

Week 11

Remind

- 抄襲一律 0 分（包含被抄襲者）
- 繳交期限: **11/29(Sun.) 11:59 p.m.**
- 繳交的檔案格式、名稱請符合以下規定
 - 請繳交 zip檔至 Ceiba作業區，名稱為 <student_id>.zip
 - 解壓縮後須符合格式、名稱
 - e.g. b12345678.zip
- 必須完成 Demo 才可以提早離開
- 若沒有完成 Demo 就中途早退，視同缺席

Problem 1 - Large numbers calculator (7%)

Introduction

While doing integers arithmetic, we might run into a problem called overflow when the representation of the numbers is out of the range of the default int data type. However, there exists needs for manipulating large numbers. In this problem, you need to implement a large numbers calculator, which can support the following five arithmetic operations: **addition**, **subtraction**, **multiplication**, **division**, and **modulus**. For simplicity, all large numbers given are positive integers, and the result of operation should be an integer.

Note

In today's problem, every array you declared (if any) should be declared **dynamically**. Note that we will run a memory test to see whether the array you declare is done dynamically.

1-1. Addition (1%)

Specification

```
string add(string s1, string s2)
```

Description

Given two large numbers, calculate the sum of them.

Parameters

Two large numbers are encoded as strings.

Return type

The sum of the two numbers encoded as a string.

1-2. Subtraction (1%)

Specification

```
string subtract(string s1, string s2)
```

Description

Given two large numbers, calculate the difference (first - second) of them.

Parameters

Two large numbers are encoded as strings.

Return type

The difference of the two numbers encoded as a string.

1-3. Multiplication (2%)

Specification

```
string multiply(string s1, string s2)
```

Description

Given two large numbers, calculate multiplication of them. You should implement the long multiplication algorithm. You should NOT add the first number s1 for s2 times to get the result. For more information, please see the appendix A.

Parameters

Two large numbers are encoded as strings.

Return type

The product of the two numbers encoded as a string.

1-4. Division (1.5%)

Specification

```
string divide(string s1, string s2)
```

Description

Given two large numbers, calculate division of them (first number/second number). You should implement the long division algorithm. You should NOT successively subtract s2 from s1 naively to get the result. For more information, please see the appendix B.

Parameters

Two large numbers are encoded as strings.

Return type

The quotient of the two numbers encoded as a string.

1-5. Modulo (1.5%)

Specification

```
string modulo(string s1, string s2)
```

Description

Given two large numbers, calculate the modulo of them. Modulo operation calculates remainder of division of them (first number/second number).

Parameters

Two large numbers are encoded as strings.

Return type

The remainder of the two numbers encoded as a string.

Final remark

1. All inputs are strings, and all outputs should also be strings.
2. For subtraction, note that the result might be a negative number, when the first number is smaller than the second number.

File name

calc.cpp

Appendix A

				<i>a</i>					4	3	2				
				<i>b</i>				x		2	1	1			
				<i>c</i>						4	3	2			
				<i>d</i>						4	3	2			
				<i>e</i>					8	6	4				
				<i>f</i>					9	1	1	5	2		

Appendix B

$$\begin{array}{r}
 0 \\
 7 \overline{)452} \\
 \underline{0} \downarrow \\
 45
 \end{array}
 \quad
 \begin{array}{r}
 06 \\
 7 \overline{)452} \\
 \underline{0} \downarrow \\
 45 \\
 \underline{42} \downarrow \\
 32
 \end{array}
 \quad
 \begin{array}{r}
 064 \\
 7 \overline{)452} \\
 \underline{0} \downarrow \\
 45 \\
 \underline{42} \downarrow \\
 32 \\
 \underline{28} \downarrow \\
 4
 \end{array}
 \quad
 (64 \text{ r}4)$$

Step 1: "How many times?"

Step 2: "Multiply"

Step 3: "Subtract"

Step 4: "Drop it down"

(repeat steps for each number, left to right)