C++ Implementation of Integer Addition and Multiplication (worth 5%, due Aug 18th 23:59PM, late submissions not accepted)

Mingyu Guo

1 Task Description

You are asked to use C++ to implement

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

2 Submission Guideline

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

Your submission should contain exactly two files: `main.cpp` and `test.txt`. 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 nonnegative integers up to 100 digits long. B represents I1 and I2's base (B is from 2 to 10).[1]

Your program should output the sum of I1 and I2, using the school method, then the product of I1 and I2, using the Karatsuba algorithm. The results should still use base B. Please separate the results using one space.

Sample input 1: 101 5 10
Sample output 1: 106 505
Sample input 2: 10 111 2
Sample output 2: 1001 1110

You are responsible for writing your own test cases. Please submit `test.txt`. This file should contain n lines (it is up to you to decide how many test cases are enough). Each line must have the following format (five integers separated by space):

I1 I2 B S P

That is, each line is a test case. The first three integers are I1, I2, and B. The next value is S (the correct sum). The last value is P (the correct product). An example `test.txt` is available on myUni.

3 Marking

Marking will be done automatically. The total mark is 5. I trust that you'd indeed implement the school method for addition and the Karatsuba algorithm for multiplication. I will randomly pick some 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] This makes your life easier as you do not have to deal with numbers like 'aff'.

- 3 marks for code correctness: We will collect all test cases submitted by all ADSA students (I expect thousands of test cases). Your code will be tested against all of them. If your code passes all test cases, then you receive 3 marks. If your code passes 90% test cases, then you receive 2 marks. If your code passes 80% test cases, then you receive 1 mark.
- 2 marks for test case quality: If your test cases contain mistakes, then your receive 0 marks. We will run your test cases against all other ADSA students' submissions. We keep track of how many submissions you test cases can break (by "breaking" a submission, I mean at least one of your test cases proves that the submission is buggy). Let x be the number of buggy submissions (our definition of buggy: a submission is not buggy if and only if passes all test cases). If your test cases can catch all buggy submissions, then you receive 2 marks. If your test cases can catch 80% buggy submissions, then you receive 1 mark.

4 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/2017/s2/adsa/assignment1/
```

aXXXXXXX should be your student ID. The directory path needs to be exactly "2017/s2/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/2017/s2/adsa/assignment1/ adsa-17-s2-assignment1/
cd adsa-17-s2-assignment1

svn add main.cpp

svn add test.txt
```

Commit the files to SVN:

```
svn commit -m "Adding ADSA assignment 1 main.cpp test.txt"
```

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

5 Websubmission

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. Your mark will be calculated offline after the deadline. You are welcome to resubmit for as many times as you wish (before the deadline).

We will compile your code using `g++ -o main.out -O2 -Wall main.cpp`. It is your responsibility to ensure that your code compiles **on the university system**.[2]

[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).