**CPP Problem Design**

|  |
| --- |
| **Subject: Adding Large Numbers** |
| **Contributor: 謝宜杭,** **林承達, 廖宣瑋** |
| **Main testing concept: String Processing**   |  |  | | --- | --- | | **Basics** | **Functions** | | □ 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 | □ 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:**  Given two Integer A,B. Please calculate the adding result, noticed that the maximum bits of number can be approximate 10000.  ≒Don’t try using long long,long long int etc.(or\_m128), which will be absolutely invalid.Because of the problem requests the maximum bits of number will be approximate 10000((10^10000)-1).  ≒Please make sure the input number is valid.If not,please output”**Not a valid number, please try again.**”(But the A,B of test data must be read)  ≒Please package the big number as structure (Ex: struct BigInt…),give it simple Object type.  ≒The Adding of big number please use the Function Add, like below.  (Among them,const& won’t affect the grammar of parameter passing,but can avoiding unnecessary memory usage.Use it or not depends on you.)  Struct BigInt  {  …  }  BigInt Add(const BigInt &lhs,const BigInt &rhs)  {  //Calculation  Return …;  }  int main()  {  BigInt a,b;  …Input a,b  BigInt result = Add(a,b);  …Output result  }  **Input:**  The first line of the program will enter N(100 > N > 0)，indicates that there’re N pieces of test data.Every test data has two Integer A,B.With the maximum of A,B is（10^l0000) – 1.  **Output:**  輸出A+B。  **Sample Input / Output：**   |  |  | | --- | --- | | Sample Input | Sample Output | | 3  43789507384925798320000000000000000000000000001  44997439848794037580000000000000000000000000002  1bbbba45  1234567  1  9 | 8878694723371983590000000000000000000000000003  Not a valid number, please try again.  10 | |
| **□ 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 more complex data types.** |
| **Expected solving time:**  25 minutes |
| **Other notes:** |