**CPP Problem Design**

|  |
| --- |
| **Subject: Print Standard Format** |
| **Contributor: 謝宜杭,** **林承達, 廖宣瑋** |
| **Main testing concept: Basic Output Format**   |  |  | | --- | --- | | **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:**  You are an IT employee in a company. Your manager requests you to print out N employee’s information, and you need to make sure the result remains clean. Each employee has three attributes: Name, Salary and Bonus.  **Input:**  The input contains several cases, each of them as described below.  The first line of input contains an integer **N** (10^6 >= N > 0), which represents the number of employees. Each of the next N lines indicates the information of each employee with three attributes: Name, Salary (between 10^9 ~ 0) and Award (between 10^9 ~ 0). Each attribute will be separated by one space.  \*The name attribute do not contain spaces.  The program should end after reading EOF.  **Output:**  For each set of input data (from the previous **N** to the next **N**), please make sure the width of each column is the same as the attribute with the longest width in each column. Use the symbol "|" to separate each column and align the content to the right. Note that for numeric attributes (Salary, Bonus), the content should be indented by two additional spaces.  **Sample Input / Output：**   |  |  | | --- | --- | | Sample Input | Sample Output | | 3  Alexandrescu 20000000 99999  Frank 100000 50  Andy 1 1  4  Andy 100 200  Anna 400 9999  Lipp 200 200  Stan 500 200 | Alexandrescu| 20000000| 99999  Frank| 100000| 50  Andy| 1| 1  Andy| 100| 200  Anna| 400| 9999  Lipp| 200| 200  Stan| 500| 200 | |
| **■ 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:**  10 minutes |
| **Other notes:** |