CPP 程式設計題

命題者：TKG

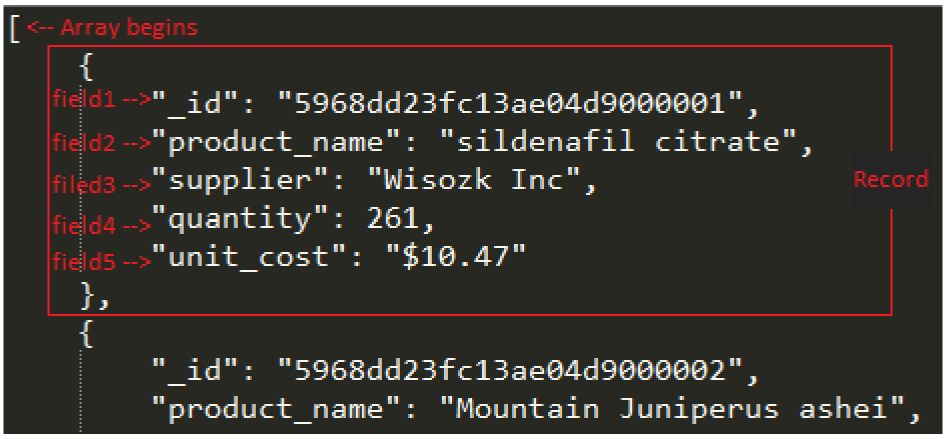
題目名稱(中文/英文)：Basic JSON

主要測試觀念： String Parsing, CRUD

Basics		Functions	
□	C++ BASICS 1	□	SEPARATE COMPILATION AND NAMESPACES
□	FLOW OF CONTROL	■	STREAMS AND FILE I/O
■	FUNCTION BASICS	□	RECURSION
□	PARAMETERS AND OVERLOADING	□	INHERITANCE
■	ARRAYS	□	POLYMORPHISM AND VIRTUAL FUNCTIONS
■	STRUCTURES AND CLASSES	□	TEMPLATES
■	CONSTRUCTORS AND OTHER TOOLS	□	LINKED DATA STRUCTURES
■	OPERATOR OVERLOADING, FRIENDS, AND REFERENCES	□	EXCEPTION HANDLING
■	STRINGS	□	STANDARD TEMPLATE LIBRARY
□	POINTERS AND DYNAMIC ARRAYS	□	PATTERNS AND UML

題目說明：

CRUD is an acronym for create, read, update, and delete. These are the four basic functions of persistent storage.
A pair (name: value) in a JSON file is composed of a name and a value. Each name is followed by a colon, and pairs are separated by a comma.
In the sample JSON file ***products.json*** containing records of merchandise in an array, each record is enclosed with curly brackets. In a record, each field is a name-value pair. A name must be a string while a value can be a string, number, boolean, etc. The figure below illustrates the structure of a sample JSON file.



```
[
    {
        "_id": "5968dd23fc13ae04d9000001",
        "product_name": "sildenafil citrate",
        "supplier": "Wisozk Inc",
        "quantity": 261,
        "unit_cost": "$10.47"
    },
    {
        "_id": "5968dd23fc13ae04d9000002",
        "product_name": "Mountain Juniperus ashei",
```

Given a JSON file containing the record of merchandise, please design and implement a class ***BasicJSON***, which has member functions to parse the file, manipulate (RUD) records in the given file, output manipulated records to a JSON file in the same format of input JSON file. Note that the class ***BasicJSON*** includes **at least** the following member functions:

1. **bool Parse(std::string *InputFileName*);** // Read a list of records from the file *InputFileName* and do string parsing to construct your data in an array. Also, if the file is parsed successfully, return true. Otherwise, return false.
2. **void Write(std::string *OutputFileName*);** // Output the modified data of an array to file *OutputFileName*.
3. **void Delete(std::string *Name*);** // Delete a pair in a record with passed *Name*.

4. **overload [] operator** such that
   1) JSONobject[*index*]: an *index* is a non-negative integer for accessing an array element (record).
   2) JSONobject[*index*][*name*] = newValue: update a value associated with given *name*.

Please note that:

i. We will provide ***main.cpp*** and ***products.json*** to test your class. Sample files ***input-main.cpp***, ***products.json*** and ***output.json*** are shown as an example of testing cases, located under the same directory, e.g.

```
..\
  ├ final##-JSON-配分##%
  …     ├ CPP 程式設計題-JSON.pdf
        ├ input-main.cpp
        ├ products.json
        └ output.json
```

ii. No comments are included in the JSON file.

iii. The comma at the end of any last pair in an object or array is **optional**

iv. A few redundant spaces and next-line characters are acceptable.

輸入說明：Substitution of your main function

輸出說明：Please refer to the sample output

IO **範例** ：

Sample Input	Sample Output
input-main.cpp products.json	output.json

附屬資料：

☑解答程式：BasicJSON.cpp, BasicJSON.h(檔名)

☑測試資料：input-main1.cpp, output1.json, input-main2.cpp, output2.json, input-main3.cpp, output3.json

☐ 易，僅需用到基礎程式設計語法與結構

■ 中，需用到多項程式設計語法與結構

☐ 難，需用到多項程式結構或較為複雜之資料型態或結構

解題時間：40 分鐘

其他註記：

input-main.cpp:

```
int main()
{
    BasicJSON json;
    // Parse the JSON file
    if (json.Parse("products.json"))
    {
        // Set the value of "quantity" in line 6 from 261 to 300
        json[0]["quantity"] = 300;
        // Set the value of "_id" in line 3 from "5968dd23fc13ae04d9000001" to
        // "000000000000000000000001"
        json[0]["_id"] = "000000000000000000000001";
        // Delete the pair with name "unit_cost" in line 21
        json[2].Delete("unit_cost");
        // Write out current data to JSON file
        json.Write("output.json");
    }
}
```

products.json:

```
[
    {
        "_id": "5968dd23fc13ae04d9000001",
        "product_name": "sildenafil citrate",
        "supplier": "Wisozk Inc",
        "quantity": 261,
        "unit_cost": "$10.47"
    },
    {
        "_id": "5968dd23fc13ae04d9000002",
        "product_name": "Mountain Juniperus ashei",
        "supplier": "Keebler-Hilpert",
        "quantity": 292,
        "unit_cost": "$8.74"
    },
    {
        "_id": "5968dd23fc13ae04d9000003",
        "product_name": "Dextromathorphan HBr",
        "supplier": "Schmitt-Weissnat",
        "quantity": 211,
        "unit_cost": "$20.53"
    }
]
```

output.json:

```
[
    {
        "_id": "000000000000000000000001",
        "product_name": "sildenafil citrate",
        "supplier": "Wisozk Inc",
        "quantity": 300,
        "unit_cost": "$10.47"
    },
    {
        "_id": "5968dd23fc13ae04d9000002",
        "product_name": "Mountain Juniperus ashei",
        "supplier": "Keebler-Hilpert",
        "quantity": 292,
        "unit_cost": "$8.74"
    },
    {
        "_id": "5968dd23fc13ae04d9000003",
        "product_name": "Dextromathorphan HBr",
        "supplier": "Schmitt-Weissnat",
        "quantity": 211
    }
]
```