Using Pointers to Send Information Back From a Function

Example: Here is a complete example in which:

- The caller is ***testMinAndMax***. It sends two numbers, ***x*** and ***y***, to the function ***minAndMax*** (using ordinary parameters, as usual).
- The caller (***testMinAndMax***) wants to get back two pieces of information from ***minAndMax***: the smaller of ***x*** and ***y***, and the larger of ***x*** and ***y***. It wants to store the former in the variable ***min*** and the latter in the variable ***max***.

```
void testMinAndMax() {
    int x = 123;
    int y = 11;
    int min;
    int max;

    minAndMax(x, y, &min, &max);


    printf("Min of x and y is: %i.   Max is: %i\n", min, max);
}

void minAndMax(int a, int b, int* pMin, int* pMax) {
    if (a < b) {
        *pMin = a;
        *pMax = b;
    } else {
        *pMin = b;
        *pMax = a;
    }
}
```

min and ***max*** are garbage at this point

Pass the ADDRESSES of ***min*** and ***max***.

The ***minAndMax*** function puts the "answers" at those ADDRESSES. Here that means that it will put the smaller of ***x*** and ***y*** at the address of ***min*** and the larger of ***x*** and ***y*** at the address of ***max***.

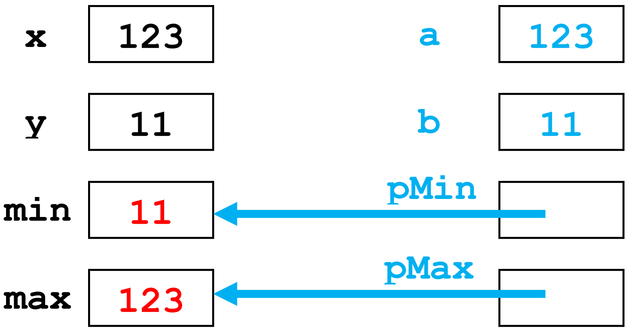
So after this function call, variables ***min*** and ***max*** have the intended values.

Tests whether ***min*** and ***max*** are set correctly by the ***minAndMax*** function.

`*pMin` means "the thing at `pMin`", that is, ***pMin***'s ***pointee***. Since the pointee is a variable in the caller, setting that pointee has the effect of sending information back to the caller.

The caller sent ***addresses*** of ***int***'s as the 3rd and 4th arguments, so ***minAndMax*** has ***pointers*** to ***int***'s as the corresponding parameters.

- The caller (**testMinAndMax)** passes ***x*** and ***y*** to **minAndMax**, plus the *addresses* of ***min*** and ***max***.
- The **minAndMax** function stores copies of ***x*** and ***x*** in ***a*** and ***b***, respectively, and copies of the addresses of ***min*** and ***max*** in ***pMin*** and ***pMax***, respectively.
- The **minAndMax** function figures out which is smaller, ***a*** or ***b***, and puts that smaller value at ***pMin***, the address of ***min***. That is, it puts that smaller value at ***pMin***'s ***pointee***. It puts the larger value at ***pMax***, the address of ***max***, i.e., at ***pMax***'s ***pointee***.



Here is the box-and-pointer diagram.

- ***Black*** shows the boxes (variables) allocated by the caller ***testMinAndMax***.
- **Blue** shows what happens when ***minAndMax*** is *called*.
- **Red** shows what the *body* of ***minAndMax*** does, in this example.

Test your understanding. Do you see:

- Why **testMinAndMax** passes the ***ADDRESSES*** of ***min*** and ***max*** instead of ***min*** and ***max*** themselves?
- The notations for:
  - ***Passing an address***? (i.e., using an ampersand &)
  - ***Receiving an address*** in a pointer variable? (i.e., using an asterisk *)
  - Putting a value at the address that a pointer specifies, that is, ***setting the pointee***? (i.e., again using an asterisk *, but in a different way)
- For functions that use pointers as parameters to send information back to the caller:
  - When to have such functions?
  - How to call such functions?
  - How to write such functions?

If any of the above is not clear to you, please ask questions as needed now!