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 a file abc.c contains function abc()'s source code. Why is it sometimes useful to generate the object code abc.o? For a function xyz() in xyz.o to make use of (i.e., call) abc(), what must be done? Suppose you make changes to the source of abc(). How does the utility make know to recompile abc.c?

(b) For the code snippet

```c
int s = 1, *t, *u; t = &s; *t = 3; printf("%d", s); *u = 5; printf("%d", *u);
```

explain what happens when we compile and execute the code.

(c) Explain how the following three function declarations are different:

```c
float mm(double), *nn(double), (*pp)(double);
```

Are the four statements `nn = &mm; pp = nn; pp = &nn; pp = &printf;` meaningful? Explain.

**PROBLEM 2** (32 pts)

(a) Explain what happens when the code snippet

```c
char s[3] = "hi", *h; h = s; printf("%s", h+1); printf("%p", s); printf("%c",*h);
```

is compiled and executed. Why do we not declare and initialize, `char s[2] = "hi",` given that the string "hi" has only two characters?

(b) Given the code snippet

```c
float ddd[3][3]; ddd[0][1] = 0.1; ddd[1][2] = 1.2; ddd[2][1] = 2.1; printf("%f", *(*ddd+1) + 2));
```

explain what gets printed.

**PROBLEM 3** (32 pts)

(a) Suppose main() calls the function `int bbb(void) { int a = 0; return a++; }` three times. What values are returned, and why? What happens if we change `int a = 0` to `static int a = 0` as global? Explain.

(b) Write a piece of code as part of int main(void) that will compile correctly but is likely to result in segmentation fault. Explain why that is the case. Find a fix for the segmentation fault.

**BONUS PROBLEM** (10 pts)

Suppose you are reading an unsigned int from stdin and you need to determine the value of the 20th bit (i.e., bit position 19 starting from bit position 0 as least significant bit). Explain, in words, the steps involved in carrying out this task.