

## Homework 6

(Due November 19)

1. (20 pts) Can you write a macro in C that “returns” the factorial of an integer argument (without calling a subroutine)? Why or why not?
2. (24 pts) Consider the following (erroneous) program in C:

```
void p ()
{
    int y;
    printf ("%d ", y);
    y = 2;
}
void main ()
{
    p();
    p();
}
```

Although the local variable `y` is not initialized before being used, the program prints two values – the first value typically is garbage (or possibly 0, if you are executing inside a debugger or other controlled environment), but the second value might be 2 (try this on Unix!).

- (a) (12 pts) Explain this behavior. Why does the local variable `y` appear to retain its value from one call to the next?
  - (b) (12 pts) Explain in what circumstances (without modifying function `p`) the local variable `y` will *not* retain its value between calls, and show an example.
3. (24 pts) 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) (12 pts) parameter passing by value.
    - (b) (12 pts) parameter passing by name.

Hint: for the case of passing by name, consider mutually dependent parameters.

4. (32 pts) Consider the following program, written in no particular language. Show what the program prints in the case of parameter passing by (a) value, (b) reference, (c) value-result, and (d) name. Justify your answer. When analyzing the case of passing by value-result, you may have noticed that there are two potentially ambiguous issues – what are they?

```
int i;  
int a[2];  
  
p (int x, int y)  
{  
    x++;  
    i++;  
    y++;  
}  
  
main ()  
{  
    a[0] = 1;  
    a[1] = 1;  
    i = 0;  
  
    p(a[i], a[i]);  
  
    print (a[0]);  
    print (a[1]);  
}
```

5. (Extra Credit - 10 pts) Does a program run faster when the programmer does not specify values for the optional parameters in a subroutine call?