PA1

Computer Programming for Engineers

Instructor: Younghoon Kim

Introduction

- **Check PA1 at http://skku.goorm.io/.**

- **Deadline : 2020.10.14**
- **You can submit PA1 for two days (~ 2020.10.16) after the deadline. (25% deduction per day)**

Problem 1. Remain unique value in the array

<Description>

- There is a fixed size(10) array that contains integer values between 0 to 10 randomly.
- The value can be duplicated. That means there can be many same values.
- Remove duplicated value in the array but do not mix the order of values.

- Input: randomly generated 10 integer values
- Output: unique values of inputs

<Output Example>

```
$ ./unique
5 3 7 1 7 7 2 3 5 5
5 3 7 1 2
```

```
$ ./unique
0 7 3 7 1 8 1 0 3 10
0 7 3 1 8 10
```

Problem 2. Pascal's Triangle

<Description>

- Pascal's triangle is a triangular arrangement of binomial coefficients. In simple form, a Pascal triangle can be made in the following ways.
  - There are N numbers in row N and the first row is 1.
  - From the second row, the value at both ends of each row is 1, and the value of the remaining number is the sum of the two adjacent numbers in the row immediately above.
- Input : number of lines in integer value between 2 to 10
- Output : corresponding Pascal's Triangle
- NOTE: Recursive function only in Pascal function

<Output Example>

```
7
1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
1 5 10 10 5 1
1 6 15 20 15 6 1
```

Problem 3. Add operator

<Description>

- Implement five types of add functions with function overloading
- Each function has different data type for its first parameter and addition style is also different from one another base on that data types.
  - `Type i: int addition`
  - `Type d: double addition`
  - `Type s: string concatenation`
  - `Type I: int array addition (every element is added with element with the same index)`
  - `Type D: double array addition (every element is added with element with the same index)`
- Read the skeleton code given in the problem and implement add functions properly
- Note that the first argument of each function is for storing the return value
- Example of type i:

```
case 'i':
{
    int output;
    add(output); cout << "Output : " << output << endl;
    break;
}
```

Problem 3. Add operator

<Output Examples>

```
yhoon@yhoonLabServer:~$ ./a.out
Input type (i/d/s/I/D) : i
First input : 10
Second input : 11
Output : 21
```

```
yhoon@yhoonLabServer:~$ ./a.out
Input type (i/d/s/I/D) : s
First input : hello
Second input : world!
Output : hello world!
```

```
yhoon@yhoonLabServer:~$ ./a.out
Input type (i/d/s/I/D) : D
Array size : 3
First input : 1.1 2.2 3.3
Second input : 5.5 4.4 3.3
Output : 6.6 6.6 6.6
```

Problem 4. Arabinglish

<Description>

- Arabians write letters right to left except numbers.
- Arabinglish is a fake language that writes English in Arabian style.
- Implement a translator that inputs English and outputs Arabinglish.

- Input: An English sentence that less than 100 characters.
- Output: Arabinglish sentence.

<Output Example>

```
$ ./arab
There is 12 apples.
.selppa 12 si erehT
```

```
$ ./arab
I am 100 years old!
!dlo sraey 100 ma I
```