

CS 430 Parameters

1. Suppose a subprogram `void f(int a) { a += 1 }` is called with the statement `f(x)`, where `x` is a local variable. Draw pictures to show how the value of `x` is passed between the subprogram and its caller using (a) pass-by-value, (b) pass-by-result, (c) pass-by-value-result, and (d) pass by reference.
2. Which parameter passing mechanisms implement *in* mode parameter semantics?
3. Which parameter passing mechanisms implement *out* mode parameter semantics?
4. Which parameter passing mechanisms implement *inout* mode parameter semantics?

Consider the following program in a C-like language with static scope.

```
program p
{
  int[] m = [2, 3, 5, 7, 11, 13, 17]
  int x, y = 0, 1;

  void g(int[] a, int b, int c)
  {
    a[b] = 0;
    b += 1;
    y += b;
    a[c] = 8;
  } // end of g

  println(m[0], m[1], m[2], x, y);
  g(m, x, y);
  println(m[0], m[1], m[2], x, y);
  g(m, y, y);
  println(m[0], m[1], m[2], x, y);
}
```

What is the output of this program under the following conditions?

5. All types are value types and all parameters are passed by value.
6. All types are value types and all parameters are passed by value-result. Results are assigned from right-to-left.
7. All types are value types and all parameters are passed by reference.
8. All types are reference types and all parameters are passed by value.
9. The `int` type is a value type, array types are reference types, and all parameters are passed by value.
10. The scalar types are value types, array types are reference types, and all parameters are passed by value-result. Results are assigned from right-to-left.