + B) 18
  + C) 17
  + D) A number that only makes sense after 3 energy drinks

1. **Why is this code problematic?**

int\* get\_number() {

int x = 10;

return &x;

}

* + A) Returns pointer to local variable that goes out of scope
  + B) The function name doesn't have enough underscores
  + C) Integer pointers are so last semester
  + D) It's missing 17 unnecessary typecasts

1. **What's the output of this masterpiece?**

#define SQUARE(x) x\*x

int main() {

int result = SQUARE(3+2);

printf("%d", result);

return 0;

}

* + A) 25
  + B) 11
  + C) 13
  + D) Your professor's disappointment

1. **What happens when you run this code snippet?**

char \*ptr = malloc(10);

strcpy(ptr, "VTU Computer Science Department");

printf("%s", ptr);

free(ptr);

* + A) Buffer overflow: preparing you for real-world programming
  + B) It prints the string then crashes spectacularly
  + C) Undefined behavior: the program might work, explode, or steal your girlfriend
  + D) The string gets truncated to "VTU Comput"

1. **What's the biggest difference between malloc() and calloc()?**
   * A) calloc() initializes memory to zero; malloc() leaves it uninitialized
   * B) calloc() sounds cooler in lab reports
   * C) malloc() is faster but causes more debugging tears
   * D) One function judges your coding skills silently; the other judges openly
2. **How many ways can you corrupt memory in C?**
   * A) More than the number of students who've failed this course
   * B) Exactly 42
   * C) It's a trick question - all C programs eventually corrupt memory
   * D) *(int*)0 = 0; // Answer not found
3. **What does "dangling pointer" mean in C?**
   * A) A pointer that refers to memory that has been freed
   * B) A pointer with existential doubt about its purpose
   * C) The state of every student during a C programming exam
   * D) A pointer that's hanging on to its last bit of sanity
4. **What's the best way to find the memory leak in your program?**
   * A) Use Valgrind or similar memory analysis tools
   * B) Add more memory to your computer until the problem goes away
   * C) Ask the lab assistant who will also be confused
   * D) Restart your computer and pretend it never happened
5. **The relationship between sleep and debugging ability is:**
   * A) Inversely proportional: less sleep = more bugs = less sleep
   * B) Directly proportional: sleeping on the keyboard produces the best code
   * C) Exponentially decaying: effectiveness drops faster than VTU attendance after week 3
   * D) Undefined: like most pointer arithmetic in your assignments
6. **What's the most accurate way to predict when your C assignment will be finished?**
   * A) Take your initial estimate and multiply by π
   * B) Count the number of semicolons, divide by your will to live
   * C) Add one day for each pointer in your code
   * D) You'll finish approximately 30 minutes after the deadline
7. **How many VTU students does it take to fix a segmentation fault?**
   * A) One to add 500 printf statements and one to interpret the results
   * B) Three - one to code, one to Google, one to pray
   * C) None - they'll change their major instead
   * D) The entire lab section plus one StackOverflow expert
8. **What happens when your code passes all test cases on the first try?**
   * A) You wake up from your dream
   * B) The external examiner becomes suspicious and fails you anyway
   * C) You frantically check if you submitted the right file
   * D) The universe creates a new bug to maintain cosmic balance
9. **What's the difference between first-year and second-year programmers debugging C code?**
   * A) First-years blame the compiler; second-years blame the professor
   * B) First-years don't know what pointers are; second-years wish they still didn't
   * C) First-years copy code from the internet; second-years know which parts to copy
   * D) First-years panic when seeing errors; second-years expect them
10. **If you comment your code thoroughly, what's most likely to happen?**
    * A) Your comments will eventually lie as the code changes but they don't
    * B) You'll spend more time commenting than coding
    * C) No one will read them anyway, including future you
    * D) All of the above, in that exact depressing order
11. **What's the most effective way to summon bugs into existence?**
    * A) Say "My code works perfectly" right before the demo
    * B) Submit your assignment 5 minutes before the deadline
    * C) Tell your friend "I just need to add one more feature"
    * D) Remove all your printf debugging statements because "it works now"
12. **What does a C programmer say when feeling overwhelmed?**
    * A) "I'm suffering from pointer decay"
    * B) "My stack has overflowed"
    * C) "I've malloc'd more than I can free()"
    * D) "seg fault (core dumped)"
13. **What's the proper sacrifice to make before your C programming viva?**
    * A) A circle of Red Bull cans and torn assignment papers
    * B) Your social life for the past month
    * C) The last 3 chapters you never understood
    * D) Your sanity, one segmentation fault at a time
14. **What's the official name for the phenomenon where code works in the lab but fails during evaluation?**
    * A) Professor's Paradox
    * B) The Demo Day Curse
    * C) Heisenbug: the act of observing the code changes its behavior
    * D) Murphy's Law of Academic Demonstration
15. **What's the actual difference between AI and a VTU C programming student?**
    * A) AI learns from its mistakes
    * B) AI can process more than one error message without crying
    * C) AI doesn't blame the compiler for logical errors
    * D) AI hasn't experienced the emotional damage of forgetting a semicolon
16. **What's the correct measurement unit for C program quality at VTU?**
    * A) Segfaults per function
    * B) Minutes spent explaining why it doesn't work
    * C) Number of last-minute fixes that caused new problems
    * D) Ratio of code that works to code you understand
17. **What's the most reliable way to generate a truly random number in C?**
    * A) Count the number of times you'll recompile before finding a missing brace
    * B) The number of minutes the lab computers take to boot
    * C) Use the rand() function but forget to seed it with srand()
    * D) Measure time between "It works!" and discovering why it works
18. **If your C program has no comments but runs perfectly, is it good code?**
    * A) Yes, but only until you need to modify it next week
    * B) No, but you'll get full marks anyway
    * C) It exists in a superposition of brilliant and terrible until your professor observes it
    * D) Yes, if you can convince yourself you'll remember how it works
19. **What happens when you finally understand someone else's uncommented C code?**
    * A) You immediately get assigned to maintain it forever
    * B) You gain the power to write equally incomprehensible code
    * C) You realize you've been staring at the screen for 6 hours and need food
    * D) Your newfound knowledge will be lost during the next compile error