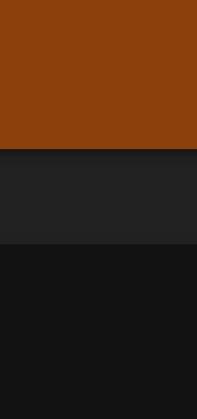


6.6 Iterating through vectors



Students:

Section 6.6 is a part of 2 assignments: CSC108 CH06.1-6.9 C6A ▾

Includes: CA

Due: 04/29/2025, 11:59 PM EDT

Please browse to this assignment through BlackboardLearn so zyBooks knows where to send your activity. [Learn more](#)

6.6 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 the annual GDP for 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 `ctryGDPs` 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.

Figure 6.6.1: Multiple vector example: Annual GDP for selected countries.

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

int main() {
    // Source: The World Bank, 2022
    vector<string> ctryNames; // Country names
    vector<int> ctryGDPs; // Country annual GDPs (trillions)
    string userCountry; // User-entered country
    bool foundCountry = false; // Country-match flag
    unsigned int i; // Loop index

    ctryNames.push_back("UK");
    ctryGDPs.push_back(3);

    ctryNames.push_back("India");
    ctryGDPs.push_back(3);

    ctryNames.push_back("Canada");
    ctryGDPs.push_back(2);

    ctryNames.push_back("Italy");
    ctryGDPs.push_back(2);

    ctryNames.push_back("Japan");
    ctryGDPs.push_back(4);

    cout << "Enter case-sensitive country to search for: ";
    cin >> userCountry;

    foundCountry = false;
    for (i = 0; i < ctryNames.size() && !foundCountry; ++i) {
        if (ctryNames.at(i) == userCountry) {
            foundCountry = true;
            cout << userCountry << " has an annual GDP of ";
            cout << "$" << ctryGDPs.at(i) << endl;
        }
    }

    if (!foundCountry) {
        cout << "Country not found; try again." << endl;
    }

    return 0;
}
```

Enter case-sensitive country to search for: Canada
Canada has an annual GDP of \$2T.
...
Enter case-sensitive country to search for: Japan
Japan has an annual GDP of \$4T.
...
Enter case-sensitive country to search for: Sweden
Country not found; try again.

Feedback?

PARTICIPATION ACTIVITY | 6.6.1: Multiple vectors.

Consider the above 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 6.6.1: Improve the program.

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

```
Load default template...
Italy
Run
```

Feedback?

CHALLENGE ACTIVITY | 6.6.1: Printing the sum of two vector elements.

Four elements are read into `origList` and four elements are read into `offsetAmount`. Add each element in `origList` with the corresponding value in `offsetAmount`. Print each sum followed by a space.

Ex: If the input is:

40 50 60 70 5 7 3 0

then `origList = {40, 50, 60, 70}` and `offsetAmount = {5, 7, 3, 0}`. So, the output is:

45 57 63 70

[Learn how our autograder works](#)

```
620890_5010016.qz3zy7
1 #include <iostream>
2 #include <vector>
3 using namespace std;
4
5 int main() {
6     const int NUM_VALS = 4;
7     vector<int> origList;
8     vector<int> offsetAmount;
9     unsigned int i;
10    int input;
11
12    for (i = 0; i < NUM_VALS; ++i) {
13        cin >> input;
14        origList.push_back(input);
15    }
16
17    for (i = 0; i < NUM_VALS; ++i) {
18        cin >> input;
Run
```

Feedback?

CHALLENGE ACTIVITY | 6.6.2: Multiple vectors: Key and value.

Four elements are read into `keysList` and four elements are read into `itemsList`. For any element in `keysList` with a value greater than 100, print the corresponding value in `itemsList`, followed by a space.

Ex: If the input is:

42 105 101 100 10 20 30 40

then `keysList = {42, 105, 101, 100}` and `itemsList = {10, 20, 30, 40}`. The elements at indices 1 and 2 of `keysList` are greater than 100. So, the elements at the same indices of `itemsList` are printed. Thus, the output is:

20 30

[Learn how our autograder works](#)

```
620890_5010016.qz3zy7
1 #include <iostream>
2 #include <vector>
3 using namespace std;
4
5 int main() {
6     const int SIZE_LIST = 4;
7     vector<int> keysList;
8     vector<int> itemsList;
9     unsigned int i;
10    int input;
11
12    for (i = 0; i < SIZE_LIST; ++i) {
13        cin >> input;
14        keysList.push_back(input);
15    }
16
17    for (i = 0; i < SIZE_LIST; ++i) {
18        cin >> input;
Run
```

Feedback?

CHALLENGE ACTIVITY | 6.6.3: Multiple vectors.

```
620890_5010016.qz3zy7
Start
```

Feedback?

Integer `numElements` is read from input. The remaining input alternates between strings and integers. Declare string vector `topicList` and integer vector `pageList`. Then:

- Read `numElements` pairs of strings and integers from input.
- Using `push_back()`, store each string into `topicList` and each integer into `pageList`.

[Click here for example](#) ▾

```
1 #include <iostream>
2 #include <vector>
3 using namespace std;
4
5 int main() {
6     int numElements;
7     unsigned int i;
8     string strngVal;
9     int intVal;
10
11    cin >> numElements;
12
13    /* Your code goes here */
14
15    for (i = 0; i < topicList.size(); ++i) {
16        cout << "Topic: " << topicList.at(i) << " ";
17        cout << "Page: " << pageList.at(i) << endl;
18    }
19
Run
```

Feedback?

How was this section?

Provide section feedback

Activity summary for assignment: CSC108 CH06.1-6.9 C6A ▾

Due: 04/29/2025, 11:59 PM EDT

Please browse to this assignment through BlackboardLearn so zyBooks knows where to send your activity. [Learn more](#)

Completion details ▾

0 / 40 points

Feedback?

1

2

3

Check

Next level

↓ 6.7 Vector resize