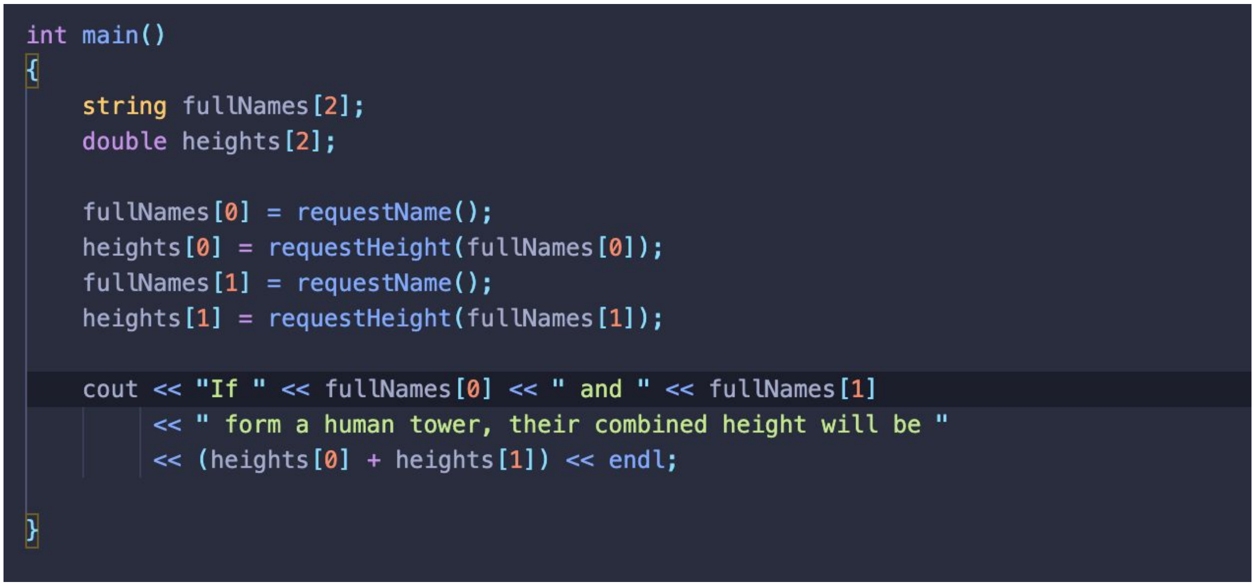
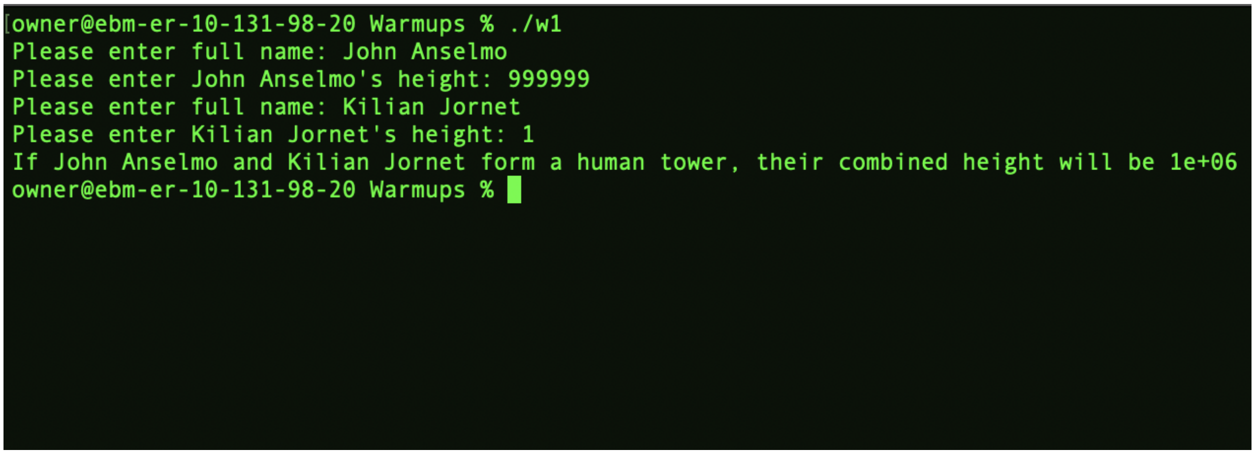
Lab 4 - ansel054

Warmup 1



```cpp
int main()
{
    string fullNames[2];
    double heights[2];

    fullNames[0] = requestName();
    heights[0] = requestHeight(fullNames[0]);
    fullNames[1] = requestName();
    heights[1] = requestHeight(fullNames[1]);

    cout << "If " << fullNames[0] << " and " << fullNames[1]
         << " form a human tower, their combined height will be "
         << (heights[0] + heights[1]) << endl;

}
```

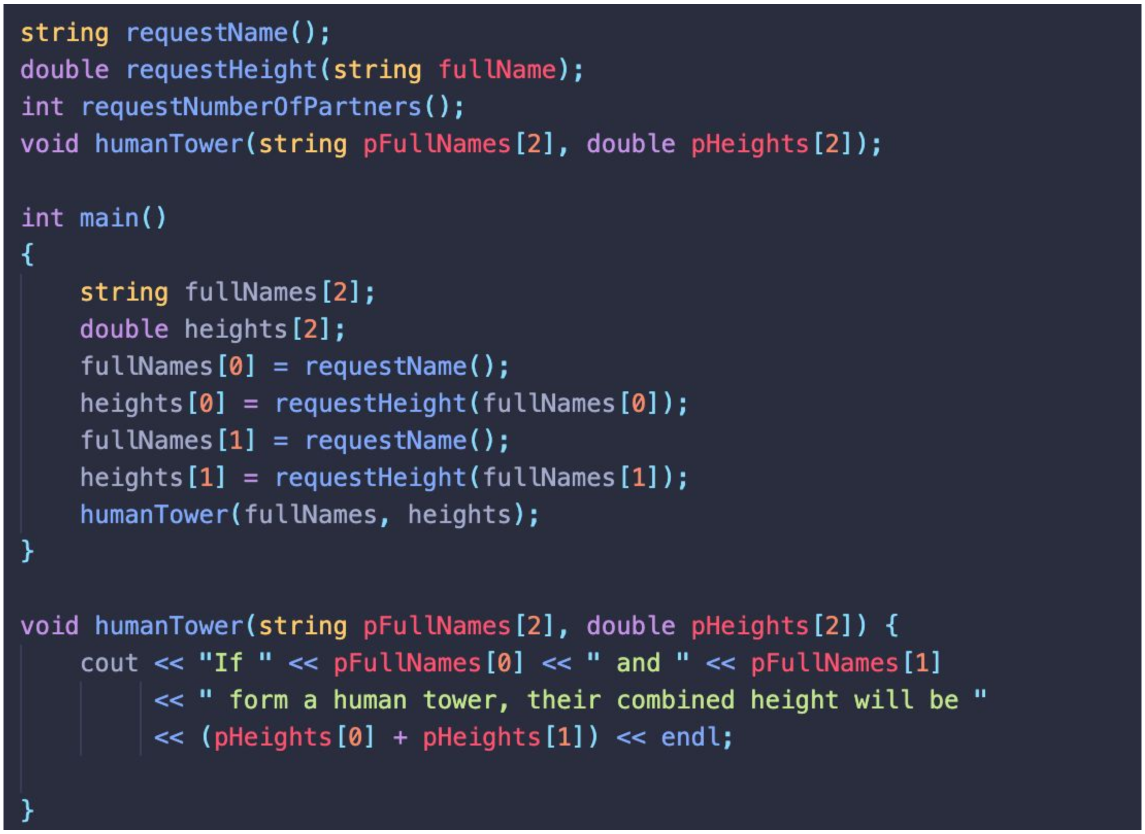


```
[owner@ebm-er-10-131-98-20 Warmups % ./w1
 Please enter full name: John Anselmo
 Please enter John Anselmo's height: 999999
 Please enter full name: Kilian Jornet
 Please enter Kilian Jornet's height: 1
 If John Anselmo and Kilian Jornet form a human tower, their combined height will be 1e+06
 owner@ebm-er-10-131-98-20 Warmups %
```

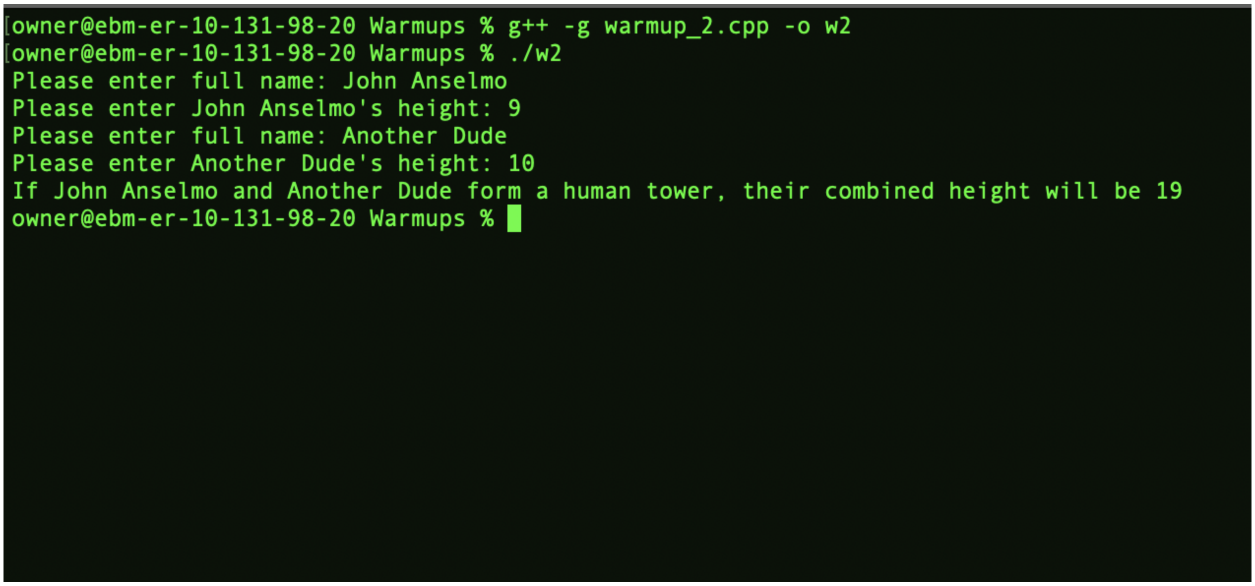
This just took the creation of two arrays tracking the two full names and the two heights. (Two arrays of size 2).

Lab 4 - ansel054

Warmup 2



```
string requestName();
double requestHeight(string fullName);
int requestNumberOfPartners();
void humanTower(string pFullNames[2], double pHeights[2]);

int main()
{
    string fullNames[2];
    double heights[2];
    fullNames[0] = requestName();
    heights[0] = requestHeight(fullNames[0]);
    fullNames[1] = requestName();
    heights[1] = requestHeight(fullNames[1]);
    humanTower(fullNames, heights);
}

void humanTower(string pFullNames[2], double pHeights[2]) {
    cout << "If " << pFullNames[0] << " and " << pFullNames[1]
        << " form a human tower, their combined height will be "
        << (pHeights[0] + pHeights[1]) << endl;

}
```

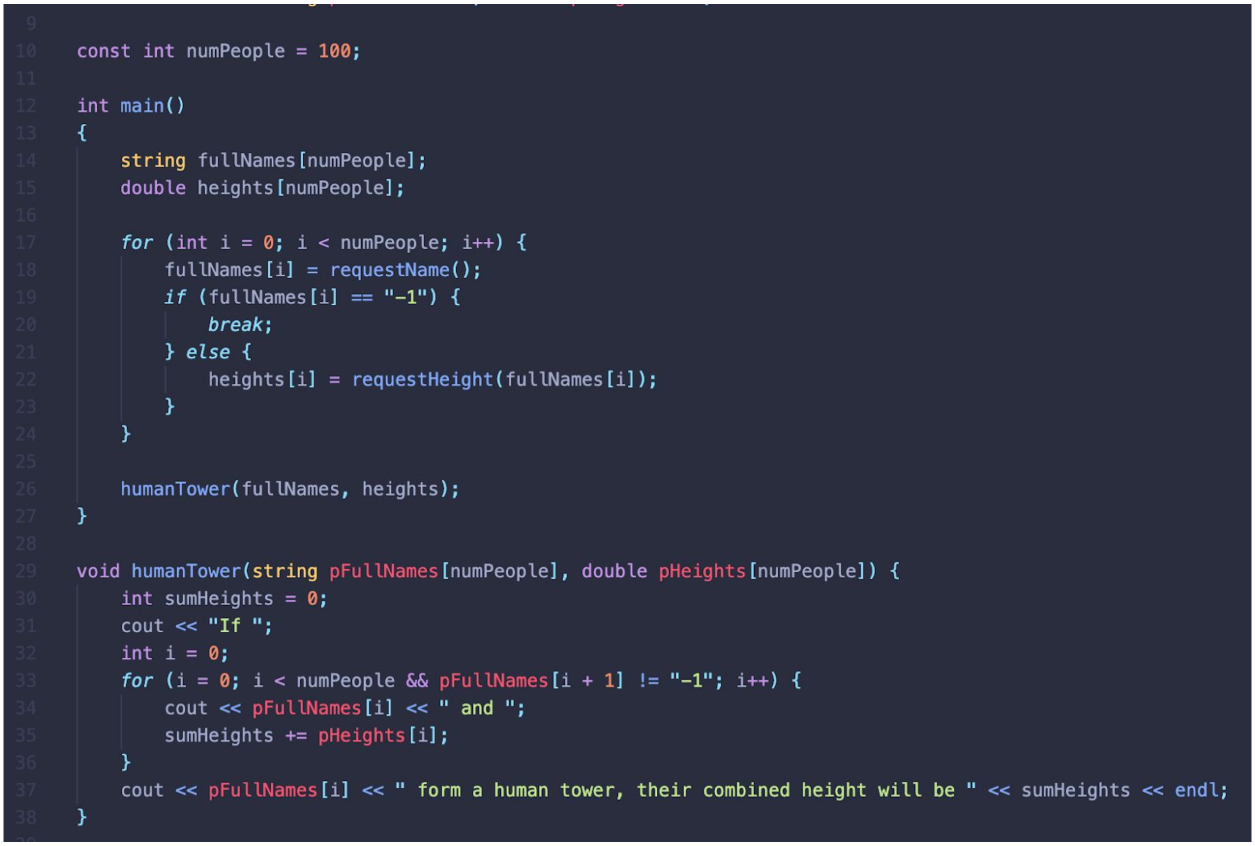


```
[owner@ebm-er-10-131-98-20 Warmups % g++ -g warmup_2.cpp -o w2
[owner@ebm-er-10-131-98-20 Warmups % ./w2
 Please enter full name: John Anselmo
 Please enter John Anselmo's height: 9
 Please enter full name: Another Dude
 Please enter Another Dude's height: 10
 If John Anselmo and Another Dude form a human tower, their combined height will be 19
 owner@ebm-er-10-131-98-20 Warmups %
```

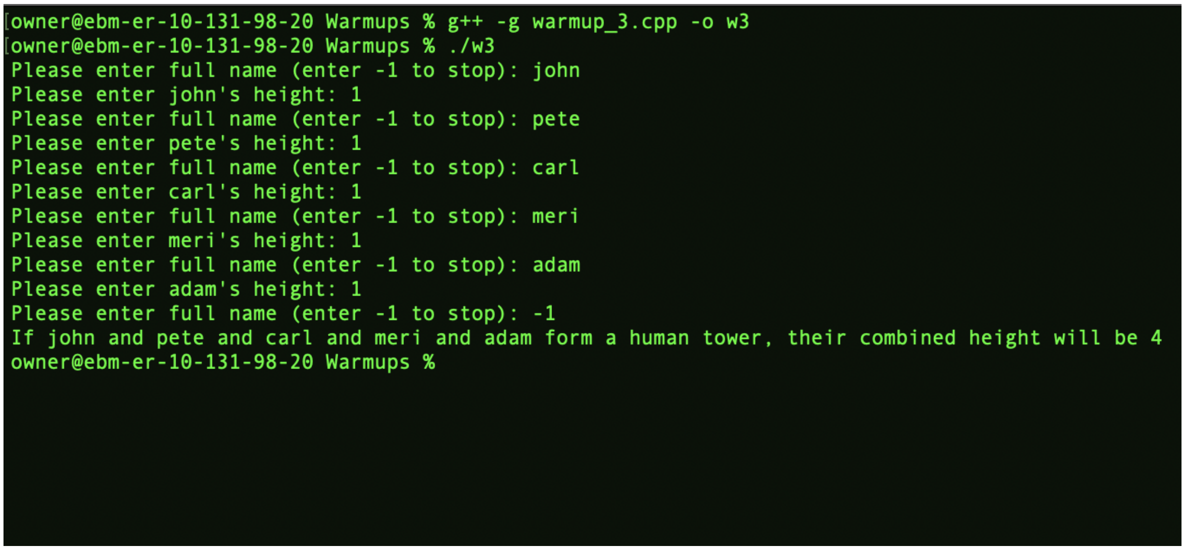
Created a function to compute the combined heights of two people from two 2-sized arrays of their names and heights.

Lab 4 - ansel054

Warmup 3



```cpp

const int numPeople = 100;

int main()
{
    string fullNames[numPeople];
    double heights[numPeople];

    for (int i = 0; i < numPeople; i++) {
        fullNames[i] = requestName();
        if (fullNames[i] == "-1") {
            break;
        } else {
            heights[i] = requestHeight(fullNames[i]);
        }
    }

    humanTower(fullNames, heights);
}

void humanTower(string pFullNames[numPeople], double pHeights[numPeople]) {
    int sumHeights = 0;
    cout << "If ";
    int i = 0;
    for (i = 0; i < numPeople && pFullNames[i + 1] != "-1"; i++) {
        cout << pFullNames[i] << " and ";
        sumHeights += pHeights[i];
    }
    cout << pFullNames[i] << " form a human tower, their combined height will be " << sumHeights << endl;
}
```

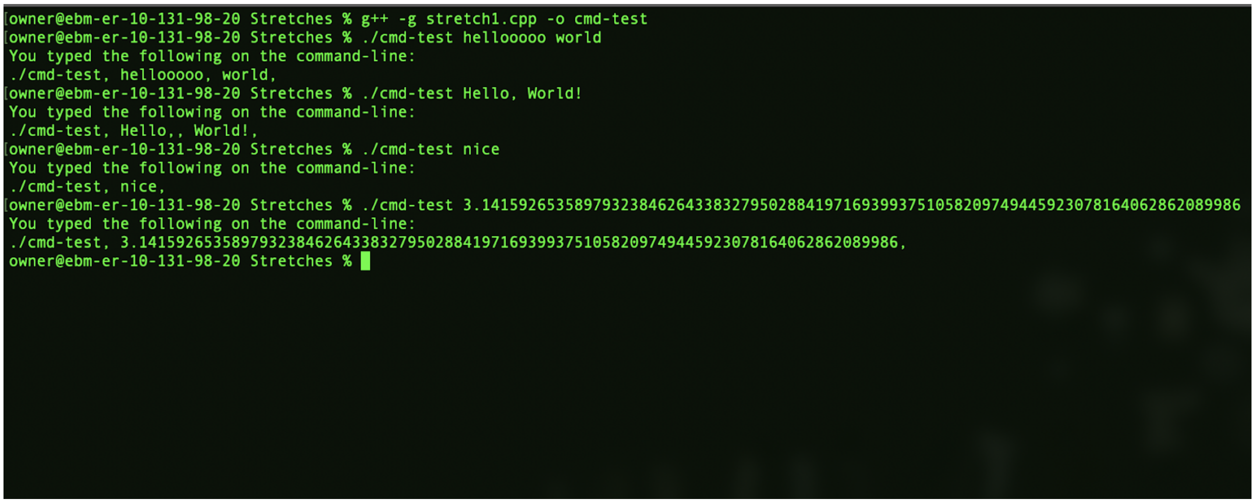


```
[owner@ebm-er-10-131-98-20 Warmups % g++ -g warmup_3.cpp -o w3
[owner@ebm-er-10-131-98-20 Warmups % ./w3
 Please enter full name (enter -1 to stop): john
 Please enter john's height: 1
 Please enter full name (enter -1 to stop): pete
 Please enter pete's height: 1
 Please enter full name (enter -1 to stop): carl
 Please enter carl's height: 1
 Please enter full name (enter -1 to stop): meri
 Please enter meri's height: 1
 Please enter full name (enter -1 to stop): adam
 Please enter adam's height: 1
 Please enter full name (enter -1 to stop): -1
 If john and pete and carl and meri and adam form a human tower, their combined height will be 4
 owner@ebm-er-10-131-98-20 Warmups %
```

Created an oversized array, filling it until the user inputs -1, then computing the combined heights of all people until reaching full name “-1”.

Lab 4 - ansel054

Stretch 1

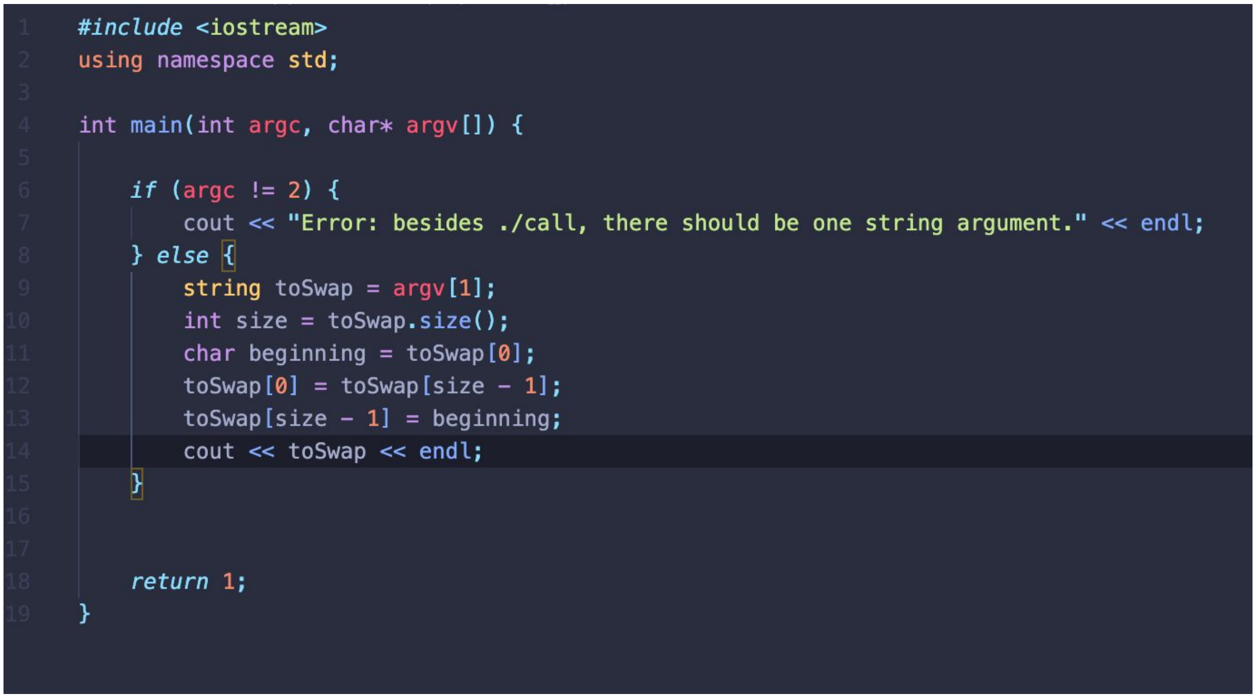


```
[owner@ebm-er-10-131-98-20 Stretches % g++ -g stretch1.cpp -o cmd-test
[owner@ebm-er-10-131-98-20 Stretches % ./cmd-test hellooooo world
You typed the following on the command-line:
./cmd-test, hellooooo, world,
[owner@ebm-er-10-131-98-20 Stretches % ./cmd-test Hello, World!
You typed the following on the command-line:
./cmd-test, Hello,, World!,
[owner@ebm-er-10-131-98-20 Stretches % ./cmd-test nice
You typed the following on the command-line:
./cmd-test, nice,
[owner@ebm-er-10-131-98-20 Stretches % ./cmd-test 3.14159265358979323846264338327950288419716939937510582097494459230781640628620899986
You typed the following on the command-line:
./cmd-test, 3.1415926535897932384626433832795028841971693993751058209749445923078164062862089986,
owner@ebm-er-10-131-98-20 Stretches % 
```

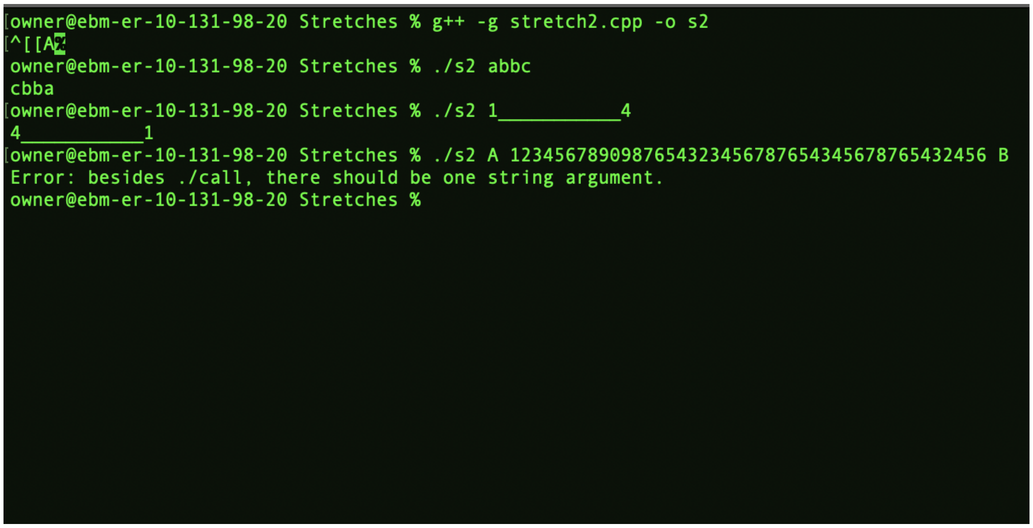
Just testing out command-line arguments!

Lab 4 - ansel054

Stretch 2



```cpp
#include <iostream>
using namespace std;

int main(int argc, char* argv[]) {

    if (argc != 2) {
        cout << "Error: besides ./call, there should be one string argument." << endl;
    } else {
        string toSwap = argv[1];
        int size = toSwap.size();
        char beginning = toSwap[0];
        toSwap[0] = toSwap[size - 1];
        toSwap[size - 1] = beginning;
        cout << toSwap << endl;
    }


    return 1;
}
```

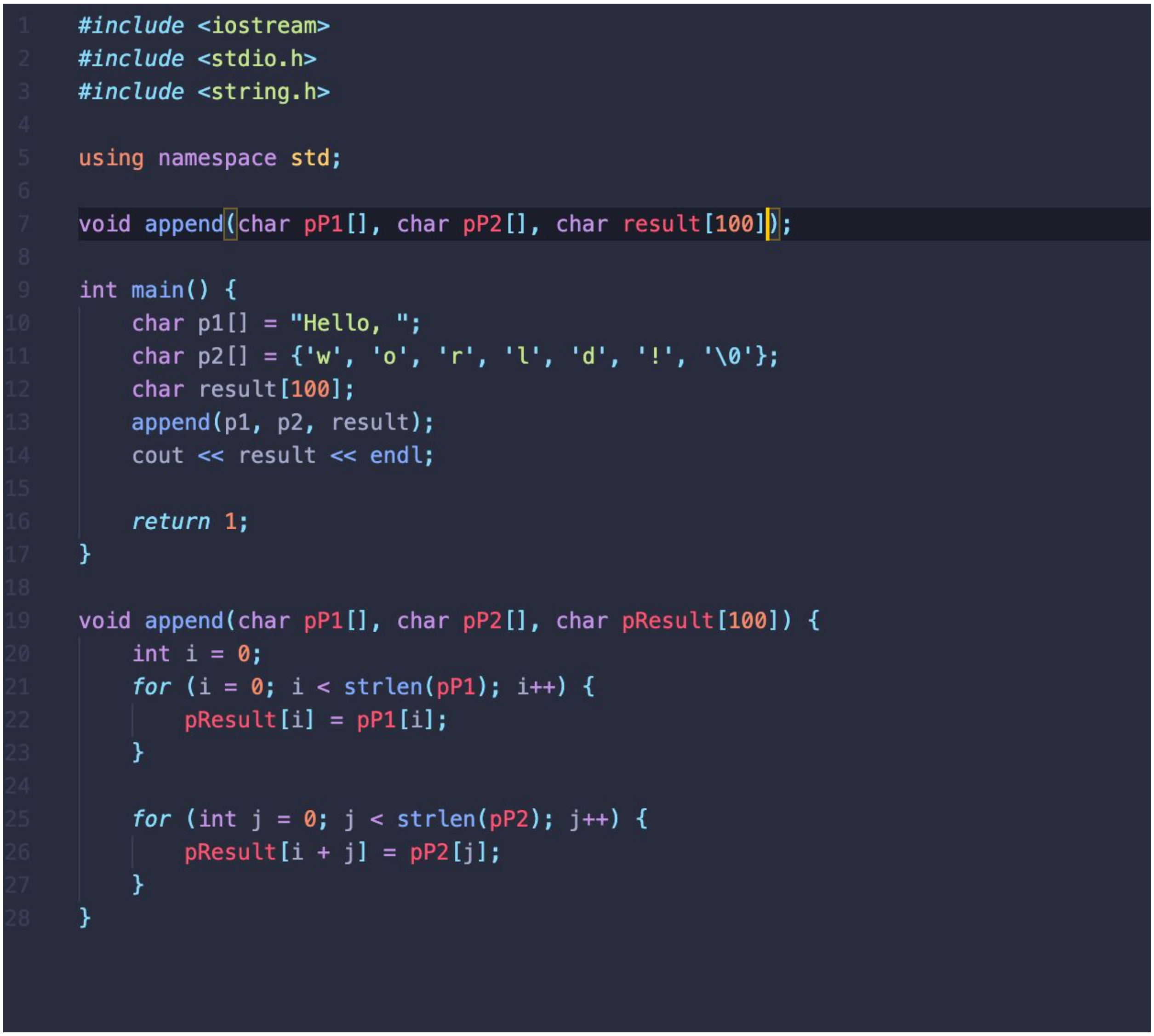


```
[owner@ebm-er-10-131-98-20 Stretches % g++ -g stretch2.cpp -o s2
[^[[A%
 owner@ebm-er-10-131-98-20 Stretches % ./s2 abbc
 cbba
[owner@ebm-er-10-131-98-20 Stretches % ./s2 1___________4
 4___________1
[owner@ebm-er-10-131-98-20 Stretches % ./s2 A 12345678909876543234567876543456787654324567 B
 Error: besides ./call, there should be one string argument.
 owner@ebm-er-10-131-98-20 Stretches %
```

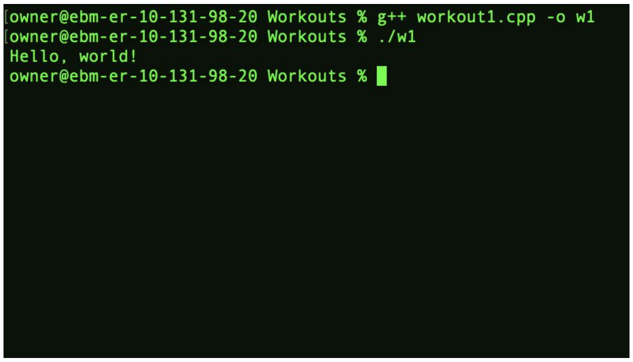
Used a temporary holder variable to keep track of the first character, replaced the first character with the last, then placed the holder variable (holding the first character) in the last slot.

Lab 4 - ansel054

Workout 1



```cpp
#include <iostream>
#include <stdio.h>
#include <string.h>

using namespace std;

void append(char pP1[], char pP2[], char result[100]);

int main() {
    char p1[] = "Hello, ";
    char p2[] = {'w', 'o', 'r', 'l', 'd', '!', '\0'};
    char result[100];
    append(p1, p2, result);
    cout << result << endl;

    return 1;
}

void append(char pP1[], char pP2[], char pResult[100]) {
    int i = 0;
    for (i = 0; i < strlen(pP1); i++) {
        pResult[i] = pP1[i];
    }

    for (int j = 0; j < strlen(pP2); j++) {
        pResult[i + j] = pP2[j];
    }
}
```



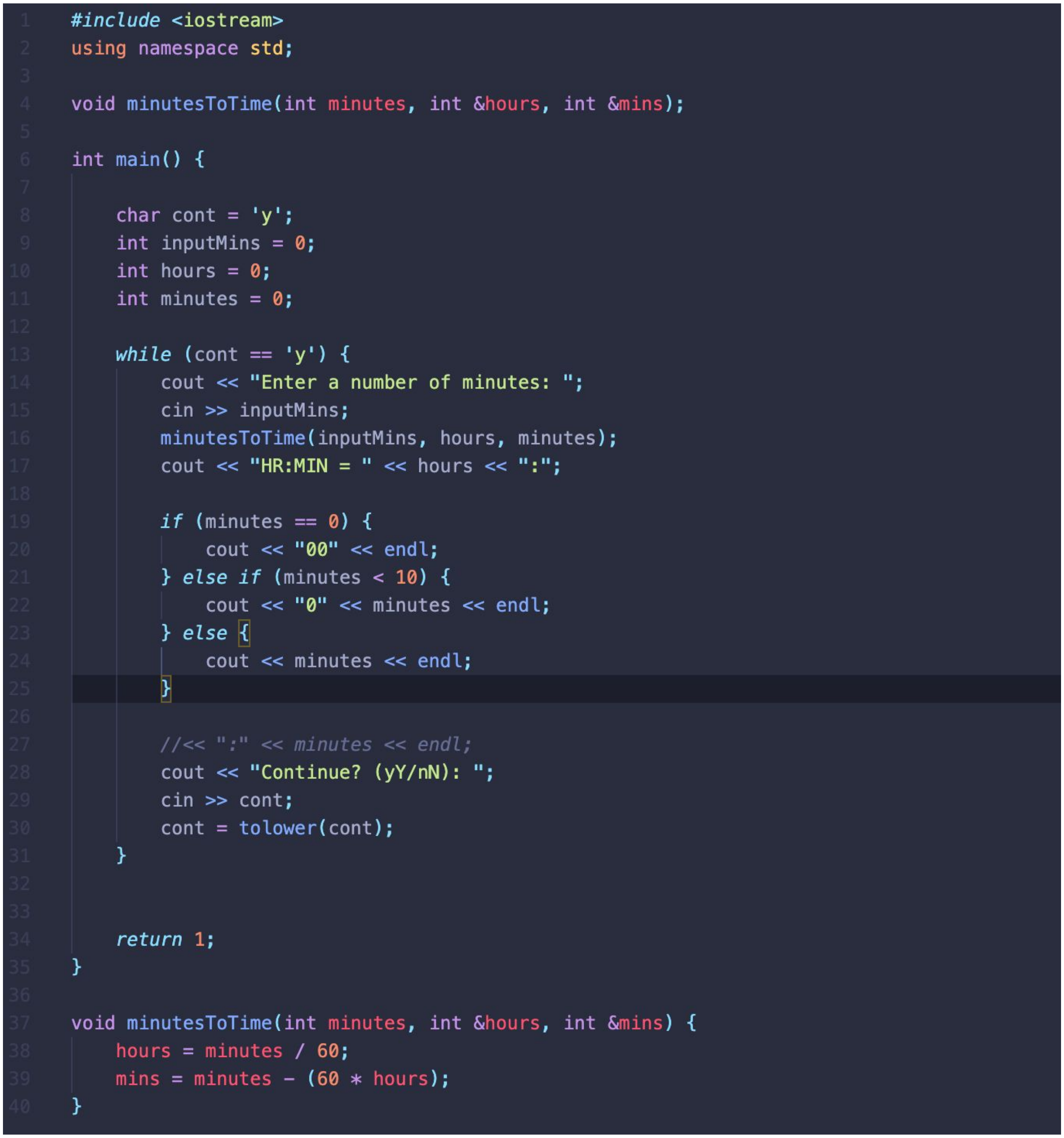
```
[owner@ebm-er-10-131-98-20 Workouts % g++ workout1.cpp -o w1
[owner@ebm-er-10-131-98-20 Workouts % ./w1
 Hello, world!
 owner@ebm-er-10-131-98-20 Workouts % 
```

Didn’t really need the sizes in the parameters! Added the second to the result but adding the previously left off index (from adding first) to the current, then accessing with just the current.

Lab 4 - ansel054

Workout 2

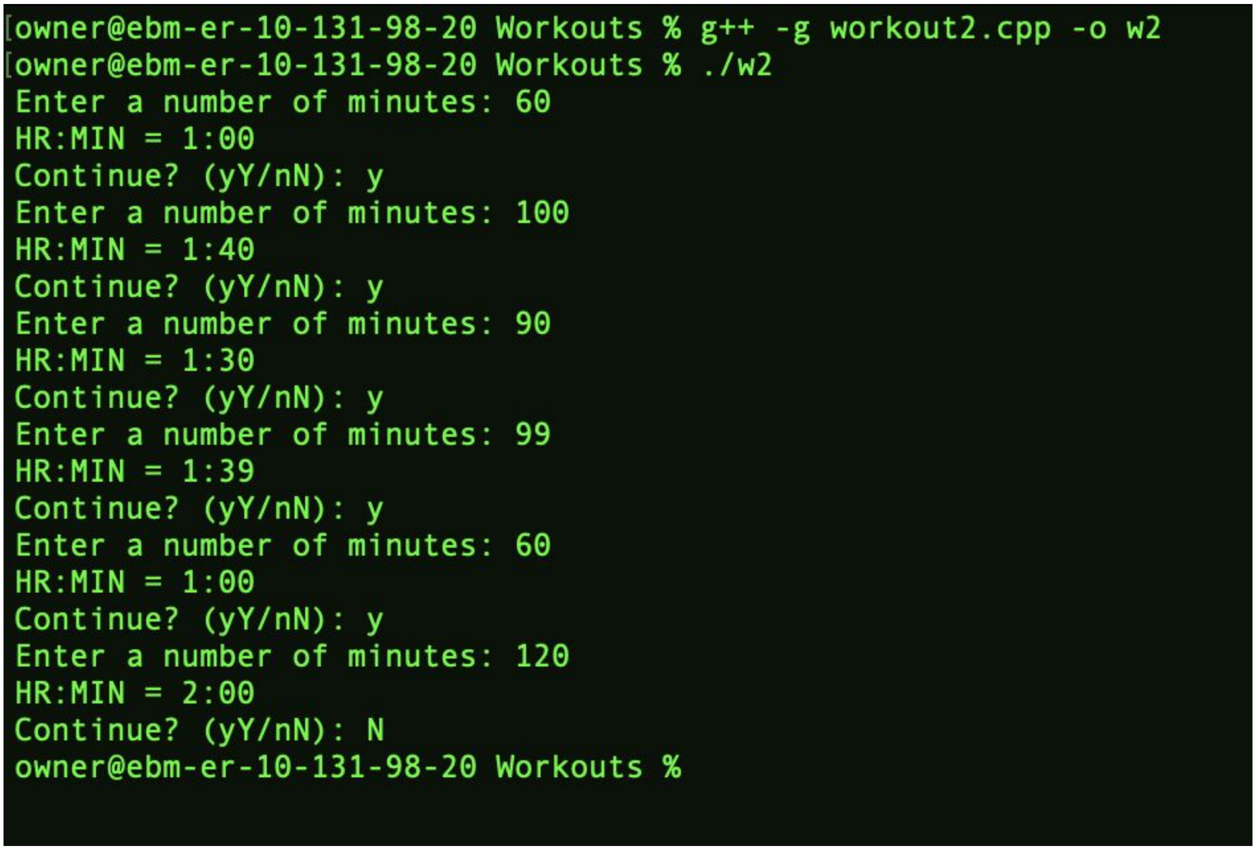
code



```
#include <iostream>
using namespace std;

void minutesToTime(int minutes, int &hours, int &mins);

int main() {

    char cont = 'y';
    int inputMins = 0;
    int hours = 0;
    int minutes = 0;

    while (cont == 'y') {
        cout << "Enter a number of minutes: ";
        cin >> inputMins;
        minutesToTime(inputMins, hours, minutes);
        cout << "HR:MIN = " << hours << ":";

        if (minutes == 0) {
            cout << "00" << endl;
        } else if (minutes < 10) {
            cout << "0" << minutes << endl;
        } else {
            cout << minutes << endl;
        }

        //<< ":" << minutes << endl;
        cout << "Continue? (yY/nN): ";
        cin >> cont;
        cont = tolower(cont);
    }


    return 1;
}

void minutesToTime(int minutes, int &hours, int &mins) {
    hours = minutes / 60;
    mins = minutes - (60 * hours);
}
```

Lab 4 - ansel054

Terminal output



```
[owner@ebm-er-10-131-98-20 Workouts % g++ -g workout2.cpp -o w2
[owner@ebm-er-10-131-98-20 Workouts % ./w2
 Enter a number of minutes: 60
 HR:MIN = 1:00
 Continue? (yY/nN): y
 Enter a number of minutes: 100
 HR:MIN = 1:40
 Continue? (yY/nN): y
 Enter a number of minutes: 90
 HR:MIN = 1:30
 Continue? (yY/nN): y
 Enter a number of minutes: 99
 HR:MIN = 1:39
 Continue? (yY/nN): y
 Enter a number of minutes: 60
 HR:MIN = 1:00
 Continue? (yY/nN): y
 Enter a number of minutes: 120
 HR:MIN = 2:00
 Continue? (yY/nN): N
 owner@ebm-er-10-131-98-20 Workouts %
```

This was done using simple division and subtraction. Hours would just be the amount of times 60 goes into the total minutes, and minutes left would just be the remaining minutes after the hours are taken out.