**CPP Problem Design Example**

|  |
| --- |
| **Subject: Memory Simulator** |
| **Contributor: 林鼎傑, 宇恩, 鍾賢廣** |
| **Main testing concept: Pointer**   |  |  | | --- | --- | | **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:**  Please write a program to simulate the process of writing and reading memory data.   * For a given memory capacity N(Byte), the memory address is from 0 to (N-1). * There are four data types:  **char**(1 Byte), **short**(2 Bytes), **int**(4 Bytes), **String** (ending with '\0' or until the end of the memory if '\0' is not found). * There are two types of instructions. * Set <*position*> <*type*> <*value*>:  Write the given *value* with given *type* into memory start with the given *position*, if overflow occurs (current position is greater than memory capacity N), please output "Violation Access". If the size of *value* exceeds the size of *type*, the excess part will be ignored. * GET <*position*> <*type*>:  Prints the value in the given *type* from the memory starting with the given *position*.   \*\* Output "Violation Access." if the given position is illegal.  \*\* *position* and *value* are always in decimal.  \*\* **char** is treated as an integer here and does not consider the character situation.  Here is an example, on the left are the commands that will be entered, the first line is the memory size, the second line is the number of commands that will be entered, and the next n lines is the Set/Get commands. On the right is a graph of the memory change process for each line of commands, all bytes are represented in hexadecimal.  C:\Users\RB510\Desktop\123.jpg  **Input:**  The first line is an integer number N representing the size of the memory, the second line is an integer number n for the number of commands, and the next n lines is the Get/Set operations to the memory.  **Output:**  Output according to the commands.  **Sample Input / Output：**   |  |  | | --- | --- | | Sample Input | Sample Output | | 1000  7  Set 998 String Hiii  Set 1 int 3158064  Set 4 short 13878  Set 6 char 0  Get 998 String  Get 1000 int  Get 1 char | Violation Access.  Hi  Violation Access.  48 | |
| **□ 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:**  45 minutes |
| **Other notes:**  The input is certain to follow the command format.  The data type of the number is stored in little-endian. |