CS 240 (Park) **Midterm** **July 7 (Thu.), 2022**

Remarks: Keep the answers compact, yet precise and to-the-point. Long-winded answers that do not address the key points are of limited value. Binary answers that give little indication of understanding are no good either. Time is not meant to be plentiful. Make sure not to get bogged down on a single problem.

PROBLEM 1 (36 pts)

(a) Consider the code snippet

int a, *b, *c; a = 3; b = &a; printf("%d", *b); *c = 5; printf("%d", *c);

Explain in detail what is likely to happen if the code snippet is compiled and executed.

(b) What are the possible outcomes if the code snippet

char r[4]; r[0] = 'H'; r[1] = 'i'; printf("%s", r);

is compiled and executed? Explain your reasoning.

(c) Suppose we have a 2-D array, int x[2][3], wherein 6 integers are stored. What array expression is *(*(x+1)+2) equivalent to, and why?

PROBLEM 2 (32 pts)

(a) Suppose main() calls function

int abc(void) { int a = 3, static int b = 1; if(++a > ++b) return a++; else return ++b;}

three times. Explain what values are returned to main() in each of the three calls to abc().

(b) Suppose the code snippet

float m, **n; m = 3.3; printf("%f", m); **n = 5.5; printf("%p", n);

is compiled and executed. What is likely to happen, and why? How would you modify the code (involving printf() calls) to facilitate ease of run-time debugging?

PROBLEM 3 (32 pts)

(a) Suppose you are supervising a team of C programmers. One of the team members is responsible for coding a function, int readpasswd(void), that reads from stdin a new password and checks that it contains upper case letters, special characters, etc. per company policy. The team member shows you part of the code

int readpasswd() { char secret[100]; scanf("%s", secret); /* code follows to check validity of password */}

that reads a password from stdin and stores it in local variable secret for further processing. Explain why you would be alarmed by the code. How would you rewrite to fix the problem in the code?

(b) Code main() that reads a file, test.out, byte by byte using fgetc() and counts how many bytes are ASCII characters. main() outputs the count to stdout. Focus on making sure that your code is robust and does not crash unexpectedly.

BONUS PROBLEM (10 pts)

Suppose you are given the code in main.c

int s[5]; int main() {int i; for(i=0; i<50; i++) s[i] = 0;}

which is compiled using gcc and executed. What are the two possible outcomes? Explain your answer.