

*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) Suppose `main()` calls the following function twice:

```
void myfunc() {int a = 0; a++; printf("%d", a); }
```

What gets printed by `myfunc()` and why? What gets printed if `int a` is declared outside of `myfunc()`? Explain.

(b) The following code snippet is likely to result in segmentation fault:

```
char *s; s[0] = 'h'; s[1] = 'i'; s[2] = '\0';
```

What is the reason for that? Fix the code, by making use of `malloc()`, so that segmentation fault does not arise.

(c) Suppose `x` is of type `unsigned int` and you need to inspect and print the value of its 7'th bit (counting from the least significant as the 0'th bit). Write a snippet of code that accomplishes this.

**PROBLEM 2** (40 pts)

(a) Suppose a function `ourfunc()` reads from standard input and stores into global `r[]` as follows:

```
void ourfunc() { extern char r[1000]; char c; int i = 0; while((c = getchar()) != '\n') r[i++] = c; }
```

What are potential problems that may lead to run-time errors? How would you fix the problems?

(b) The simple shell program discussed in class followed a concurrent server design. Why can a shell not be coded as an iterative server (i.e., a server that does not fork child processes but performs a requested task itself)? When calling `execlp()` in the child process, why was it necessary to check the return value of `execlp()`? After cloning a process using `fork()`, how does a process know whether it is the child or parent? What additional check should be performed after calling `fork()` (beyond checking whether a process is the child or parent) to prevent a potential run-time bug? Why was a call to `waitpid()` needed in the parent process? What happens if `waitpid()` is not called by the parent?

**PROBLEM 3** (24 pts)

Suppose you are given a file containing an arbitrary number of strings in the following format:

```
3
5 hello
2 hi
4 boat
```

The first integer specifies the number of strings and the integer preceding each string specifies the number of characters in the string. Assuming the file has been opened and is pointed to a file pointer `fp`, write a snippet of code that uses `malloc()` to read in and store the strings in a variable `s` declared as `char **s`. Use `fscanf()` to read the input.

**BONUS PROBLEM** (10 pts)

In one of the code examples discussed in class which had the form

```
int *x; printf("hi"); *x = 5;
```

the running program terminated with a segmentation fault without printing "hi" on the terminal display, even though we verified that `printf()` was called successfully. What was the reason for that? How did we fix the problem (the printing part)?