Source File: `~/2336/38/lab38.(C|CPP|cpp|c++|cc|cxx|cp)`
Input: under control of `main` function
Output: under control of `main` function
Value: 3

A ***bucket sort*** begins with a one-dimensional array of positive integers to be sorted and a two-dimensional array of integers with rows subscripted from 0 to 9 and columns subscripted from 0 to $n - 1$, where n is the number of values in the array to be sorted. Each row of the two-dimensional array is referred to as a bucket. Write a function `bucketSort` that takes as arguments a vector of `unsigned int`egers and the maximum number of digits of a vector element and performs as follows:

a) Place each value of the one-dimensional array into a row of the bucket array based on the value's ones digit. For example, 97 is placed in row 7, 3 is placed in row 3 and 100 is placed in row 0. This is called a "distribution pass."

b) Loop through the bucket array row by row, and copy the values back to the original array. This is called a "gathering pass." The new order of the preceding values in the one-dimensional array is 100, 3, and 97.

c) Repeat this process for each subsequent digit position (tens, hundreds, thousands, etc.). On the second pass, 100 is placed in row 0, 3 is placed in row 0 (because 3 has no tens digit) and 97 is placed in row 9. After the gathering pass, the order of the values in the one-dimensional array is 100, 3, and 97. After the last gathering pass, the original array is now in sorted order.

The contents of the one-dimensional vector should be printed to the standard output device at the conclusion of each gathering pass.

Implement the buckets as a `vector` of `vector`s.

A sample `main` function for testing this function is shown in Figure 1. A sample execution sequence is shown in Figure 2. A second `main` function, similar to the one used for testing the other sort functions, is shown in Figure 3. To use this function, modify your sort function to eliminate the printing of the vector after each of the gathering passes. The execution sequence for this second `main` function is shown in Figure 4. To use the `Makefile` as distributed in class, add a target of `lab38` to `targets2srcfiles`.

```
1  #include <iostream>
2  #include <cstdlib> // contains prototypes for functions srand and rand
3  #include <vector>
4  #include <cmath>
5
6  using namespace std;
7
8  ostream& operator<<(ostream& os, const vector<uint>& v);
9
10 void bucketSort(vector<uint>& v, uint numDigits);
11
12 int main()
13 {
14   uint numDigits, n, shiftValue, scalingFactor, i;
15   vector<uint> v;
16
```

Figure 1. `/usr/local/2336/src/lab38main.C` (Part 1 of 2)

```
17    // randomize random number generator using current time
18    srand(time(0));
19
20    cout << "Enter the number of digits in each of the values to be sorted:"
21         << endl;
22    cin >> numDigits;
23
24    cout << "Enter the number of values to be sorted:" << endl;
25    cin >> n;
26
27    shiftValue = uint(pow(10.0, int(numDigits - 1)));
28    scalingFactor = uint(pow(10.0, int(numDigits))) - 1 - shiftValue;
29
30    for (i = 0; i < n; ++i)
31      v.push_back(shiftValue + rand() % scalingFactor);
32
33    cout << v << endl;
34    bucketSort(v, numDigits);
35
36    return 0;
37  }
38
39  ostream& operator<<(ostream& os, const vector<uint>& v)
40  {
41    vector<uint>::const_iterator itr;
42
43    os << "vector" << endl << '{' << endl;
44    for (itr = v.begin(); itr < v.end(); ++itr)
45      os << "  [" << itr - v.begin() << "] = " << *itr << endl;
46    os << '}' << endl;
47
48    return os;
49  }
```

Figure 1. `/usr/local/2336/src/lab38main.C` (Part 2 of 2)

```
newuser@csunix ~> cd 2336
newuser@csunix ~/2336> ./getlab.ksh 38
  * Checking to see if a folder exists for Lab 38. . .No
  * Creating a folder for Lab 38
  * Checking to see if Lab 38 has sample input and output files. . .No
  * Checking to see if /usr/local/2336/src/lab38main.C exists. . .Yes
  * Copying file /usr/local/2336/src/lab38main.C to folder ./38
  * Checking to see if /usr/local/2336/include/lab38.h exists. . .No
  * Copying file /usr/local/2336/src/Makefile to folder ./38
  * Adding a target of lab38 to targets2srcfiles
  * Touching file ./38/lab38.cpp
  * Edit file ./38/lab38.cpp in Notepad++
newuser@csunix ~/2336> cd 38
newuser@csunix ~/2336/38> ls
Makefile          lab38.cpp         lab38main.C       lab38main.C.test
newuser@csunix ~/2336/38> make lab38
g++ -g -Wall -std=c++11 -c lab38main.C -I/usr/local/2336/include -I.
g++ -g -Wall -std=c++11 -c lab38.cpp -I/usr/local/2336/include -I.
g++ -o lab38 lab38main.o lab38.o -L/usr/local/2336/lib -lm -lbits
newuser@csunix ~/2336/38> ./lab38
Enter the number of digits in each of the values to be sorted:
3
Enter the number of values to be sorted:
10
```

```
vector
{
  [0] = 331
  [1] = 294
  [2] = 125
  [3] = 664
  [4] = 556
  [5] = 220
  [6] = 745
  [7] = 470
  [8] = 469
  [9] = 449
}
```

```
vector
{
  [0] = 220
  [1] = 470
  [2] = 331
  [3] = 294
  [4] = 664
  [5] = 125
  [6] = 745
  [7] = 556
  [8] = 469
  [9] = 449
}
```

```
vector
{
  [0] = 220
  [1] = 125
  [2] = 331
  [3] = 745
  [4] = 449
  [5] = 556
  [6] = 664
  [7] = 469
  [8] = 470
  [9] = 294
}
```

```
vector
{
  [0] = 125
  [1] = 220
  [2] = 294
  [3] = 331
  [4] = 449
  [5] = 469
  [6] = 470
  [7] = 556
  [8] = 664
  [9] = 745
}
```

Figure 2. Commands to Compile, Link, & Run Lab 38 (Part 1 of 2)

```
newuser@csunix ~/2336/38> ./lab38
Enter the number of digits in each of the values to be sorted:
4
Enter the number of values to be sorted:
10
```

```
vector
{
  [0] = 7265
  [1] = 3039
  [2] = 5678
  [3] = 3581
  [4] = 3955
  [5] = 9075
  [6] = 5897
  [7] = 2043
  [8] = 6465
  [9] = 2703
}
```

```
vector
{
  [0] = 3581
  [1] = 2043
  [2] = 2703
  [3] = 7265
  [4] = 3955
  [5] = 9075
  [6] = 6465
  [7] = 5897
  [8] = 5678
  [9] = 3039
}
```

```
vector
{
  [0] = 2703
  [1] = 3039
  [2] = 2043
  [3] = 3955
  [4] = 7265
  [5] = 6465
  [6] = 9075
  [7] = 5678
  [8] = 3581
  [9] = 5897
}
```

```
vector
{
  [0] = 3039
  [1] = 2043
  [2] = 9075
  [3] = 7265
  [4] = 6465
  [5] = 3581
  [6] = 5678
  [7] = 2703
  [8] = 5897
  [9] = 3955
}
```

```
vector
{
  [0] = 2043
  [1] = 2703
  [2] = 3039
  [3] = 3581
  [4] = 3955
  [5] = 5678
  [6] = 5897
  [7] = 6465
  [8] = 7265
  [9] = 9075
}
```

```
newuser@csunix ~/2336/38>
```

Figure 2. Commands to Compile, Link, & Run Lab 38 (Part 2 of 2)

```
#include <cmath>
#include <cstdlib>
#include <iostream>
#include <vector>
#include <algorithm>
#include <chrono>
#include <random>

using namespace std;

void bucketSort(vector<uint>& v, uint numDigits);

const int N = 1000000;
enum TEST_TYPE {RANDOM, ASCENDING, DESCENDING};

int main()
{
  vector<uint> v, w, x;
  default_random_engine ran;
  uniform_int_distribution<> dis; // [1,UINT_MAX]
  TEST_TYPE testType;
  int i;

  for (testType = RANDOM;
       testType <= DESCENDING;
       testType = static_cast<TEST_TYPE>(testType + 1))
  {
    if (!v.empty())
      v.clear();
    switch (testType)
    {
      case RANDOM:
        for (i = 0; i < N; ++i)
          v.push_back(dis(ran));
        cout << "Random Data:" << endl;
        break;
      case ASCENDING:
        for (i = 0; i < N; ++i)
          v.push_back(i);
        cout << "Ascending Data:" << endl;
        break;
      case DESCENDING:
        for (i = 0; i < N; ++i)
          v.push_back(N - i);
        cout << "Descending Data:" << endl;
        break;
    }
```

Figure 3. `/usr/local/2336/src/lab38main.C.test` (Part 1 of 2)

```
48      x = v;
49      sort(x.begin(), x.end());
50
51      w = v;
52      auto start = chrono::system_clock::now();
53      bucketSort(w, 10);
54      auto stop = chrono::system_clock::now();
55      cout << "Bucket Sort: "
56           << chrono::duration_cast<chrono::milliseconds>(stop-start).count()
57           << "ms" << endl;
58      if (x != w)
59        cout << "Sort didn't work correctly" << endl;
60      cout << endl;
61    }
62    return EXIT_SUCCESS;
63  }
```

Figure 3. `/usr/local/2336/src/lab38main.C.test` (Part 2 of 2)

```
1  newuser@csunix ~/2336/38> mv lab38main.C.test lab38main.C
2  newuser@csunix ~/2336/38> # Edit lab38.cpp to eliminate the printing of the vector
3  newuser@csunix ~/2336/38> # after each of the gathering passes
4  newuser@csunix ~/2336/38> make lab38
5  g++ -g -Wall -std=c++11 -c lab38main.C -I/usr/local/2336/include -I.
6  g++ -g -Wall -std=c++11 -c lab38.cpp -I/usr/local/2336/include -I.
7  g++ -o lab38 lab38main.o lab38.o -L/usr/local/2336/lib -lm -lbits
8  newuser@csunix ~/2336/38> ./lab38
9  Random Data:
10 Bucket Sort: 339ms
11
12 Ascending Data:
13 Bucket Sort: 340ms
14
15 Descending Data:
16 Bucket Sort: 336ms
17
18 newuser@csunix ~/2336/38>
```

Figure 4. Commands to Compile, Link, & Run Lab 38