

# Assignment 2

CPSC 2620

Department of Mathematics and Computer Science  
University of Lethbridge

February 4, 2021

## 1 Description

**Assignment 2 is due on Monday Feb 15, 2021 by 23:55.**

**Assignment 2 is worth 4% of your final grade.**

Please refer to the description of balanced ternary numbers in Assignment 1.

In Assignment 2 you are asked to complete the definition, implementation and testing of class `BTernary`.

As you already know, a ternary number (as specified in the assignment) is input/output as a `C++` string while stored internally as an integer. As such, the class `BTernary` needs two conversion tasks, namely,

1. `ternaryTodecimal`
2. `decimalToternary`

Consider the ternary number  $(-11011)_3$ . Its value in decimal system is computed by multiplying each ternary digit (henceforth *trit*) by its positional value and then summing them.

$$(-1) \times 3^4 + 1 \times 3^3 + 0 \times 3^2 + 1 \times 3^1 + 1 \times 3^0 = (-50)_{10}$$

On the other hand, given a decimal number, its equivalent ternary number can be found by algorithm `DECIMAL2TERNARY`. The algorithm is given in pseudocode where,

- the assignment operation is indicated by symbol  $\leftarrow$
- any text appearing after symbol  $\triangleright$  up to the end of line is considered comment.

## DECIMALTOTERNARY

```

1  Let  $d$  be a decimal number to be converted to balanced ternary
2  Let  $s$  be a string variable,  $trit$  and  $temp$  be integer variables, and  $negative$  be a Boolean variable
3   $temp \leftarrow \text{abs}(d) \triangleright$  Get the absolute value.
4   $negative \leftarrow \text{False}$ 
5  if  $d < 0$ 
6      then  $negative \leftarrow \text{True}$ 
7  while  $temp > 0$ 
8      do
9           $trit \leftarrow temp \text{ modulo } 3$ 
10         if  $trit$  is 0
11             then  $s \leftarrow "0" + s \triangleright$  append character 0 to the front of string  $s$ 
12                  $temp \leftarrow temp / 3$ 
13         else if  $trit$  is 1
14             then if  $negative$ 
15                 then  $s \leftarrow "-1" + s \triangleright$  append the string "-1" to the front of string  $s$ 
16                 else
17                      $s \leftarrow "1" + s \triangleright$  append character 1 to the front of string  $s$ 
18                      $temp \leftarrow temp / 3$ 
19         else if  $trit$  is 2
20             then if  $negative$ 
21                 then  $s \leftarrow "1" + s$ 
22                 else
23                      $s \leftarrow "-1" + s$ 
24
25                  $temp \leftarrow (temp + 1) / 3$ 

```

## 2 Complete Specification for class BTernary

Given below is the complete specification for class BTernary. Code examples are used to illustrate the essential properties of the class objects.

1. Implement two different constructors to create BTernary class objects in a consistent state.

```

string s1 = "10-1-1";
string s2 = "41-10";
BTernary a, b(s1), c(s2); // create objects of class type

```

**Meaning.** Object *a* is created and the data member `decimal_equivalent` is initialized with value 0. Object *b* is a valid BTernary number whose decimal value is 23; the data member `decimal_equivalent` is initialized with value 23. Object *c* is an invalid BTernary number and the constructor should give an error message and the data member `decimal_equivalent` be initialized with value 0.

2. Other member functions you need to implement in class BTernary are as illustrated below

(a)            `BTernary d(s1);`  
              `a = b.plus(d);`

**Meaning.** `BTernary` class object `a` contains the sum of `BTernary` class objects `b` and `d`.

(b)            `a = b.minus(d);`

**Meaning.** `BTernary` class object `a` contains the result of `BTernary` class object `d` subtracted from `BTernary` class object `b`.

(c)            `a = b.times(d);`

**Meaning.** `BTernary` class object `a` contains the product of `BTernary` class objects `b` and `d`.

(d)            `a.insert(cout);`

**Meaning.** `BTernary` class object `a` is displayed on the terminal (standard output).

(e)            `c.extract(cin);`

**Meaning.** `BTernary` class object `c` is read from the keyboard (standard input).

- (f) Implement a member function called `ternaryTodecimal` that takes a C++ string (e.g., "-11011") as parameter and interprets it as a balanced ternary number and returns its decimal value. (read discussion in Section 1)
- (g) Implement a member function called `decimalToternary` that takes a decimal number as parameter and returns its balanced ternary number representation as a C++ string using algorithm `DECIMALTOTERNARY`.
- (h) Implement a member function called `isTernary` that takes a C++ string as parameter and returns true if the passed argument is a valid ternary number. The function returns false if the passed argument is not a valid ternary number.

### 3 What to submit

1. Class definition for `BTernary` in a text file named `BTernary.h` that meets the specification in Section 2. Give brief but convincing arguments, as comments, for your choice of class member access restriction (i.e., public/private).
2. In a file named `BTernary.cc`, implement the member functions of class `BTernary`.
3. Write a test program in a file named `test_BTernary.cc` to test the correctness of implemented member functions (see, for example, `Rational` class test program).

### 4 How to submit

There will be a link on CPSC2620 moodle page to upload your code. Information will be provided.

## 5 Grading

The assignment will be graded as follows.

1. The program is complete and compiles without errors and warnings: 20 points.
2. The class definition and implementation meets the specification: 50 points.
3. The explanation for the choice of members' access-restriction is convincing : 10 points.
4. Program is appropriately commented and indented: 10 points.
5. The file `test_BTernary.cc` contains code that demonstrates the correct working of each implemented member function: 10 points.

Total: 100 points