Homework 6

Keizou Wang

April 22, 2025

1. Consider the following C program:

```
void foo()
{
    int i;
    printf("%d ", i++);
}

void main()
{
    int j;
    for (j = 1; j <= 10; j++)
        foo();
}
```

Local variable i in subroutine foo is never initialized. On many systems, however, variable i appears to "remember" its value between the calls to foo, and the program will print 0 1 2 3 4 5 6 7 8 9.

(a) Suggest an explanation for this behavior

i is declared without a default value so the value is determined by what's already at the address. Therefore, whenever the foo is ran, i takes the same address and the same value is incremented.

(b) Change the code above (without modifying function foo) to alter this behavior.

```
void main()
{
    int j, i;
    for (j = 1; j <= 10; j++){
        printf("%d ", i);
        foo();
    }
}
```

2. Can you write a macro in C that "returns" the factorial of an integer argument (without calling a subroutine)? Why or why not?

It would be impossible because recursion is impossible with just textual substitution and a loop solution can't directly "return" a value.

3. Consider a subroutine `swap` that takes two parameters and simply swaps their values. For example, after calling `swap(X,Y)`, `X` should have the original value of `Y` and `Y` the original value of `X`. Assume that variables to be swapped can be simple or subscripted (elements of an array), and they have the same type (integer). Show that it is *impossible* to write such a general-purpose `swap` subroutine in a language with:

(a) Parameters passing by value.

(b) Parameters passing by name.