

Trạng thái	Đã xong
Bắt đầu vào lúc	Thứ Bảy, 10 tháng 5 2025, 7:33 AM
Kết thúc lúc	Thứ Bảy, 10 tháng 5 2025, 7:35 AM
Thời gian thực hiện	2 phút 5 giây

Câu hỏi 1

Đúng

Đạt điểm 1,00

Mô tả tiếng Việt:

Cho struct `SCP` lưu trữ thông tin các vật thể dị thường được mô tả như sau:

```
struct SCP {
    int id;
    int objClass;
    string speConProcedures;
    string description;
    string* addendums;
    int numAddendums;
};
```

Chỉ số `objClass` của mỗi vật thể dị thường đánh giá mức độ khó khăn trong việc quản thúc đối tượng. Chỉ số này càng cao, đối tượng càng khó bị quản thúc, và việc quản thúc cần tiêu tốn nhiều tài nguyên hơn.

Hiện thực một hàm với prototype sau:

```
int compareObjectClass(const SCP &objA, const SCP &objB);
```

Hàm so sánh mức độ khó khăn trong việc quản thúc của hai đối tượng `objA` và `objB` và trả về `-1` nếu `objA` dễ quản thúc hơn `objB`, `1` nếu `objA` khó quản thúc hơn `objB`, `0` nếu mức độ khó khăn trong việc quản thúc của hai vật thể này tương đương nhau.

Ghi chú: (Các) thư viện `iostream` và `string` đã được khai báo, và namespace `std` đã được sử dụng.

English version:

Struct `SCP` used to store information about paranormal objects is declared as below:

```
struct SCP {
    int id;
    int objClass;
    string speConProcedures;
    string description;
    string* addendums;
    int numAddendums;
};
```

The `objClass` of each object indicates how difficult an object is to contain. The higher the value, the more resources and budget are needed to contain the object.

```
int compareObjectClass(const SCP &objA, const SCP &objB);
```

The function returns `-1` if `objA` is easier to be contained than `objB`, `1` if `objA` is more difficult to be contained than `objB`, or `0` if the difficulties in containing two objects are equivalent.

Note: Libraries `iostream` and `string` have been imported, and namespace `std` has been used.

For example:

Test	Result
SCP objA {2, 1, "SCP-002 is to remain connected to a suitable power supply at all times.", "SCP-002 resembles a tumorous, fleshy growth with a volume of roughly 60 cubed meter.", nullptr, 0}; SCP objB {55, 2, "Object is kept within a five (5) by five (5) by two point five (2.5) meter square room.", "All known information about the object is that it is not round.", nullptr, 0}; cout << compareObjectClass(objA, objB);	-1

Answer: (penalty regime: 0, 0, 0, 0, 0, 100 %)

Reset answer

```
1 int compareObjectClass(const SCP &objA, const SCP &objB) {
2     if (objA.objClass < objB.objClass)
3         return -1;
4     else if (objA.objClass > objB.objClass)
5         return 1;
6     else
7         return 0;
8 }
9
```

Passed all tests! ✓

Câu hỏi 2

Đúng

Đạt điểm 1,00

Mô tả tiếng Việt:

Cho struct **SCP** lưu trữ thông tin các vật thể dị thường được mô tả như sau:

```
struct SCP {
    int id;
    int objClass;
    string speConProcedures;
    string description;
    string* addendums;
    int numAddendums;
};
```

Tổ chức SCP lưu trữ thông tin các vật thể dị thường dưới dạng một array được cấp phát động. Tuy nhiên, vì một vài lý do, các hồ sơ không được sắp xếp theo một trật tự nào.

Hiện thực một hàm với prototype sau:

```
SCP** sortDatabase(SCP** arr, int n);
```

Trong đó **arr** là một array chứa các pointer trỏ đến các instance kiểu SCP, **n** là độ dài của mảng. Hàm thực hiện sắp xếp theo chiều tăng dần mã định danh các đối tượng SCP và trả về mảng sau khi sắp xếp.

Ghi chú: (Các thư viện **iostream** và **string** đã được khai báo, và namespace **std** đã được sử dụng.

English version:

Struct **SCP** used to store information about paranormal objects is declared as below:

```
struct SCP {
    int id;
    int objClass;
    string speConProcedures;
    string description;
    string* addendums;
    int numAddendums;
};
```

SCP Foundation's classified archives are stored as a dynamic allocated array. However, because of some specific reason, the array is not sorted in any order.

Implement the function with the following prototype:

```
SCP** sortDatabase(SCP** arr, int n);
```

Where **arr** is the array of **SCP** typed pointers, **n** is the length of the array. The function returns the sorted array in item number (**id**) increasing order.

Note: Libraries **iostream** and **string** have been imported, and **namespace std** has been used.

For example:

Test	Result
<pre>int n = 3; SCP** data = new SCP* [n]; data[0] = new SCP {2, 1, "SCP-002 is to remain connected to a suitable power supply at all times.", "SCP-002 resembles a tumorous, fleshy growth with a volume of roughly 60 cubed meter.", nullptr, 0}; data[1] = new SCP {55, 2, "Object is kept within a five (5) by five (5) by two point five (2.5) meter square room.", "All known information about the object is that it is not round.", nullptr, 0}; data[2] = new SCP {49, 1, "SCP-049 is contained within a Standard Secure Humanoid Containment Cell in Research Sector-02 at Site-19.", "SCP-049 is a humanoid entity, roughly 1.9 meters in height, which bears the appearance of a medieval plague doctor.", nullptr, 0}; data = sortDatabase(data, n); for (int i = 0; i < n; i++) cout << data[i]->id << " "; for (int i = 0; i < n; i++) { delete [] data[i]->addendums; delete data[i]; } delete [] data;</pre>	2 49 55

Answer: (penalty regime: 0 %)

Reset answer

<pre>1 SCP** sortDatabase(SCP** arr, int n) { 2 for (int i = 0; i < n - 1; i++) { 3 for (int j = i + 1; j < n; j++) { 4 if (arr[i]->id > arr[j]->id) { 5 // Hoán đổi con trỏ 6 SCP* temp = arr[i]; 7 arr[i] = arr[j]; 8 arr[j] = temp; 9 } 10 } 11 } 12 return arr; 13 } 14 }</pre>	
---	--

Passed all tests! ✓

