# Assignment 1 (COMP SCI 2201 and COMP SCI 7201)

**Due:** 23:59 PM 1st April 2018, late submissions will have penalties.

**Important**

- This is an individual assignment.
- This assignment has two parts. Part 1 must be completed by all the students (COMP SCI 2201 and COMP SCI 7201) and Part 2 is **only** for the postgraduate students (COMP SCI 7201).

## Part I

# C++ Implementation of Integer Addition and Multiplication

The first part of the assignment will contribute to 5% of your final grade.

## 1 Task Description

You are asked to use C++ to implement:

- School Method for Integer Addition
- Karatsuba Algorithm for Integer Multiplication

**You must follow this guideline!! Your submission will be marked automatically. Failure to follow this guideline will result in 0.**

### 1.1 Submission Guideline

- You are given the skeletal implementation for a program that includes: i) main method with argument parsing (`main.cpp`); ii) type definitions (`typedef.h`); iii) several utility functions (`utils.cpp` and `utils.h`) and iv) a header file with method signatures for school method addition, Karatsuba multiplication (`operations.h`) in the Canvas. The functionalities of the given methods are described in the respective comments.
- **You cannot change any of the provided files. You are required to provide implementations for the methods signatures provided in the `operations.h` file**. Please submit the implementations for the required functions in your own source file(s) having the file extension with `.cpp`. You do not need to submit a design.

- Your program takes one line as input. The input line contains three integers separated by spaces. Let the three integers be ‘I1 I2 B’. I1 and I2 are both non-negative integers up to 100 digits long. B represents I1 and I2’s base (B is from 2 to 10).[1]
- The code for parsing the arguments and displaying the final result is already written in the `main.cpp`.

  The output will include 2 integers: i) the sum of I1 and I2, using the school method; and ii) the product of I1 and I2, using the Karatsuba algorithm. The results will still use base B.
  Sample input 1: 101 5 10

  Sample output 1: 106 505

  Sample input 2: 10 111 2

  Sample output 2: 1001 1110

## 2 Marking

Marking will be done automatically. The total mark is 5 for the first part of this assignment. We trust that you’d indeed implement the school method for addition and the Karatsuba algorithm for multiplication. We will randomly pick submissions to verify whether it is the case. If you use other methods in your submission, it will be considered cheating and you will receive 0.

- 1 marks for correct compilation
- 4 marks for correct outputs for randomly generated integer inputs

## 3 SVN Instructions

First of all, you need to create a directory under version control:

```
svn mkdir --parents -m "Creating ADSA Assignment 1 folder" https://version-control.adelaide.edu.au/svn/aXXXXXXX/2018/s1/adsa/assignment1/
```

aXXXXXXX should be your student ID. The directory path needs to be exactly “2018/s1/adsa/assignmentK”, where “K” is the assignment number.

To check out a working copy, type

```
svn checkout https://version-control.adelaide.edu.au/svn/aXXXXXXX/2018/s1/adsa/assignment1/ adsa-18-s1-assignment1/
cd adsa-18-s1-assignment1

svn add *.cpp

svn add *.h
```

Commit the files to SVN:

```
svn commit -m "Adding ADSA assignment 1"
```

SVN helps keeping track of file changes (over different commits). You should commit your work early and often.

[1] This makes your life easier as you do not have to deal with numbers like ‘aff’.

## 4 Web-submission

- You are asked to submit via the web interface `https://cs.adelaide.edu.au/services/websubmission/`. The submission steps should be self-explanatory. Simply choose the correct semester, course, and assignment. The websubmission system will automatically fetch the latest version of your work from your SVN repository (you may also choose to submit older versions). Once your work is submitted, the system will launch a script checking the format of your submission. Click "View Feedback" to view the results. You are welcome to resubmit for as many times as you wish (before the deadline).

- During marking we will be replacing the provided files (`main.cpp`, `typedef.h`, `utils.cpp`, `utils.h`, `operations.h`) with their original versions to ensure that the provided files have not been changed.

- Your code will be complied using the command `g++ -o main -std=c++11 *.cpp` It is your responsibility to ensure that your code compiles **on the university system**.[2]

## Part II

# Complexity notation

**Important note:**

- This should be ONLY completed by **POSTGRADUATE students (COMP SCI 7201)**. This part of the assignment will contribute to 5% of your final grade.

- **Your answers to the part two of this assignment should be submitted to the CANVAS on or before the deadline.**

1. Prove following statements (3 marks):

   (a) $cf(n) \in \Theta(f(n))$ for $c > 0$

   (b) $f(n) + g(n) \in \Omega(f(n))$

   (c) $f'(n) \cdot g'(n) \in O(f(n) \cdot g(n))$ if $f'(n) \in O(f(n))$ and $g' \in O(g(n))$

2. Given $p(n) = \sum_{i=0}^{k} a_i n^i \in \Theta(n^k)$ for $a_k > 0$, then prove that $p(n) = a_k n^k + o(n^k)$ (1 mark).

3. Given $f(n) \in \Theta(g(n))$ and $g(n) \in \Theta(h(n))$, then prove that $h(n) \in \Theta(f(n))$ (1 mark).

[2] g++ has too many versions, so being able to compile on your laptop does not guarantee that it compiles on the university system. You are encouraged to debug your code on a lab computer (or use SSH).