# Arrays

void main() {

int counter; // create one int named "counter" in local memory (stack)

int stuff[20]; // create 20 ints in local memory (stack)

//int stuff[counter]; Must be created with constant at compile time. Other ways

// to do dynamic

stuff[0] = 1; // : 0 is the "index"

stuff[1] = 2;

stuff[20] = 20;

cout << stuff[0] << " " << stuff[1] << " " << stuff[20] << endl;

cout << stuff[-1] << endl;

// :Unitialized stuff ... whatever was in memory

// :The size is NOT kept in the data structure. The compiler does not check.

// :If you are lucky it will write to an invalid memory location and your program will die.

// :Likely you'll change another piece of data. Difficult to debug.

system("pause");

}

int stuff[6] = {1,2,3,1,4,1}; // Can be initialized like other vars

//int stuff[] = {1,2,3,4}; // Let the compiler count for you

// You can only use the {} initializer at creation time.

//

//int more[5];

//more = {1,2,3,4,5}; does not work

// Write a program to count the 1s in the array

int count = 0;

int x = 1; // Start with first element

while(x<5) { // Elements 0 through 5 NOT 6, right?

if(stuff[x]==1) {

count = count + 1;

}

x=x+1;

}

// Show as a for loop

cout << count << endl;

Take first two

Swap if needed

Move down one place

Go to end

Repeat

3

2

1

1

4

1

// What is being passed here? Not all 6 ints ... the function is

// changing the values in what is passed.

// Pointer to existing data. Arrays and pointers are tied.

void sortArray(int data[], int size) {

bool changed;

do {

changed = false;

for(int x=0;x<5;++x) {

int a = data[x];

int b = data[x+1];

if(b<a) {

data[x] = b;

data[x+1] = a;

changed = true;

}

}

} while(changed);

}

void printArray(int data[], int size) {

for(int x=0;x<size;++x) {

cout << data[x];

if(x!=(size-1)) {

cout << ", ";

}

}

cout << endl;

}

void main() {

int data[] = {3,2,1,1,3,1};

sortArray(data,6);

printArray(data,6);

/\*

//for(int w=0;w<1000;++w) {

for(int x=0;x<5;++x) {

int a = data[x];

int b = data[x+1];

if(b<a) {

data[x] = b;

data[x+1] = a;

}

}

//}

cout << data[0] << "," << data[1] << "," << data[2] << ","

<< data[3] << "," << data[4] << "," << data[5] << ","

<< endl;

\*/

system("pause");

}