

2025.2 Object Oriented Programming, Project #2

(Due : November 2 (Sunday), 11:59pm)

Submission

Case 1. If you are supposed to do project 2 as team project:

```
[ class 01: team 5, 6, 7, 8, class 02: team 5, 6, 7, 8, class 03: team 3, 4 ]
```

(see our class webpage for team assignment)

- Only the team leader should submit the project results (including source code files, execution demo video file in .mp4 format [length: around 2 minutes or more] and project report) through eclass system. A team member should give presentation. You should use English for project presentation and project report.

Case 2. If you are supposed to do project 1 as an individual project or your team size is 1,

[If you do not belong to case 1, that means you belong to case 2.]

- eClass submission - you should submit one zip file that contains two directories as follows
 - (i) directory "prob1"
 - compilable source code package (inf_int.h, inf_int.cpp, main.cpp, ...),
 - readme.txt: briefly explain how to compile and execute your code.
 - demo video file (.mp4 format, around 2 minutes) showing execution of the test programs.
 - (ii) directory "prob2" - this directory should contain
 - source code package (inf_int.h, inf_int.cpp, main.cpp, ...),
 - readme.txt: briefly explain how to compile and execute your code.
 - demo video file (.mp4 format, around 2 minutes) showing execution of the calculator program.

Problem1. Complete implementing `inf_int` class that represents infinite precision integer and its operations. The specification of `inf_int` is provided to you in "`inf_int.h`". You should add its implementation in "`inf_int.cpp`". Test the correctness of your implementation by writing "`main.cpp`". "`inf_int.h`" and an example of "`main.cpp`" are downloadable from our class website. After compilation, your code should generate correct result. The result of this problem should be submitting compilable source code package (`inf_int.h`, `inf_int.cpp`, `main.cpp`, `.sln` files (files for opening with visual studio 20xx)). FYI, I will test your code by replacing the "`main.cpp`" with my own `main.cpp` files that contain various test cases and checking whether your code generates correct result according to my own `main.cpp`. You have to effectively use dynamic memory allocation(`new`) and deallocation(`delete`) for handling the variable sizes of infinite precision integer numbers.

Problem2. Write a calculator program using the C++ files (inf_int.h and inf_int.cpp) you made for problem1. Your calculator program should support addition(+), subtraction(-), and multiplication(*) of infinite precision integers. Your calculator should follow input and output formats as shown below.

Keyboard Input Format : After executing your program on command line, your program prints “Input:” and wait for a user keyboard input. The user keyboard input should be an expression that has a format like :

(positive integer)(space)(operator)(space)(positive integer)

After a user finishes the keyboard input and presses an enter key, your program should print the calculation result of the input expression. The program repeat this input and output processes. The program is terminated when 0 is given as an input.

Execution Example:

```
> prob2.exe
```

```
Input: 345 - 1111111111111111111111111111111111111111111111
```

Output: -1110766

Input: 33333333333333333333333333333333 * 73755

Output: 245849999999999999999999999975415

Input: 2555555555551111 + 2111111111

Output: 25555557666662222

Input: 0

 \triangleright