**CPP Problem Design Example**

|  |
| --- |
| **Subject: Number Game** |
| **Contributor: 陳泳峰,袁瑋成,范茗翔** |
| **Main testing concept: Class Implementation**   |  |  | | --- | --- | | **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:**  A mathematician has designed a number game, but he doesn't know how to implement it on a computer, so he wants to ask for your help.  The rule of the game is giving an integer **A** and an integer list **S**(1<=S[i]<= 2^32 – 1), and you need to find the number in the list **S** that can be obtained by using the product of the digits decomposed by the integer **A**.  For example, given the integer **A** = 456 which can be decomposed into **A1** = 4, **A2** = 5 and **A3** = 6. Assuming that one of the numbers we read from list **S** is **E** = 30, since **E** = **A2** \* **A3**, we consider **E** to be legitimate. But if we read the number **E** = 12, we can see that **E** cannot be derived from **A1**, **A2** and **A3** by multiplication, so we consider **E** to be illegitimate. Your task is to find all the legitimate numbers in list **S**.  Please design a class called **NumberGame** to implement this game.  The class **NumberGame** should contains following member functions:   * + void SetInput(int): set the given integer **A**.   + void ProcessInput (): splitting the integer **A** into several digits.   + void SetFileName(string): set the file name of the file where list **S** is located.   + void LoadNumberList(): read list **S** from the file.   + void PrintAllValid(): print all the valid numbers in **S** ascendingly   + void Reset(): reset all variables.   Time limit: 2 second.  **Input:**  No inputs.  \*\* List **S** contains up to 50,000,000 numbers.  \*\* The main() function in your submission will be replaced when judging.  \*\* You can use the main() function in “Other Notes” to test your program.  **Output:**  The result of executing your program with the given main function.  **Sample Input / Output：**   |  |  | | --- | --- | | Sample Input | Sample Output | | No inputs. | 1  2  3  4  8  24  2  3  4  5  8  24  60 | |
| **■ Easy, 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:**  30 minutes |
| **Other notes:**  int main() {  NumberGame Game;  Game.SetInput(1234);  Game.ProcessInput();  Game.SetFileName("number.txt");  Game.LoadNumberList();  Game.PrintAllValid();  Game.Reset();  cout << "\n";  Game.SetInput(2345);  Game.ProcessInput();  Game.SetFileName("number.txt");  Game.LoadNumberList();  Game.PrintAllValid();  system("pause");  } |