1. Give the three address code (including labels!) for the following piece of code.

```c
for (i = 0; i < N; i = i + 1) {
    for (j = i; j < N; j = j + 1) {
        k = k + 1;
    }
}
```

Assume that you have a three-address instruction `BLT A B L`, which branches to label `L` if `A < B`, and another instruction `JMP L`, which is an unconditional jump to label `L`. Otherwise, use the same three address instructions defined in problem set 4.

2. Consider the following piece of code:

```c
void main() {
    int i, j;
    i = foo(i, j);
}
```

```c
int foo(int a, int b) {
    int c;
    ...
}
```

Assuming that, before executing `main`, the stack has nothing on it, show the stack immediately after calling `foo` (i.e., before `foo` returns). Assume there are 4 registers that need to be saved (not including the frame pointer and the stack pointer), and that we are using a callee-saves convention. On your stack, show each item on the stack, and give the size of each. Show what the frame pointer points to.

3. Cam Piler, of problem set 4 fame, has another interesting idea. He thinks that if a program has no global variables, and all of its functions only take one argument, he can implement “pass by value-result” (i.e., copy-in, copy-out) by treating all such arguments as pass-by-reference. Is he right? Why or why not? If he is right, why would this be a good optimization?