QUIZ 2 – SOLUTION

NAME:

Problem 1: Loops

Convert the following for-loop to an equivalent while-loop.

```
for (int i = 1; i <= SIZE; i++) {
     cin >> nums[i];
     cout << nums[i] << endl;
}
```

```
int i = 1;
while (i <= SIZE) {
    cin >> nums[i];
    cout << nums[i] << endl;
    i++;
}
```

Note that the value of i must be updated at the *end* of the loop, not at the beginning.

Problem 2: Arrays

The following function is supposed to reverse the elements of an array of size SIZE. For example, if the array is {1,2,3,4}, then the function should make it {4,3,2,1}. Yet, it doesn't. What is wrong with it and how can it be fixed? (Hint: try "running" the code by hand for a couple of small arrays and see what happens.)

```
void reverseArray(int array[]) {
    int temp;
    for (int i = 0; i <= SIZE; i++) {
        temp = array[i];
        array[i] = array[SIZE-i-1];
        array[SIZE-i-1] = temp;
    }
}
```

There are two problems here:

First, i should not be allowed to take the value SIZE, since the array will be overrun in that case (recall that indices go from 0 to SIZE-1).

Second, by allowing the loop to iterate SIZE times, we are reversing the array twice. Instead, the loop should only iterate SIZE/2 times.

The corrected loop is

```
void reverseArray(int array[]) {
    int temp;
    for (int i = 0; i < SIZE/2; i++) {
        temp = array[i];
        array[i] = array[SIZE-i-1];
        array[SIZE-i-1] = temp;
    }
}
```

Note that `i <= SIZE/2` would work for arrays of odd size (it would just "swap" the middle element twice), but not for arrays of even size.

Problem 3: Pointers

1. Declare an integer variable x, and a pointer to integer ptr, using only one statement.

```
int *ptr, x;
```

2. Show two different ways to initialize ptr. You do not have access to variables other than x and ptr.

   Any two of the following are acceptable:

```
ptr = &x;
ptr = NULL;
ptr = new int;
```