

**BÁCH KHOA E-LEARNING**

[Trang của tôi](#) / [Khoá học](#) / [Học kỳ II năm học 2021-2022 \(Semester 2 - Academic year 2021-2022\)](#)

/ [Đại Học Chính Quy \(Bachelor program \(Full-time study\)\)](#)

/ [Khoa Khoa học và Kỹ thuật Máy tính \(Faculty of Computer Science and Engineering.\)](#) / [Khoa Học Máy Tính](#)

/ [Cấu trúc dữ liệu và giải thuật \(thực hành\) \(CO2004\)_L06 \(DH_HK212\)](#) / [Final Test \(Review\)](#) / [Final Test Review](#)

Câu hỏi 1

Không hoàn thành

Chấm điểm của 1,00

Cho node root của một danh sách liên kết đơn, hiện thực hàm sau đây:

```
void reduceDuplicate(Node* root);
```

để giảm các phần tử trùng lặp liên tiếp trong danh sách. Nếu có các phần tử liên tiếp giống nhau, ta chỉ giữ lại 1 phần tử.

Ví dụ, ta có 1 danh sách 122234452, sau khi thực hiện hàm reduceDuplicate ta thu được danh sách 123452. (số 2 cuối cùng giữ nguyên do nó không liên tiếp với dãy 222 phía trước)

Lưu ý: Các bạn có thể include thêm thư viện nếu cần thiết

Cấu trúc của một node được cho bên dưới. Các bạn chỉ cần viết nội dung hàm reduceDuplicate, các cấu trúc khác đã được import sẵn.

```
class Node
{
    int data;
    Node* next;
public:
    Node(): data(0), next(nullptr){}

    Node(int data, Node* next)
    {
        this->data = data;
        this->next = next;
    }

    int getData()
    {
        return this->data;
    }

    void setData(int data)
    {
        this->data = data;
    }

    Node* getNext()
    {
        return this->next;
    }

    void setNext(Node* next)
    {
        this->next = next;
    }
};
```

For example:

Test	Result
------	--------

Test	Result
Node* node1 = new Node(1, nullptr); Node* node2 = new Node(1, node1); Node* node3 = new Node(0, node2); printList(node3); reduceDuplicate(node3); printList(node3);	HEAD -> 0 -> 1 -> 1 -> NULL HEAD -> 0 -> 1 -> NULL

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

[Reset answer](#)

Ace editor not ready. Perhaps reload page?

Falling back to raw text area.

```
void reduceDuplicate(Node* root)
{
}
```

Kiểm tra

Câu hỏi **2**

Không hoàn thành

Chấm điểm của 1,00

Cho template của class PrinterQueue có 2 phương thức bắt buộc:

1. addNewRequest(int priority, string fileName)

Phương thức đầu tiên sẽ thêm 1 file vào danh sách hàng đợi của máy in (bao gồm độ ưu tiên và tên file). Test case sẽ có tối đa 100 file cùng lúc trong hàng đợi

2. print()

Phương thức thứ hai sẽ in tên file kèm xuống dòng và xóa nó ra khỏi hàng đợi. Nếu không có file nào trong hàng đợi, phương thức sẽ in ra "No file to print" kèm xuống dòng.

PrinterQueue tuân theo các quy tắc sau:

- fileName có độ ưu tiên cao nhất sẽ được in trước.
- Các fileName có cùng độ ưu tiên sẽ in theo thứ tự FIFO (First In First Out) order.

Nhiệm vụ của bạn là hiện thực class PrinterQueue thỏa mãn các yêu cầu dữ liệu trên

Lưu ý: Bạn có thể thay đổi mọi thứ, thêm thư viện cần thiết ngoại trừ thay đổi tên class, prototype của 2 public method bắt buộc.

Giải thích testcase 1: File goodbye.pdf có độ ưu tiên là 2 và được thêm vào sớm hơn file goodnight.pdf (độ ưu tiên = 2) nên sẽ được in trước, sau đó đến file goodnight.pdf và cuối cùng là hello.pdf có độ ưu tiên thấp nhất (1)

For example:

Test	Result
<pre>PrinterQueue* myPrinterQueue = new PrinterQueue(); myPrinterQueue->addNewRequest(1, "hello.pdf"); myPrinterQueue->addNewRequest(2, "goodbye.pdf"); myPrinterQueue->addNewRequest(2, "goodnight.pdf"); myPrinterQueue->print(); myPrinterQueue->print(); myPrinterQueue->print();</pre>	<pre>goodbye.pdf goodnight.pdf hello.pdf</pre>
<pre>PrinterQueue* myPrinterQueue = new PrinterQueue(); myPrinterQueue->addNewRequest(1, "hello.pdf"); myPrinterQueue->print(); myPrinterQueue->print(); myPrinterQueue->print();</pre>	<pre>hello.pdf No file to print No file to print</pre>

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

Reset answer

Ace editor not ready. Perhaps reload page?

Falling back to raw text area.

```
class PrinterQueue
{
    // your attributes
public:
    // your methods

    void addNewRequest(int priority, string fileName)
    {
        // your code here
    }

    void print()
    {
        // your code here
        // After some logic code, you have to print fileName with endl
    }
};
```

Kiểm tra

Thời gian còn lại 0:54:18

Câu hỏi **3**

Không hoàn thành

Chấm điểm của 1,00

Trong ngày hội CSE Job Fair 2020, Ban tổ chức đã treo cờ dọc theo đường dẫn vào tòa H6, có N lá cờ được đánh số từ 1 đến N , lá cờ thứ i có màu A_i . Tuy nhiên, sau khi treo cờ lên, người tổ chức sự kiện - bạn Quang Anh thấy dãy cờ có nhiều màu sắc khác nhau là không hợp lý. Phía Đoàn hội tiến hành rà soát và cho biết còn dư M lá cờ, đánh số từ 1 đến M , lá cờ thứ j có màu B_j . Ban tổ chức quyết định sẽ thay thế một số lá cờ trong số N lá cờ ban đầu để được dãy cờ mới có ít màu nhất có thể. Lá cờ bị thay xuống sẽ bị rách nên không thể dùng lại cho các lần thay tiếp theo, cũng như do thời gian có hạn, bất kì lá cờ nào thay lên cũng không được phép gỡ xuống.

Yêu cầu: Hãy tìm cách thay một số (hoặc giữ nguyên) lá cờ đã treo bằng một số lá cờ trong số cờ còn dư sao cho tổng số màu xuất hiện trên dãy cờ chính thức (N lá) là ít nhất.

Dữ liệu:

- Dòng đầu ghi hai số N và M lần lượt là số lá cờ chính thức và số lá cờ còn dư
- Dòng thứ hai ghi N số nguyên A_i cho biết màu của các lá cờ đã treo ($0 \leq A_i \leq 255$, $1 \leq i \leq N$)
- Dòng thứ ba ghi M số nguyên B_j cho biết màu của các lá cờ còn dư ($0 \leq B_j \leq 255$, $1 \leq j \leq M$)

Kết quả: in ra một số duy nhất, không có bất kì khoảng trắng nào, là số màu mới của dãy cờ chính thức sau khi được thay thế.

Giải thích testcase 1:

Dãy cờ mới sẽ là 1 2 5 5 2 5 5 5 5 -> Có 3 màu trong dãy cờ mới.

Các thư viện ĐÃ có trong bài, các bạn KHÔNG được phép thêm bất kì thư viện nào khác:

```
#include <iostream>
```

```
#include <fstream>
```

```
#include <string>
```

```
#include <cmath>
```

```
#include <vector>
```

```
#include <algorithm>
```

```
#include <stack>
```

```
#include <queue>
```

```
#include <map>
```

For example:

Test

```
int n=9;
int m=4;
vector<int> A({1,2,5,4,8,9,3,5,5});
vector<int> B({2,5,5,5});
cout << flag(n, m,A,B);
```

```
int n=45;
int m=1;
vector<int>
A({32,134,131,44,194,254,63,209,140,181,29,108,94,153,165,117,159,2,33,31,133,229,255,47,144,74,120,15,88,0,111,30,137,143,156,75,
vector<int> B({151});
cout << flag(n, m,A,B);
```

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

Reset answer

Ace editor not ready. Perhaps reload page?

Falling back to raw text area.

```
//Helping functions goes here
int flag(int n, int m,vector<int> A, vector<int> B){
    //TODO
}
```

Kiểm tra

Câu hỏi 4

Không hoàn thành

Chấm điểm của 1,00

2

Có một trò chơi thẻ bài với n người chơi. Trong trò chơi này, có 60 loại thẻ bài (số lượng thẻ bài của mỗi loại không có giới hạn) được đánh số từ 0 đến 59. Mỗi người chơi được biểu diễn bởi một số nguyên a_i (a_i đại diện cho người chơi thứ i). Nếu người chơi thứ i có thẻ bài loại j trong bộ bài, thì bit thứ j (2^j) sẽ được bật (giá trị 1) trong biểu diễn nhị phân của a_i . Ngược lại, bit này sẽ không được bật (giá trị 0). Ví dụ, người chơi thứ 2 được biểu diễn bởi a_2 . a_2 có giá trị là 5, trong biểu diễn nhị phân là 101, tức là người chơi thứ 2 sẽ có thẻ bài loại 0 và 2 trong bộ bài của mình.

Một bộ bài x được xem như mạnh hơn bộ bài y nếu trong x có ít nhất một loại thẻ bài mà bộ bài y lại không có loại thẻ này. Hai bộ bài có thể mạnh hơn lẫn nhau (Ví dụ như bộ bài x chỉ có thẻ bài loại 0 và bộ bài y chỉ có thẻ bài loại 1).

Rin muốn tổ chức một giải đấu giữa các người chơi. Để đảm bảo công bằng, Rin muốn chọn ra một nhóm người chơi từ n người chơi ban đầu sao cho trong nhóm người này, không ai có bộ bài mạnh hơn tất cả bộ bài của các người chơi còn lại trong nhóm. Bạn hãy giúp Rin tìm ra số lượng người chơi tối đa có thể tham gia vào giải đấu này. Lưu ý nếu $a_i = 0$, tức là người chơi thứ i không có thẻ bài nào, đồng nghĩa với việc anh ấy không đủ tư cách để tham dự giải đấu.

Lưu ý: nhóm người tham gia giải đấu phải có ít nhất hai người

Các giới hạn dữ liệu:

- $1 \leq n \leq 3000$.
- $0 \leq a_i < 2^{60}$ ($0 \leq i < n$).
- Thời gian mỗi testcase: 1 giây.

Giải thích testcase 1:

Trong testcase này có 3 người chơi:

- Người đầu tiên chỉ có thẻ bài loại 1.
- Người thứ hai chỉ có thẻ bài loại 1.
- Người thứ ba có thẻ bài loại 0 và và 1. Do đó bộ bài người chơi này mạnh hơn tất cả bộ bài của những người chơi còn lại.

Vì vậy, chỉ có thể tổ chức giải đấu giữa hai người chơi đầu tiên và người chơi thứ hai.

For example:

Test	Result
<pre>long long a[] = { 2, 2, 3 }; int n = sizeof(a) / sizeof(long long); cout << maxNumOfPlayers(n, a);</pre>	2
<pre>long long a[] = { 1, 2, 7, 16 }; int n = sizeof(a) / sizeof(long long); cout << maxNumOfPlayers(n, a);</pre>	0

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

Reset answer

Ace editor not ready. Perhaps reload page?

Falling back to raw text area.

```
// Here are some given libraries, you can add more libraries if needed.
#include<stdio.h>
#include<iostream>

using namespace std;

// You can also add more helping functions if needed.

int maxNumOfPlayers(int n, long long a[]) {
    // TODO: return the maximum number of players who can participant in the tournament. In case cannot
    set up a tournament that meets all the conditions, return 0.
}
```

Kiểm tra

◀ Graph 2

Chuyển tới...

Copyright 2007-2021 Trường Đại Học Bách Khoa - ĐHQG Tp.HCM. All Rights Reserved.

Địa chỉ: Nhà A1- 268 Lý Thường Kiệt, Phường 14, Quận 10, Tp.HCM.

Email: elearning@hcmut.edu.vn

Phát triển dựa trên hệ thống Moodle