CS 240 (Park) **Midterm** **July 8 (Mon.), 2024**

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 (45 pts)

(a) Consider the code snippet

int x, *y; x = 5; printf("%p\n", y); *y = 6; y = &x;

Explain what is likely to happen if the code snippet is compiled and executed. What will be likely consequence of removing '\n' in the format string of printf()?

(b) Suppose we want to use variable, int x, as a mask where the first 16 bits (starting with the least significant bit) are set to 1 and the remaining bits to 0. Using bit processing techniques only as discussed in class, provide an assignment statement for x that makes it so.

(c) Given a 1-D array, char h[5], suppose we want to call scanf() to read a single ASCII character into h[3]. Provide two different ways of calling scanf() so that this is accomplished.

PROBLEM 2 (30 pts)

(a) Given the code, int main() { float z[5]; int i; for(i=0; i<7; i++) z[i] = 0.5; }, what is likely to happen if the code is compiled and executed? What if z is made global? What about the case where z, float *z, is local and

z = malloc(5 * sizeof(float));

is used to dynamically allocate memory? Explain your reasoning in the three cases.

(b) Suppose, FILE *fp, points to a binary file containing both non-ASCII and ASCII characters that has been opened for read. Using library function, int fgetc(FILE *fp), write code snippet that reads the content of the file byte-by-byte, counts how many bytes are contained in the file (i.e., file size), and how many bytes are ASCII characters. Store the value returned by fgetc() into variable, int y. No need to worry about header files or what function your code snippet is part of.

PROBLEM 3 (25 pts)

Suppose an ASCII text file contains an essay where each line of the file ends with newline character '\n' inclusive the last line. Without writing C code, describe in words how you would determine using fgetc() how many lines the file contains (N) and the maximum length of a line (M) in bytes inclusive newline. Assuming N and M have been determined, suppose the goal is to read the content of the file into a 2-D data structure pointed to by, char **essayptr, where memory is dynamically allocated using malloc(). Describe the layout of the 2-D structure (you may draw if you wish) in main memory. Write the C statement that assigns an address to essayptr by calling malloc(). Explain what the statement does.

BONUS PROBLEM (10 pts)

In Problem 2(b), we use library function fgetc() that reads the content of a file one byte at a time but its return type is defined as int, not char. Based on the discussion in class, explain why this is so.