|  |  |
| --- | --- |
| **Ch09-01.cpp:** *Bubble Sort* | |
| **Line#** | **Code** |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85 | #include <iostream>  #include <string.h>  using namespace std;  #define NOI(\_arr) (sizeof(\_arr)/sizeof(\_arr[0]))  struct STUDENT {  const char\* Name;  //char Name[20+1];  unsigned char Age;  float CGPA;  };  void showStudentInfo(STUDENT\* ps, int noi) {  for (int i = 0; i < noi; i++) {  cout << "Age:" << (int)ps->Age << " CGPA:" << ps->CGPA  << " Name:" << ps->Name << endl;  ps++;  }  cout << endl;  }  static void sortStudentByAge(STUDENT\* students, int noi) {  for (int x = 0; x < (noi - 1); x++) {  for (int y = 0; y < (noi - x - 1); y++) {  if (students[y].Age > students[y + 1].Age) {  STUDENT student = students[y];  students[y] = students[y + 1];  students[y + 1] = student;  }  }  }  }  static void sortStudentByCGPA(STUDENT\* students, int noi) {  for (int x = 0; x < (noi - 1); x++) {  for (int y = 0; y < (noi - x - 1); y++) {  if (students[y].CGPA > students[y + 1].CGPA) {  STUDENT student = students[y];  students[y] = students[y + 1];  students[y + 1] = student;  }  }  }  }  static void sortStudentByName(STUDENT\* students, int noi) {  for (int x = 0; x < (noi - 1); x++) {  for (int y = 0; y < (noi - x - 1); y++) {  if (strcmp(students[y].Name, students[y + 1].Name) > 0) {  STUDENT student = students[y];  students[y] = students[y + 1];  students[y + 1] = student;  }  }  }  }  int main() {  STUDENT students[] = {  {"Yong Tau Foo", 18, 3.15F},  {"Low Shi Fun", 21, 3.35F},  {"Lee Chee kang",20, 2.15F},  {"Low Mai Kai", 22, 3.05F},  {"Low Mee", 19, 2.95F},  };  showStudentInfo(students, NOI(students));  sortStudentByAge(students, NOI(students));  showStudentInfo(students, NOI(students));  sortStudentByCGPA(students, NOI(students));  showStudentInfo(students, NOI(students));  sortStudentByName(students, NOI(students));  showStudentInfo(students, NOI(students));  return 0;  }  /\*  Amend the above code to preform sorting of the student list by:  1) Age  2) CGPA  3) Name  \*/ |