CS 240 (Park) **Midterm** **July 6 (Thu.), 2023**

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 (30 pts)

(a) Consider the code snippet

int x, *y, *z; x = 5; y = &x; *y = 10; printf("%d %p\n", x, y); *z = 3;

Explain what is likely to happen if the code snippet is compiled and executed as part of main().

(b) Explain what the declarations of g and h mean:

char *g(char *), (*h)(char *);

For the two assignment statements to be meaningful

x = g(s); h = y;

what must be the types of x and y? Provide the C statements for their type declarations.

PROBLEM 2 (30 pts)

(a) For the function

void fun(float a) { float x[5], i; for(i=0; i<8; i++) x[i] = a; }

explain what is likely to happen if fun() is called by main(). Explain how things change if 1-D array x is made to be global.

(b) What are potential issues associated with code snippet

FILE *f; char r[100]; f = fopen("data.dat", "r"); fscanf(f, "%s", r);

Provide modified code that fixes the issues.

PROBLEM 3 (40 pts)

(a) A 2-D integer array, int d[100][200], declaration is restrictive in that it hardcodes the number of rows and columns to fixed values 100 and 200, respectively. Suppose two integers N and M are read from stdin that specify the number of rows and columns of a 2-D integer array which is then used to read N x M integers from stdin into main memory. Provide C code main() that uses malloc() to achieve this task. Your code should be complete but for including header files.

(b) Provide code that reads a value of type unsigned int from stdin, then uses bit processing techniques to count how many of the 32 bits contain bit value 0. Annotate your code to note what the different parts are doing.

BONUS PROBLEM (10 pts)

Explain why printf("%d", x) passes argument x by value whereas scanf("%d", &x) passes the argument by reference. Can one code printf() so that it passes x by reference? If so, why is it not done?