5.8 Loop-modifying or copying/comparing vectors

Modifying vector elements

A program may need to modify elements while iterating through a vector. The program below uses a loop to convert any negative vector element value to 0.

Figure 5.8.1: Modifying a vector during iteration example: Converting negatives to 0.

```
#include <iostream>
#include <vector>
using namespace std;
int main() {
   const int NUM ELEMENTS = 5;  // Number of elements
   vector<int> userVals(NUM ELEMENTS); // User values
   unsigned int i;
                                         // Loop index
   // Prompt user to populate vector
   cout << "Enter " << NUM_ELEMENTS << " integer values..." << endl;</pre>
   for (i = 0; i < userVals.size(); ++i) {</pre>
      cout << "Value: ";</pre>
      cin >> userVals.at(i);
   // Convert negatives to 0
   for (i = 0; i < userVals.size(); ++i) {</pre>
      if (userVals.at(i) < 0) {</pre>
         userVals.at(i) = 0;
   // Print numbers
   cout << "New values:";</pre>
   for (i = 0; i < userVals.size(); ++i) {</pre>
      cout << " " << userVals.at(i);</pre>
   cout << endl;</pre>
   return 0;
```

```
Enter 5 integer values...
Value: 67
Value: -5
Value: -99
Value: 4
Value: 22
New values: 67 0 0 4 22
```

Feedback?

PARTICIPATION ACTIVITY

5.8.1: Modifying a vector in a loop.

What is the resulting vector contents, assuming each question starts with a vector of size 4 having contents -55, -1, 0, 9?

- 1) for (i = 0; i < 4; ++i) {
 itemsList.at(i) = i;
 }</pre>
 - O -54, 0, 1, 10
 - **O** 0, 1, 2, 3
 - O 1, 2, 3, 4
- 2) for (i = 0; i < 4; ++i) {
 if (itemsList.at(i) < 0) {
 itemsList.at(i) =
 itemsList.at(i) * -1;
 }
 }</pre>
 - O -55, -1, 0, -9
 - **O** 55, 1, 0, -9
 - **O** 55, 1, 0, 9
- 3) for (i = 0; i < 4; ++i) {
 itemsList.at(i) =
 itemsList.at(i+1);
 }</pre>
 - **O** -1, 0, 9, 0
 - 0, -55, -1, 0
 - O Error (program aborts)
- 4) for (i = 0; i < 3; ++i) {
 itemsList.at(i) =
 itemsList.at(i+1);
 }</pre>
 - O -1, 0, 9, 9
 - O Error (program aborts)
 - O -1, 0, 9, 0
- 5) for (i = 0; i < 3; ++i) {
 itemsList.at(i+1) =
 itemsList.at(i);
 }</pre>
 - O -55, -55, -55, -55
 - 0, -55, -1, 0
 - O Error (program aborts)

zyDE 5.8.1: Modifying a vector during iteration example: Doubling element values.

Complete the following program to double each number in the vector.

```
67 -5 -99 4 22
                         Load default template...
1 #include <iostream>
2 #include <vector>
3 using namespace std;
                                                         Run
5 int main() {
       const int NUM ELEMENTS = 5;
       vector<int> userVals(NUM ELEMENTS); // \[ \]
8
       unsigned int i;
9
10
      // Prompt user to populate vector
       cout << "Enter " << NUM_ELEMENTS << " ir</pre>
11
       for (i = 0; i < userVals.size(); ++i) {</pre>
12
          cout << "Value: " << endl;</pre>
13
14
          cin >> userVals.at(i);
15
16
       // Convert negatives to 0
17
       for (i = 0; i < userVals.size(); ++i) {</pre>
18
19
          if (userVals.at(i) < 0) {</pre>
              11con/12 = 0+/il
20
21
                                                                          Feedback?
```

Element by element vector copy

In C++, the = operator conveniently performs an element-by-element copy of a vector, called a **vector copy operation**. The operation vectorB = vectorA resizes vectorB to vectorA's size, appending or deleting elements as needed. vectorB commonly has a size of 0 before the operation.

Figure 5.8.2: Using = to copy a vector: Original and sale prices.

Original prices: 10 20 30 40 Sale prices: 10 20 27 35

```
#include <iostream>
#include <vector>
using namespace std;
int main() {
   const int
                                            // Number of elements
                NUM ELEMENTS = 4;
   vector<int> origPrices(NUM_ELEMENTS); // Original prices
   vector<int> salePrices(NUM_ELEMENTS); // Sale prices
   unsigned int i;
                                             // Loop index
   // Assign original prices
   origPrices.at(0) = 10;
   origPrices.at(1) = 20;
   origPrices.at(2) = 30;
   origPrices.at(3) = 40;
   // Copy original prices to sales prices
   salePrices = origPrices;
   // Update salePrices. Note: does not affect origPrices
   salePrices.at(2) = 27;
   salePrices.at(3) = 35;
   // Output original and sale prices
   cout << "Original prices: "</pre>
   for (i = 0; i < origPrices.size(); ++i) {
  cout << " " << origPrices.at(i);</pre>
   cout << endl;</pre>
   cout << "Sale prices:</pre>
   for (i = 0; i < salePrices.size(); ++i) {</pre>
      cout << " " << salePrices.at(i);</pre>
   cout << endl;</pre>
   return 0;
```

Feedback?

PARTICIPATION ACTIVITY

5.8.2: Vector copy operation.

Assume vectors have been declared as follows and have been initialized as indicated in the comments:

```
vector<int> userVals(4); // {44, 55, 66, 77}
vector<int> newVals; // No elements yet
```

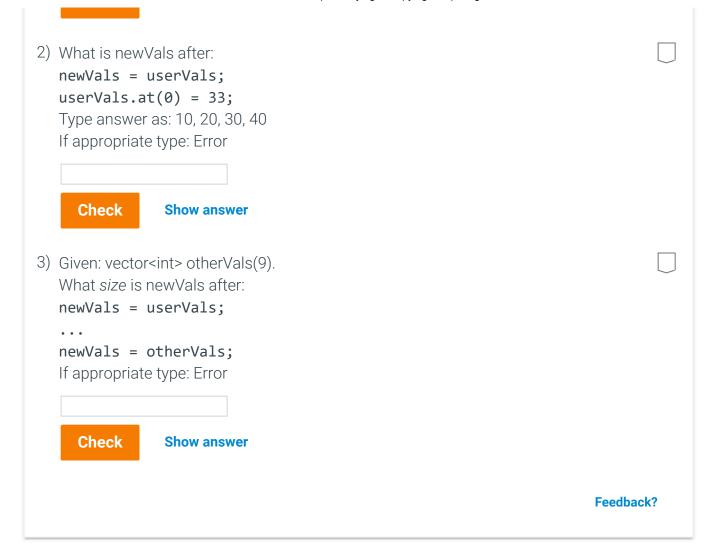
What is newVals after: newVals = userVals;

Type answer as: 10, 20, 30, 40

If appropriate type: Error

Check

Show answer



Element by element vector comparison

In C++, the == operator conveniently compares vectors element-by-element, called a **vector equality operation**, with vectorA == vectorB evaluating to true if the vectors are the same size AND each element pair is equal.

PARTICIPATION ACTIVITY 5.8.3: Vector copying.	
Assume vectors have been declared as follows and have been initialized as indicated in the comments:	
<pre>vector<int> vectorX(2); // {3,4} vector<int> vectorY(5); // {3,4,0,7,8} vector<int> vectorZ(5); // {3,4,0,6,8}</int></int></int></pre>	
1) (vectorX == vectorY) will evaluate to:	
O True	
O False	

CHALLENGE ACTIVITY

5.8.1: Decrement vector elements.



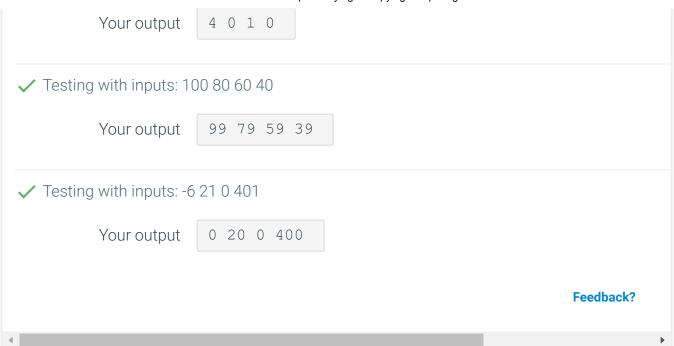
Write a loop that subtracts 1 from each element in lowerScores if the original element was greater than 0, and otherwise just assigns the element with 0. Ex: lowerScores = $\{5, 0, 2, -3\}$ becomes $\{4, 0, 1, 0\}$.

```
o using namespace stu,
4
5 int main() {
       const int SCORES SIZE = 4;
6
       vector<int> lowerScores(SCORES_SIZE);
       unsigned int i;
8
       for (i = 0; i < lowerScores.size(); ++i) {</pre>
10
11
          cin >> lowerScores.at(i);
12
13
       /* Your solution goes here */
14
15
       for (i = 0; i < lowerScores.size(); ++i) {</pre>
16
          if (lowerScores.at(i)>0)
17
18
             lowerScores.at(i)--;
19
             lowerScores.at(i) = 0;
20
21
22
23
       for (i = 0; i < lowerScores.size(); ++i) {</pre>
24
```

Run

✓ All tests passed

✓ Testing with inputs: 5 0 2 -3



CHALLENGE ACTIVITY

5.8.2: Copy and modify vector elements.

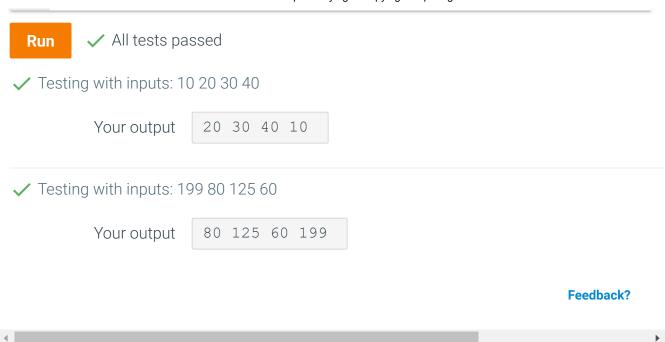


Write a loop that sets newScores to oldScores shifted once left, with element 0 copied to the end. Ex: If oldScores = {10, 20, 30, 40}, then newScores = {20, 30, 40, 10}.

Note: These activities may test code with different test values. This activity will perform two tests, both with a 4-element array. See "How to Use zyBooks".

Also note: If the submitted code tries to access an invalid array element, such as newScores[9] for a 4-element array, the test may generate strange results. Or the test may crash and report "Program end never reached", in which case the system doesn't print the test case that caused the reported message.

```
const int SCURES_SIZE = 4;
       vector<int> oldScores(SCORES_SIZE);
7
8
       vector<int> newScores(SCORES_SIZE);
9
      unsigned int i;
10
       for (i = 0; i < oldScores.size(); ++i) {</pre>
11
          cin >> oldScores.at(i);
12
13
14
15
       /* Your solution goes here */
       for (i = 0; i < oldScores.size()-1; ++i) {</pre>
16
          newScores.at(i) = oldScores.at(i+1);
17
18
       newScores.at(oldScores.size()-1) = oldScores.at(0);
19
20
       for (i = 0; i < newScores.size(); ++i) {</pre>
21
          cout << newScores.at(i) << " ";</pre>
22
23
24
       cout << endl;</pre>
25
       return 0;
```



CHALLENGE ACTIVITY

5.8.3: Modify vector elements using other elements.



Write a loop that sets each vector element to the sum of itself and the next element, except for the last element which stays the same. Be careful not to index beyond the last element. Ex:

Initial scores: 10, 20, 30, 40 Scores after the loop: 30, 50, 70, 40

The first element is 30 or 10 + 20, the second element is 50 or 20 + 30, and the third element is 70 or 30 + 40. The last element remains the same.

```
1 #include <iostream>
2 #include <vector>
3 using namespace std;
5 int main() {
      const int SCORES SIZE = 4;
6
      vector<int> bonusScores(SCORES_SIZE);
      unsigned int i;
8
9
      for (i = 0; i < bonusScores.size(); ++i) {</pre>
10
          cin >> bonusScores.at(i);
11
12
13
14
      /* Your solution goes here */
       for (i = 0; i < bonusScores.size()-1; ++i) {
15
          bonusScores.at(i) = bonusScores.at(i) + bonusScores.at(i+1);
16
17
18
19
      for (i = 0; i < bonusScores.size(); ++i) {</pre>
20
          cout << bonusScores.at(i) << " ";</pre>
21
```

```
All tests passed
✓ Testing with inputs: 10 20 30 40
          Your output
                         30 50 70 40

✓ Testing with inputs: 199 299 399 499

          Your output
                         498 698 898 499

✓ Testing with inputs: -100 -200 -300 -400

          Your output
                         -300 -500 -700 -400
                                                                          Feedback?
```

CHALLENGE ACTIVITY

5.8.4: Modify a vector's elements.



Subtract 4 to any element's value that is greater than maxVal. Ex: If maxVal = 10, then dataPoints = {2, 12, 9, 20} becomes {2, 8, 9, 16}.

```
4
5
   int main() {
      int maxVal;
6
      const int NUM POINTS = 4;
      vector<int> dataPoints(NUM_POINTS);
8
      unsigned int i;
10
11
      cin >> maxVal;
12
      for (i = 0; i < dataPoints.size(); ++i) {</pre>
13
14
          cin >> dataPoints.at(i);
15
16
       /* Your solution goes here */
17
18
       for (i = 0; i < dataPoints.size(); ++i) {</pre>
19
          if (dataPoints.at(i)> maxVal)
20
             dataPoints.at(i) = dataPoints.at(i)-4;
21
22
23
24
      }
```

Run

All tests passed

✓ Testing with inputs: 10 2 12 9 20

