**CPP Problem Design Example**

|  |
| --- |
| **Subject: Student Records** |
| **Contributor: 溫勇威, 陳靖升, 鍾賢廣** |
| **Main testing concept: Structures**   |  |  | | --- | --- | | **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:**  Write a program that records at most 10 student data by structures. There are four functions for user to use:   1. insert (up to 10 records), (2) search, (3) delete, (4) print.   **A record of a student is defined as follow:**  typedef struct {  char firstName[25];  char lastName[30];  char phone[15];  } StRec;  **Input:**  There are four kinds of command formats (insert, delete search and print). Except the “print” command, the other three command lines contain **firstName**, **lastName** and **phone**. Use space to separate each data. The print command only needs to enter “print”.  **Format of four commands:**  (1) insert + firstName + lastName + phone  (E.g. insert Harry Potter 0987654321)  (2) delete + firstName + lastName + phone  (E.g. delete Harry Potter 0987654321)  (3) search + firstName + lastName + phone  (E.g. search Harry Potter 0987654321)  (4) print  (firstName <= 25 letters, lastName <= 30 letters, phone <= 15 numbers)  User can keep entering commands until reading EOF.  **Output:**  Users need to check for the input format accuracy, including:   1. If the length of **firstName** or **lastName** or **phone** is too long. 2. The string that inputted was not one of the four commands. 3. If **phone** is a number.   If any problems meet the above conditions, print “**Input Error**” and re-enter a command.  When **insert** is called, insert the record after last record. If there are already 10 records or the record already exists, print “**Insert Error**”.  When **delete** is called, find the record and delete it. If it does not exist, print “**Delete Error**”.  When **search** is called, find the record and print which index the record is in. If it does not exist, print “**Search Error**”.  When **print** is called, print all three data of records and separate them by space (e.g. Harry Potter 0987654321). If there are no records, print “**Print Error**”.  **Sample Input / Output：**   |  |  | | --- | --- | | Sample Input | Sample Output | | print  insert Elijah Smith 0912345585  print  insert Nicol Green 0901563245  insert Tom Taylor 0905615613  insert Paul Miller 0916548960  print  search Elijah Smith 0912345585  search Tom Taylor 0905615613  search Alen Lee 0953440450  delete Tom Taylor 0905615613  print  delete David King 0946549409  print  insert Obmar Wood 0965406546  print  insert Jone Smith 0916504894  insert Nicol Green 0901563245  print  insert ovuvuevuevueenyetuenwuevueugbemugbemosas Tom 0123456789012345  insert Rainy Jazz 0987a12345987654  insert Jone Six 0987580780  insert Tom Seven 0951348632  insert Jack Eight 886923654321  insert Sam Nine 00886958643215  insert TF Ten 0913648762  insert Howard Eleven 0913215468 | Print Error  Elijah Smith 0912345585  Elijah Smith 0912345585  Nicol Green 0901563245  Tom Taylor 0905615613  Paul Miller 0916548960  0  2  Search Error  Elijah Smith 0912345585  Nicol Green 0901563245  Paul Miller 0916548960  Delete Error  Elijah Smith 0912345585  Nicol Green 0901563245  Paul Miller 0916548960  Elijah Smith 0912345585  Nicol Green 0901563245  Paul Miller 0916548960  Obmar Wood 0965406546  Insert Error  Elijah Smith 0912345585  Nicol Green 0901563245  Paul Miller 0916548960  Obmar Wood 0965406546  Jone Smith 0916504894  Input Error  Input Error  Insert Error | |
| **□ 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 complex data types.** |
| **Expected solving time:**  30 minutes |
| **Other notes:**  delete Elijah Smith 0912345585  insert Tony Stark 0987654321  print  insert Chris Hemsworth 0912345678  insert Chris Evans 0956587145  delete Chris Evans 0956587145  search Chris Evans 0912345678  search zxcvbbnmasdfghjklqwertyuiop zxcvbbnmasdfghjklqwertyuiop 091234567890123456  search Chris Hemsworth 0912345678  print  delete Chris Hemsworth 0912345678  insert Ryan Reynolds 0937035846  search Tony Stark 0987654321  insert Ryan Reynolds 0937035846  search Chris Evans 0956587145  search Ryan Reynolds 0937035846  insert Thomas Stanley 0978587613  print  insert Harry Potter 0956134851  insert Ron Weasley 0913285698  insert Hermione Granger 0934568264  search Harry Potter 0956134851  insert Rubeus Hagrid 0916486352  insert Albus Dumbledore 0946325816  search Hermione Granger 0934568264  insert Severus Snape 0946213587  insert Hermione Granger 0934568264  insert Minerva MGonagall 0978463215  print  insert Lucius Malfoy 0981654873  insert Sirius Black 09615774832  delete asqwfrtihgindufstiasdfhcik Osas 09236548972  insert Moaning Myrtle 0973456892  insert Olympe Maxime 0911222333  insert Cedric Diggory 0955777999  delete Hermione Granger 0934568264  search Albus Dumbledore 0946325816  print |