

Đã bắt đầu vào lúc	Thứ sáu, 24 Tháng mười một 2023, 12:50 PM
Tình trạng	Đã hoàn thành
Hoàn thành vào lúc	Thứ sáu, 24 Tháng mười một 2023, 1:12 PM
Thời gian thực hiện	21 phút 39 giây
Điểm	3,00/3,00
Điểm	10,00 của 10,00 (100%)

Câu hỏi 1

Chính xác

Điểm 1,00 của 1,00

[Tiếng Việt]

Một hàm tìm ước số chung lớn nhất của 2 số nguyên dương có thể viết thông qua đệ quy và vòng lặp đơn giản. Bạn hãy viết hàm **gcdRecursion** để hiện thực tìm ước chung lớn nhất bằng đệ quy và hàm **gcdIteration** để tìm ước số chung lớn nhất bằng vòng lặp

Đầu vào:

Lần lượt 2 số nguyên p, q ($1 \leq p, q < 10^9$).

Đầu ra:

Hàm **gcdRecursion** và **gcdIteration** lần lượt trả về giá trị là ước chung lớn nhất của p, q

[English]

A function that finds the greatest common divisor of two positive integers can be written through simple recursion and looping. You write the function **gcdRecursion** to perform the greatest common divisor by recursion and the function **gcdIteration** to find the greatest common divisor by loop.

Input:

Two integers p, q respectively ($1 \leq p, q < 10^9$).

Output:

The **gcdRecursion** and **gcdIteration** functions return the greatest common divisor of p, q , respectively.

Template of full code:

```
#include<iostream>
#include<string>
#include <string>
#include <sstream>
#include <fstream>
#include <vector>
using namespace std;

/* END of library */

int gcdRecursion(int p, int q)
{
    // BEGIN YOUR IMPLEMENTATION [1]
    // TODO

    // END YOUR IMPLEMENTATION [1]
}

int gcdIteration(int p, int q)
{
    hiddenCheck();
    // BEGIN YOUR IMPLEMENTATION [2]
    // TODO

    // END YOUR IMPLEMENTATION [2]
    return 0;
}

int main()
{
    hiddenCheck();
    int p,q;
    cin>>p>>q;
    cout<<gcdRecursion(p,q)<< " "<<gcdIteration(p,q);
    return 0;
}
```

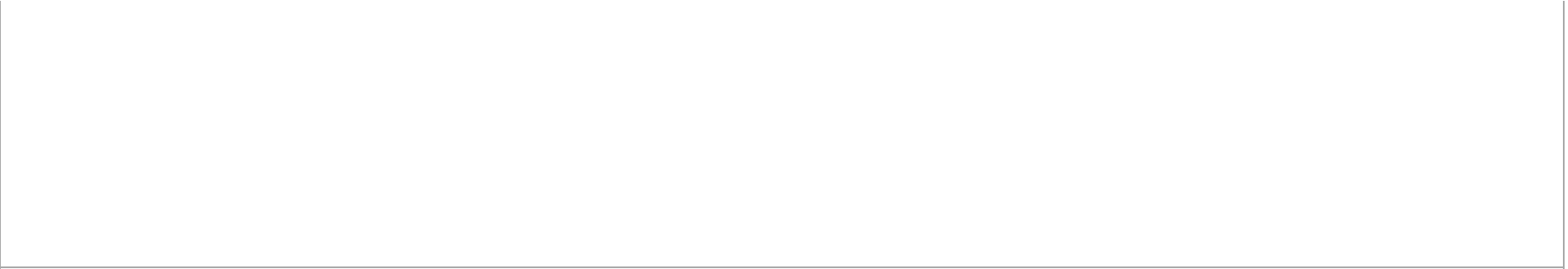
For example:

Test	Input	Result
1	5 5	5 5

Answer: (penalty regime: 0 %)

Reset answer

```
1 int gcdRecursion(int p, int q)
2 {
3     if (q == 0) {
4         return p;
5     } else {
6         return gcdRecursion(q, p % q);
7     }
8     // BEGIN YOUR IMPLEMENTATION [1]
9     // TODO
10    // END YOUR IMPLEMENTATION [1]
11 }
12 int gcdIteration(int p, int q)
13 {
14     while (q != 0) {
15         int temp = q;
16         q = p % q;
17         p = temp;
18     }
19     return p;
20    // BEGIN YOUR IMPLEMENTATION [2]
21    // TODO
22    // END YOUR IMPLEMENTATION [2]
23 }
```



	Test	Input	Expected	Got	
✓	1	5 5	5 5	5 5	✓

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]

Cho một chuỗi, hiện thực hàm

```
int strLen(char* str){}
```

để tính độ dài của chuỗi sử dụng đệ quy.

Lưu ý không được sử dụng các từ khóa như for, while, goto (thậm chí là tên biến, comment).

Trong bài tập này đã khai báo #include <iostream> và using namespace std;

[English]

Given a string, implement function

```
int strLen(char* str){}
```

to calculate length of the string using recursion.

Please note that you can't using key work for, while, goto (even in variable names, comment).

For this exercise, we have #include <iostream> and using namespace std;

For example:

Test	Result
<pre>char str[] = "Truong DH Bach Khoa"; cout << strLen(str);</pre>	19

Answer: (penalty regime: 0 %)

[Reset answer](#)

```
1 ▼ int strlen(char* str) {  
2 ▼     if (*str == '\\0') {  
3         return 0;  
4 ▼     } else {  
5         return 1 + strlen(str + 1);  
6     }  
7 }
```

	Test	Expected	Got	
✓	<pre>char str[] = "Truong DH Bach Khoa"; cout << strlen(str);</pre>	19	19	✓

Passed all tests! ✓

Chính xác

Điểm cho bài nộp này: 1,00/1,00.

Câu hỏi 3

Chính xác

Điểm 1,00 của 1,00

[Tiếng Việt]

Cho một số dương, in theo một mẫu mà không sử dụng bất kỳ vòng lặp nào.

Đầu vào: $n = 16$

Đầu ra: 16, 11, 6, 1, -4, 1, 6, 11, 16 (không có khoảng trắng ở cuối)

Đầu vào: $n = 10$

Đầu ra: 10, 5, 0, 5, 10 (không có khoảng trắng ở cuối)

Về cơ bản, đầu tiên chúng ta bớt 5 cái một cho đến khi chúng ta đạt đến số âm hoặc 0. Sau khi chúng tôi đạt đến 0 hoặc số âm, chúng tôi thêm 5 cho đến khi chúng tôi đạt đến n .

Lưu ý: Xin lưu ý rằng bạn không thể sử dụng key work for, while, goto (ngay cả trong tên biến, comment).

Bạn có thể triển khai các hàm đệ quy khác nếu cần.

Đối với bài tập này, chúng ta có `#include <iostream>` và sử dụng namespace std;

[English]

Given a positive number, print following a pattern without using any loop.

Input: $n = 16$

Output: 16, 11, 6, 1, -4, 1, 6, 11, 16 (has no space at the end)

Input: $n = 10$

Output: 10, 5, 0, 5, 10 (has no space at the end)

We basically first reduce 5 one by one until we reach a negative or 0. After we reach 0 or negative, we one add 5 until we reach n .

Note: Please note that you can't using key work for, while, goto (even in variable names, comment).

You can implement other recursive functions if needed.

For this exercise, we have `#include <iostream>` and `using namespace std;`

For example:

Test	Result
<code>printPattern(16);</code>	16 11 6 1 -4 1 6 11 16
<code>printPattern(10);</code>	10 5 0 5 10

Answer: (penalty regime: 0 %)

Reset answer

```
1 void printPattern(int n) {  
2     if (n <= 0) {  
3         cout << n;  
4         return;  
5     }  
6     cout << n << " ";  
7     printPattern(n - 5);  
8     cout << " " << n;  
9 }
```

	Test	Expected	Got	
✓	printPattern(16);	16 11 6 1 -4 1 6 11 16	16 11 6 1 -4 1 6 11 16	✓
✓	printPattern(10);	10 5 0 5 10	10 5 0 5 10	✓

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