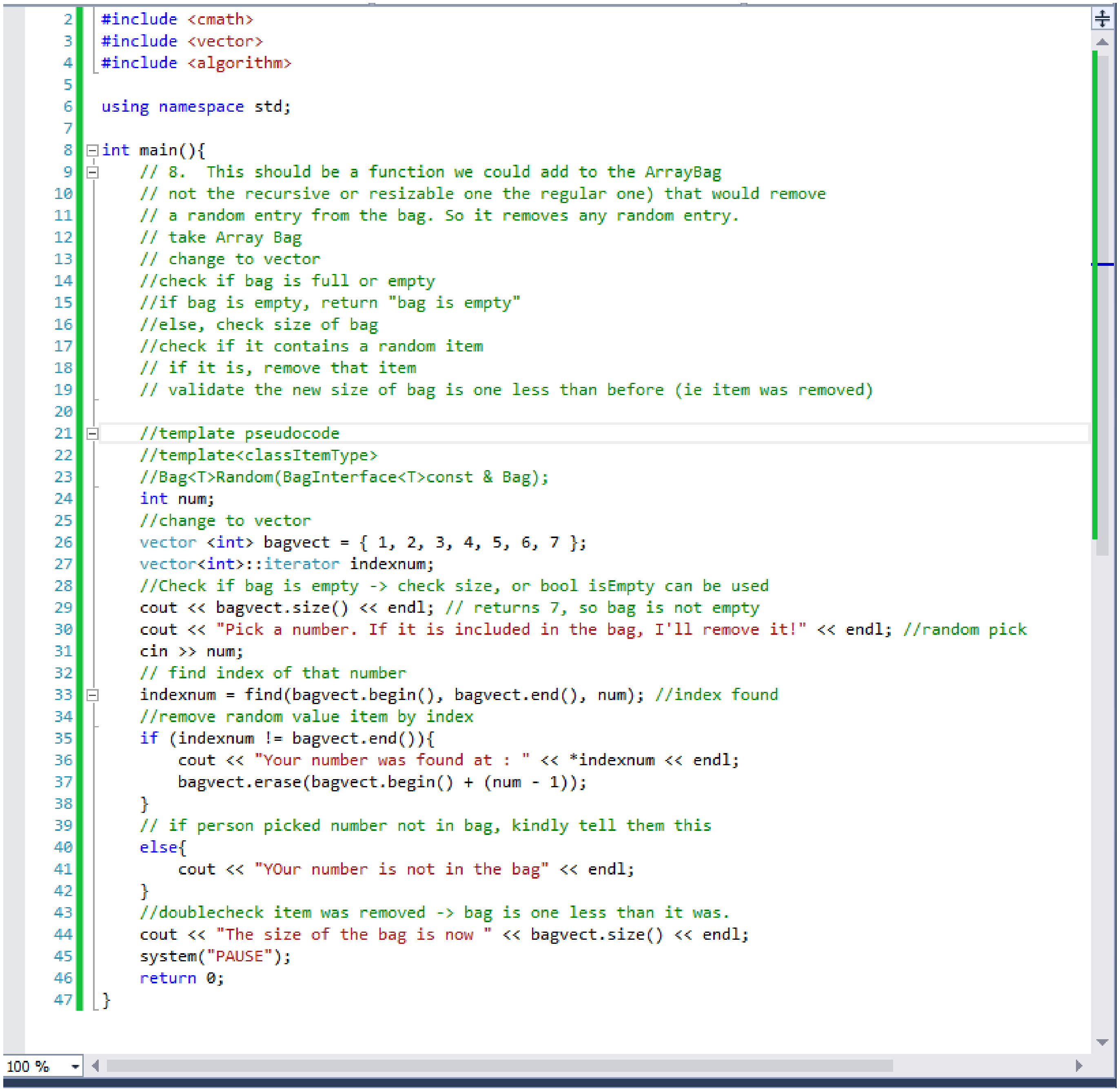
CSIS 211 -> Krystal Maughan -> Homework 4 # 8

Class Random

+BagSize
+BagItemRemove
+BagIndex
+BagIsEmpty
+BagToVector

Private:
-int num
-int indexnum



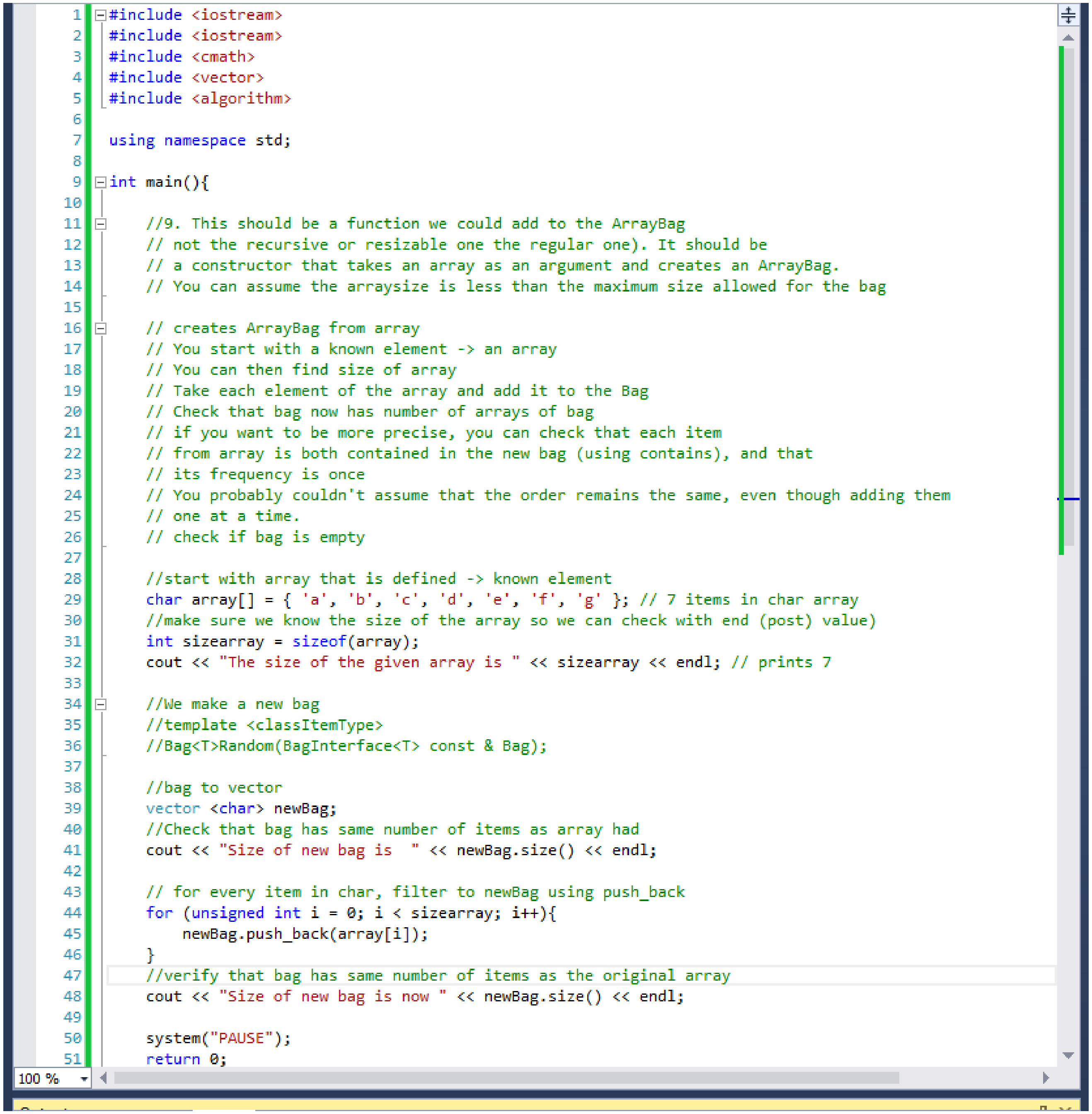
```
#include <cmath>
#include <vector>
#include <algorithm>

using namespace std;

int main(){
    // 8.  This should be a function we could add to the ArrayBag
    // not the recursive or resizable one the regular one) that would remove
    // a random entry from the bag. So it removes any random entry.
    // take Array Bag
    // change to vector
    //check if bag is full or empty
    //if bag is empty, return "bag is empty"
    //else, check size of bag
    //check if it contains a random item
    // if it is, remove that item
    // validate the new size of bag is one less than before (ie item was removed)

    //template pseudocode
    //template<classItemType>
    //Bag<T>Random(BagInterface<T>const & Bag);
    int num;
    //change to vector
    vector <int> bagvect = { 1, 2, 3, 4, 5, 6, 7 };
    vector<int>::iterator indexnum;
    //Check if bag is empty -> check size, or bool isEmpty can be used
    cout << bagvect.size() << endl; // returns 7, so bag is not empty
    cout << "Pick a number. If it is included in the bag, I'll remove it!" << endl; //random pick
    cin >> num;
    // find index of that number
    indexnum = find(bagvect.begin(), bagvect.end(), num); //index found
    //remove random value item by index
    if (indexnum != bagvect.end()){
        cout << "Your number was found at : " << *indexnum << endl;
        bagvect.erase(bagvect.begin() + (num - 1));
    }
    // if person picked number not in bag, kindly tell them this
    else{
        cout << "YOur number is not in the bag" << endl;
    }
    //doublecheck item was removed -> bag is one less than it was.
    cout << "The size of the bag is now " << bagvect.size() << endl;
    system("PAUSE");
    return 0;
}
```

CSIS 211 -> Krystal Maughan -> Homework 4 # 9

Class ChartoBag

+BagSize
+BagIsEmpty
+BagToVector

Private:
-char array
-int num
-int indexnum



```cpp
#include <iostream>
#include <iostream>
#include <cmath>
#include <vector>
#include <algorithm>

using namespace std;

int main(){

    //9. This should be a function we could add to the ArrayBag
    // not the recursive or resizable one the regular one). It should be
    // a constructor that takes an array as an argument and creates an ArrayBag.
    // You can assume the arraysize is less than the maximum size allowed for the bag

    // creates ArrayBag from array
    // You start with a known element -> an array
    // You can then find size of array
    // Take each element of the array and add it to the Bag
    // Check that bag now has number of arrays of bag
    // if you want to be more precise, you can check that each item
    // from array is both contained in the new bag (using contains), and that
    // its frequency is once
    // You probably couldn't assume that the order remains the same, even though adding them
    // one at a time.
    // check if bag is empty

    //start with array that is defined -> known element
    char array[] = { 'a', 'b', 'c', 'd', 'e', 'f', 'g' }; // 7 items in char array
    //make sure we know the size of the array so we can check with end (post) value)
    int sizearray = sizeof(array);
    cout << "The size of the given array is " << sizearray << endl; // prints 7

    //We make a new bag
    //template <classItemType>
    //Bag<T>Random(BagInterface<T> const & Bag);

    //bag to vector
    vector <char> newBag;
    //Check that bag has same number of items as array had
    cout << "Size of new bag is  " << newBag.size() << endl;

    // for every item in char, filter to newBag using push_back
    for (unsigned int i = 0; i < sizearray; i++){
        newBag.push_back(array[i]);
    }
    //verify that bag has same number of items as the original array
    cout << "Size of new bag is now " << newBag.size() << endl;

    system("PAUSE");
    return 0;
```