1. Arguments to C functions are “passed by value”, meaning that the called function is given copies of its arguments. Thus, in the following code, `func1` is given a copy of its argument `a`.

```c
int func1(int a) {
    a = a+2;
    printf("a + 2 = %d\n", a);
    return 1;
}

int main() {
    int x = 3;
    func1(x);
    if (x != 3)
        printf("shouldn’t happen\n");
    return 0;
}
```

Consider the following code, in which `func2` changes the contents of the array passed to it.

```c
int func2(int a[], int size) {
    for (int i=0; i<size; i++)
        a[i] += 2;
    return 1;
}

int main() {
    int x[4]={0, 1, 2, 3};
    func2(x);
    for (int i=0; i<4; i++)
        if (x[i] == i)
            printf("shouldn’t happen\n");
    return 0;
}
```

Though the argument `a` was passed by value, only its pointer value was copied; the actual contents of the array it refers to were not. Thus `func2` was able to modify the contents of the array, and have these modifications seen by the caller. Such copying of arguments (as in done in C), in which just the top-level pointer is copied, is known as shallow copying. In contrast, if deep copying were employed (this doesn’t happen in C!), `func2` would have received a copy of the entire array. But if `func2` then modified this copy, the changes would not be seen by the caller.
A further variation on parameter passing (and, again, not employed by C), is known as value-result. In this variation, the arguments are copied to the called function, as with pass by value. But when the called function terminates, the (possibly modified) arguments are copied back to the caller. If this were used in conjunction with deep copying in our example, the main function would see the changes made to the array it passed to func2.

Consider func3 below.

```c
int func3(int a1[], int size1, int a2[], int size2) {
    a1[0] = 1000;
    a2[size2-1] = 1000;
    return 1;
}
```

One might ask whether calls to func3 have the same effect if func3’s arguments are passed by value as they do if its arguments are passed by deep-copy value-result. It turns out that there are cases in which the effects are different. Provide an example showing this; i.e., provide an example of passing arguments to func3 such that their values, after the call, are different depending on the argument passing technique. You may assume that arguments are copied in the order in which they appear.