

## fCPSC 2600 Fall 2021 Homework Assignment #3

Due date: Oct 29, 2021 11:59pm

Write and use a C++ class called `IntSet` that models a set of positive integers as bit strings (also known as bit vectors). The universal set is  $\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$  so you will use 10-bit strings to represent the subsets.

### Requirements and reminders:

- The use of STL, templates, `bitset`, and operator overloading is not permitted in any form.
- Remember: You must use a bit vector (`int`) when representing a set. You may not use an array of Boolean variables, strings, or any other data structure.
- Programs that fail to compile will receive a grade of 0. You are allowed to re-submit your program before the due date and time, but only the last submission submitted before the due date and time will be graded. **Late submissions are not accepted and result in a zero.**

**Part 1.** Implement the class by creating the following public member functions. The underlying class must use a bit string as discussed in lecture.

- A default constructor. The set is initially a null set.
- `void addElement(int num):` Adds the specified integer to the set if not already an element.
- `void removeElement(int num):` Removes the specified integer if it is in the set. Note that nothing happens if the element is in the domain but not in the set.
- `bool contains(int num) const:` Returns true if the specified integer is in the set. Returns false otherwise.
- `void printSet() const:` Displays the set to the screen in the format:  $\{1, 3, 4\}$   
Note: The elements must appear in a comma separated list in ascending order and there should not be a trailing comma after the last entry. The empty set should be specified as  $\{\}$ .
- `void printBit() const:` Displays the set to the screen in the format: 1011000000
- `IntSet unions(const IntSet& operand) const:` Returns a set that is the union of the invoking set and the set operand.
- `IntSet intersect(const IntSet& operand) const:` Returns a set that is the intersection of the invoking set and the set operand.
- `IntSet difference(const IntSet& operand) const:` Returns a set that is the difference between the invoking set (call it A) and the set operand (call it B). In other words, compute  $A - B$ .
- `IntSet complement() const:` Returns the complement of the set.

You can assume that integer argument to `addElement`, `removeElement`, or `contains` is within the domain of the set.

Remember to think about how to print to the screen all elements of a set from a bit string. You may need the right shift operator (`>>`), the bitwise-and (`&`), the bitwise-or (`|`), an auxiliary bit string for the purpose of “mask”, and a loop to complete the functions.

**Part 2.** Use the `IntSet` class to write a program that performs the following steps:

1. Create the following sets with the universal domain:
  - a.  $A = \{x \mid x \text{ is an even number}\}$
  - b.  $B = \{x \mid x \text{ is a number divisible by } 3\}$
  - c.  $C = \{1, 2, 4, 6, 9\}$
2. Compute the following sets:
  - a.  $D = \overline{A}$
  - b.  $E = A \cup B$
  - c.  $F = B \cap C$
  - d.  $G = A - B$
  - e.  $H = B \cup (A \cap C)$
  - f.  $I = (A \cup B) - C$
3. Print to the screen the sets A through C using `printBit` (print the name of the sets as well)
4. Print to the screen the sets D through I using `printSet`.
5. Print to the screen either True or False:
  - a.  $6 \in H$ ?
  - b.  $2 \in D$ ?

The program requires no user input – simply create the sets, perform the calculations, and display the results. You must perform the listed set operations and comparisons using your `IntSet` class.

Do not consider completion of part 2 adequate testing of your class in part 1. You should write other tests to make sure your class is working correctly.

### Code Organization

You will use three files for this assignment

- The class definition must be stored in a file called `'intSet.h'`.
- The class implementation must be stored in a file called `'intSet.cpp'`.
- The code from part 2 (which includes `main`) must be stored in a file called `'hw3.cpp'`.

You can copy starting versions of these files by running:

```
cp /home/fac/hkong/cpsc2600/hw3/* .
```

The file `intSet.h` contains the class definition and the member function interfaces. Do NOT alter the member function interfaces. You can add data members and additional private member functions if appropriate.

The file `intSet.cpp` should the function implementations. You can also add additional private member functions.

Write all the tests in the `hw3.cpp` file.

To compile the program (creating executable `a.out`), use:

```
g++ hw3.cpp intSet.cpp
```

## Grading

Your grade will be computed as follows:

Default constructor	2 points
Correct output for Part 2 (requires Part 1 to work)	10 points
<code>displaySet</code> format correct (sorted, proper use of commas, etc.)	4 points
<code>displayBit</code> format correct	4 points
<code>addElement / removeElement</code> test	6 points
<code>contains</code> test	5 points
<code>unions</code> test	5 points
<code>intersection</code> test	5 points
<code>difference</code> test	5 points
<code>complement</code> test	4 points
<b>Total</b>	<b>50 points</b>

Tests that pass will receive full credit and tests that fail will receive no credit (0 points) unless it is a formatting problem with `displaySet` (in which case you will lose points for display but still receive full credit for the test). Note that all tests depend on the constructor, `addElement`, and `displaySet` functions working correctly.

The following additional penalties are possible:

- If you fail to use a bit vector, you will lose 15 points regardless of whether the program is working correctly.
- If the program is very unreadable, you will receive a deduction up to 10 points.

## Submitting your Program

On `cs1`, run the following script in the directory with your program:

```
/home/fac/hkong/submit/cpsc2600/hw3_submit
```