CPP Problem Design

Subject: Atoi class

Contributor: 張子樂，王聖文，林岳儒

Main testing concept: class

Basics

■ C++ BASICS
■ FLOW OF CONTROL
□ FUNCTION BASICS
□ PARAMETERS AND OVERLOADING
□ ARRAYS
□ STRUCTURES AND CLASSES
■ CONSTRUCTORS AND OTHER TOOLS
□ OPERATOR OVERLOADING, FRIENDS,AND REFERENCES
■ STRINGS
□ POINTERS AND DYNAMIC ARRAYS

Functions

□ SEPARATE COMPILATION AND NAMESPACES
□ STREAMS AND FILE I/O
□ RECURSION
□ INHERITANCE
□ POLYMORPHISM AND VIRTUAL FUNCTIONS
□ TEMPLATES
□ LINKED DATA STRUCTURES
□ EXCEPTION HANDLING
□ STANDARD TEMPLATE LIBRARY
□ PATTERNS AND UML

Description:

Write a class that convert a string into an integer. For example, given the string “1234” the function should return the integer 1234.If you do some research, you will find that there is a function named atoi and also the stringstream class that can do this conversion for you. However, in this Programming Project, you should write your own Atoi class to do the conversion.

```
class Atoi {

private:

     string beTrans;

     char sign;

public:

     Atoi();

     Atoi(string s);

     void SetString(string s);

     int Length();

     bool IsDigital();

     int StringToInteger();

};
```

Input:

Please use the code below as main function, inputs are numbers.

```
int main(void) {

    string beTrans;
    while (cin >> beTrans) {
        Atoi atoi(beTrans + "20");
        if (atoi.IsDigital()) {
            cout << atoi.Length() << endl;
            cout << atoi.StringToInteger() << endl;
            cout << sizeof(atoi.StringToInteger()) << endl;
        }
        cout<<” -----” <<endl;
        atoi.SetString(beTrans);
        if (atoi.IsDigital()) {
            cout << atoi.Length() << endl;
            cout << atoi.StringToInteger() << endl;
            cout << sizeof(atoi.StringToInteger()) << endl;
        }
        cout<<” -----” <<endl;
    }
    return 0;
}
```

Output:

Sample Input / Output：

Sample Input	Sample Output
05 11 23 -10 0 -11 8946 1891231	4 520 4 ----- 2 5 4 ----- 4 1120 4 ----- 2 11 4 ----- 4 2320 4 ----- 2 23

```
4
-----
4
-1020
4
-----
2
-10
4
-----
3
20
4
-----
1
0
4
-----
4
-1120
4
-----
2
-11
4
-----
6
894620
4
-----
4
8946
4
-----
9
189123120
4
-----
7
1891231
4
-----
```

■ Eazy,Only basic programming syntax and structure are required.
□ Medium,Multiple programming grammars and structures are required.
□ Hard,Need to use multiple program structures or complex data types.

Expected solving time:

10 minutes

Other notes: