

5.5 Multiple vectors

Programmers commonly use multiple same-sized vectors to store related lists. The program below maintains a list of country names, and another list indicating average minutes of TV watched per day in each corresponding country.

The statement `if (ctryNames.at(i) == userCountry)` compares the current `ctryNames` element with the user-entered country name. If the names match, the program prints the `ctryMins` element at the same index.

The loop's expression `(i < ctryNames.size()) && (!foundCountry)` depends on the value of the variable `foundCountry`. This expression prevents the loop from iterating through the entire vector once the correct country is found.

The program's numbers aren't made up, by the way: Americans watch nearly 5 hours of TV per day on average.

Figure 5.5.1: Multiple vector example: TV watching time program.

```
Enter country name: USA
People in USA watch 274 mins of
TV daily.

...

Enter country name: Sweden
People in Sweden watch 154 mins
of TV daily.

...

Enter country name: Brazil
Country not found; try again.
```

```

#include <iostream>
#include <vector>
#include <string>
using namespace std;

int main() {
    // Source: www.statista.com, 2015
    const int NUM_COUNTRIES = 5;           // Num
countries supported
    vector<string> ctryNames(NUM_COUNTRIES); // Country
names
    vector<int> ctryMins(NUM_COUNTRIES); // Mins TV
watched daily
    string userCountry;                    // User
defined country
    bool foundCountry = false;             // Match to
country supported
    unsigned int i;                        // Loop
index

    // Fill vector contents
    ctryNames.at(0) = "China";
    ctryMins.at(0) = 155;

    ctryNames.at(1) = "Sweden";
    ctryMins.at(1) = 154;

    ctryNames.at(2) = "Russia";
    ctryMins.at(2) = 246;

    ctryNames.at(3) = "UK";
    ctryMins.at(3) = 216;

    ctryNames.at(4) = "USA";
    ctryMins.at(4) = 274;

    // Prompt user for country name
    cout << "Enter country name: ";
    cin >> userCountry;

    // Find country's index and average TV time
    foundCountry = false;
    for (i = 0; (i < ctryNames.size()) &&
(!foundCountry); ++i) {
        if (ctryNames.at(i) == userCountry) {
            foundCountry = true;
            cout << "People in " << userCountry << " watch
";
            cout << ctryMins.at(i) << " mins of TV daily."
<< endl;
        }
    }
    if (!foundCountry) {
        cout << "Country not found; try again." << endl;
    }

    return 0;
}

```

[Feedback?](#)

Consider the above TV watching program involving multiple vectors.

1) Multiple vectors saved memory over using one larger vector.

- ☐ True
☐ False

2) Each vector should be the same data type.

- ☐ True
☐ False

3) Each vector should have the same number of elements.

- ☐ True
☐ False

[Feedback?](#)

zyDE 5.5.1: Improve the TV watching time program.

Modify the program such that if a user types a country name that isn't found, print known countries.

Load default template...

```
1 #include <iostream>
2 #include <vector>
3 #include <string>
4 using namespace std;
5
6 int main() {
7     // Source: www.statista.com, 2015
8     const int NUM_COUNTRIES = 5;
9     vector<string> ctryNames(NUM_COUNTRIES);
10    vector<int> ctryMins(NUM_COUNTRIES);
11    string userCountry;
12    bool foundCountry = false;
13    unsigned int i;
14
15    // Fill vector contents
16    ctryNames.at(0) = "China";
17    ctryMins.at(0) = 155;
18
19    ctryNames.at(1) = "Sweden";
20    ctryMins.at(1) = 154;
21
```

USA

Run

[Feedback?](#)**CHALLENGE
ACTIVITY**

5.5.1: Printing the sum of two vector elements.



Add each element in `origList` with the corresponding value in `offsetAmount`. Print each sum followed by a space. Ex: If `origList = {40, 50, 60, 70}` and `offsetAmount = {5, 7, 3, 0}`, print:

45 57 63 70

```
5 int main() {
6     const int NUM_VALS = 4;
7     vector<int> origList(NUM_VALS);
8     vector<int> offsetAmount(NUM_VALS);
9     unsigned int i;
10
11     for (i = 0; i < origList.size(); ++i) {
12         cin >> origList.at(i);
13     }
14
15     for (i = 0; i < offsetAmount.size(); ++i) {
16         cin >> offsetAmount.at(i);
17     }
18
19     /* Your solution goes here */
20     for (i = 0; i < offsetAmount.size(); ++i) {
21         cout << origList.at(i) + offsetAmount.at(i) << " ";
22     }
23
24     cout << endl;
25
26     return 0;
}
```

Run

✓ All tests passed

✓ Testing with inputs: 40 50 60 70 5 7 3 0

Your output

45 57 63 70

✓ Testing with inputs: 3 30 300 3000 -5 -10 14 -1500

Your output

-2 20 314 1500

[Feedback?](#)

CHALLENGE
ACTIVITY

5.5.2: Multiple vectors: Key and value.



For any element in `keysList` with a value greater than 100, print the corresponding value in `itemsList`, followed by a space. Ex: If `keysList = {42, 105, 101, 100}` and `itemsList = {10, 20, 30, 40}`, print:

20 30

Since `keysList.at(1)` and `keysList.at(2)` have values greater than 100, the value of `itemsList.at(1)` and `itemsList.at(2)` are printed.

```
12     cin >> keysList.at(1);
13 }
14
15 for (i = 0; i < itemsList.size(); ++i) {
16     cin >> itemsList.at(i);
17 }
18
19 /* Your solution goes here */
20 for (i = 0; i < itemsList.size(); ++i) {
21     if(keysList.at(i) > 100)
22     {
23         cout<< itemsList.at(i) << " ";
24     }
25
26 }
27
28
29 cout << endl;
30
31 return 0;
32 }
```

Run

✓ All tests passed

✓ Testing with inputs: 42 105 101 100 10 20 30 40

Your output

20 30

✓ Testing with inputs: 99 100 97 98 14 -11 13 15

Your output

✓ Testing with inputs: 101 312 541 120 1 2 17 98

Your output

1 2 17 98

