Lab 1

1. This assignment tests the concepts of:
    - Containers
    - Static arrays

Program Objective:

Implement 2 non-member functions and one member function.

1. The provided driver code reads in a bunch of integers from the command line and places them into a bag container.
2. We will explore two ways to remove all items in the bag
    1. The authors bag has only TWO ways to remove items, erase and erase_one. Both of these functions require you to send the target number you want removed.
    2. Problem: You need to empty all items from the bag with NO KNOWLEDGE of what numbers are, only knowing what the smallest and largest values are.
        - Implement `emptyBag(bag& nums, const stats& bagStats)` which loops over calling the erase member function on the bag until empty. On each iteration pick a random number min >= x <= max. Do not worry about trying to remove the same random number more than once.
    3. Problem: emptyBag took unnecessary runtime. Fix this by implementing a member function in the authors bag named grab_random. The function should REMOVE one item from the bag each time and return the number removed. Implement `emptyBag(bag& nums)` to take advantage of this new member function you implemented.
3. Think about, what is the BigOh of each function.

DELIVERABLES:

Description	**filename**
Your implementation of the 3 functions	**lab01.cpp**

You should not modify labdriver.cpp, bag1.h, bag1.cpp, or baglab.cpp.

Submitted assignments without the above file named correctly will render your assignment as uncompilable and will be detrimental to your assignment grade.

Instructions:

1. The provided driver is expecting command line arguments. It is best to add arguments to your IDE of choice.
    1. The program should compile, however, the harder part is adding command-line arguments to your favorite IDE.
        - For Eclipse CDT
            - Right click the project and choose properties
            - Go to Run/Debug Settings
            - Choose your launch configuration and click “Edit”
            - Click the “Arguments” tab
            - Type in some numbers separated by a space

- For Microsoft Visual Studio
  - Right click the project and choose properties
  - Go to “Configuration Properties->Debugging”
  - On the right hand side, type in some numbers separated by a space in the “Command Arguments” text area.
- For CLion
  - Open the menu “Run->Edit Configurations…”
  - Type in some numbers separated by a space in the “program arguments” text area.
- For Dev++
  - Stop using Dev++ and use one of the above IDE’s

2. Once you add the command line arguments to your IDE, these values will be appended to the program executable when executed (ie: run).

Additional requirements:

- No STL containers may be used. Include only iostream, cstdlib, ctime, and algorithm (if you wish to use copy).

Example output (from multiple runs with different arguments given):

```
$ ./lab02 1 2 4 8 16 2 8 256 0 255
All 10 items removed in 308 tries.
All 10 items removed in 10 tries.

$ ./lab02 1 -50 1256
All 3 items removed in 1535 tries.
All 3 items removed in 3 tries.
```

Help

Starter shell for lab01.cpp

```
#include <cassert>  //For assert
#include <cstdlib>  //For rand
#include "bag1.h"
#include "baglab.h"

namespace DS {

bag::value_type bag::grab_random()
{
    //Remove a random item in the bag

    return value_type(); //Remove this line
}
```

```
size_t emptyBag(DS::bag& nums, const DS::stats& bagStats) {
    //Implement assuming grab_random does NOT exist
    //Use stats struct to know the max and min
    //Use cstdlib's rand for random function

    return 0; //Remove this line
}

size_t emptyBag(DS::bag& nums) {
    //Implement assuming member grab_random exist

    return 0; //Remove this line
}

} //End of DS namespace
```