

Đã bắt đầu vào lúc	Thứ hai, 27 Tháng mười một 2023, 8:10 AM
Tình trạng	Đã hoàn thành
Hoàn thành vào lúc	Thứ ba, 28 Tháng mười một 2023, 4:15 PM
Thời gian thực hiện	1 ngày 8 giờ
Điểm	2,00/2,00
Điểm	10,00 của 10,00 (100%)

Câu hỏi**1**

Chính xác

Điểm 1,00 của
1,00**[Tiếng Việt]**Viết một hàm **đệ quy**`bool checkDuplicate(int* ar,int size)`

```
{  
    //Implement content in function here  
}
```

kiểm tra xem tất cả phần tử của một dãy số nguyên có duy nhất hay không (tức là dãy sẽ không có hai phần tử riêng biệt nào có cùng giá trị). Ví dụ dãy {1,2,3,4,5} là duy nhất, còn dãy {1,2,3,2,5} là không duy nhất vì lặp lại giá trị 2.

Đầu vào:

Hàm **checkDuplicate** sẽ nhận vào một dãy các số nguyên có giá trị trong khoảng $(-10^9; 10^9)$ và dãy đó có kích thước trong đoạn $[1; 10^4]$ cùng với kích thước của mảng.

Đầu ra:

Trả về **true** nếu dãy số nguyên không có 2 phần tử nào có cùng giá trị, ngược lại trả về **false**

***Lưu ý: sinh viên chỉ viết phần định nghĩa trong ngoặc của hàm checkDuplicate.**

[English]Write a **recursive** function`bool checkDuplicate(int* ar,int size)`

```
{  
    //Implement content in function here  
}
```

to check if all elements in an array are unique (meaning there are no identical value element). I.e {1,2,3,4,5} is unique, and {1,2,3,2,5} is not unique because it has two element with value 2.

Input:

Function **checkDuplicate** will receive an integer array with value between $(-10^9; 10^9)$ and the size in $[1; 10^4]$ and array size.

Output:

Return **true** if there is no identical element, else return **false**

***Note: Student only writes the definition of checkDuplicate in the bracket.**

For example:

Test	Input	Result
1	5 1 2 3 4 5	1
2	5 1 2 2 3 5	0
3	10 1 2 3 4 5 6 7 8 9 1	0
4	10 1 2 3 4 5 6 7 8 9 9	0
5	1 1	1
6	20 0	0
7	5 1 1 1 1 1	0
8	8 1 2 3 45 67 89 21 31	1
9	2 1 2	1
10	9 1 2 3 4 5 6 7 8 9	1

Answer: (penalty regime: 0 %)

[Reset answer](#)

```
1 ▼ if (size == 1) {  
2     return true;  
3 }  
4  
5 // Check if the last element of the array is present in the rest of the array  
6 ▼ for (int i = 0; i < size - 1; i++) {  
7 ▼     if (ar[i] == ar[size - 1]) {  
8         return false;  
9     }  
10 }  
11  
12 // Recursively check the rest of the array  
13 return checkDuplicate(ar, size - 1);
```



	Test	Input	Expected	Got	
✓	1	5 1 2 3 4 5	1	1	✓

Passed all tests! ✓

Chính xác

Điểm cho bài nộp này: 1,00/1,00.

Câu hỏi 2

Chính xác

Điểm 1,00 của 1,00

[Tiếng Việt]

Viết một hàm **recursiveSearch** nhận đầu vào là một dãy có N phần tử, và giá trị M. Hàm recursiveSearch sẽ trả về vị trí đầu tiên của giá trị M trong dãy (giả sử phần tử này có chỉ số là **k**) và xóa phần tử ở vị trí **k** ra khỏi dãy. Nếu không tìm thấy thì sẽ trả về -1

Đầu vào:

Hàm recursiveSearch nhận các giá trị đầu vào sau:

- Số phần tử của dãy N ($0 \leq N \leq 10^8$)
- Dãy N phần tử arr và mỗi phần tử có giá trị trong khoảng (0; 10^9)
- Số M ($0 < M < 10^9$)
- Giá trị khởi đầu của biến index từ hàm main là 0

Đầu ra:

Hàm recursiveSearch sẽ trả về theo yêu cầu của bài toán.

[English]

Write a **recursiveSearch** function that takes an array of N elements as input, and the value M. The recursiveSearch function returns the first position of the value M in the sequence (assuming this element has an index of k) and deletes the part. the element at position k out of the sequence. If not found, will return -1

Input:

The recursiveSearch function takes the following input values:

Number of elements of the sequence N ($0 \leq N \leq 108$)

The sequence of N elements arr and each element has a value in the range (0; 10^9)

Number M ($0 < M < 10^9$)

The initial call for index variable in main() is 0

Output:

The recursiveSearch function will return the request of the problem.

[Template code]

```
int recursiveSearch(int& n , int m, int arr[], int index)
{
    // Implement function content here
}
```

For example:

Test	Input	Result
1	5 1 1 2 3 4 5	0 2 3 4 5
2	5 6 1 2 3 4 5	-1 1 2 3 4 5
3	0 15	-1
4	5 100 100 100 2000 500 0	0 100 2000 500 0
5	5 15 1 2 3 4 15	4 1 2 3 4
6	6 5 5 5 5 5 5 5	0 5 5 5 5 5
7	0 0	-1
8	1 1 1	0
9	1 2 1	-1 1
10	20 123 23 45 6 4 8 1 9 12 7 12 22 33 76 90 12 123 21 56 98 0	15 23 45 6 4 8 1 9 12 7 12 22 33 76 90 12 21 56 98 0

Answer: (penalty regime: 0 %)

Reset answer

```

1  ▼ | if (index >= n) {
2      return -1;
3  }
4
5      // Check if arr[index] equals M
6  ▼ | if (arr[index] == m) {
7      // Store the position and delete the element from the array
8      int position = index;
9  ▼ | for (int i = index; i < n - 1; i++) {
10         arr[i] = arr[i + 1];
11     }
12     n--; // Decrement the size of the array
13     return position;
14 }
15
16 // Recursive call with index increased by 1
17 return recursiveSearch(n, m, arr, index + 1);

```

	Test	Input	Expected	Got	
✓	1	5 1 1 2 3 4 5	0 2 3 4 5	0 2 3 4 5	✓

Passed all tests! ✓

Chính xác

Điểm cho bài nộp này: 1,00/1,00.

BÁCH KHOA E-LEARNING



WEBSITE

HCMUT

MyBK

BKSI

LIÊN HỆ

📍 268 Lý Thường Kiệt, P.14, Q.10, TP.HCM

☎ (028) 38 651 670 - (028) 38 647 256 (Ext: 5258, 5234)

✉ elearning@hcmut.edu.vn

Copyright 2007-2022 BKEL - Phát triển dựa trên Moodle