

Trạng thái	Đã xong
Bắt đầu vào lúc	Thứ Tư, 16 tháng 4 2025, 4:45 PM
Kết thúc lúc	Thứ Tư, 16 tháng 4 2025, 4:55 PM
Thời gian thực hiện	9 phút 39 giây

Câu hỏi 1

Đúng

[Tiếng Việt]

Cho số nguyên n và số nguyên dương e , trong đó n là số cần tính lũy thừa và e là số mũ. Hãy viết một hàm đệ quy

```
int calculate_power(int n, int e){}
```

để tính giá trị của n^e .

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 integer n and positive integer e , where n is the base and e is the exponent. Write a recursive function

```
int calculate_power(int n, int e){}
```

to calculate the value of n^e .

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>int n = 2; int e = 3; cout << calculate_power(n ,e);</pre>	8

Answer: (penalty regime: 0 %)

[Reset answer](#)

```
1  /**
2   * Hàm tính lũy thừa n^e bằng phương pháp đệ quy
3   * @param n: Số nguyên cơ số
4   * @param e: Số nguyên dương mũ
5   * @return Giá trị n^e
6   */
7  int calculate_power(int n, int e) {
8      // Trường hợp cơ sở: e = 0, kết quả là 1
9      if (e == 0) {
10         return 1;
11     }
12
13     // Trường hợp e = 1, kết quả là n
14     if (e == 1) {
15         return n;
16     }
17
18     // Trường hợp e là số chẵn: n^e = (n^(e/2))^2
19     if (e % 2 == 0) {
20         int half_power = calculate_power(n, e / 2);
21         return half_power * half_power;
22     }
23     // Trường hợp e là số lẻ: n^e = n * n^(e-1)
24     return n * calculate_power(n, e - 1);
25 }
```

```
25     else {  
26         return n * calculate_power(n, e - 1);  
27     }
```

	Test	Expected	Got	
✓	int n = 2; int e = 3; cout << calculate_power(n ,e);	8	8	✓

Passed all tests! ✓

Câu hỏi 2

Đúng

[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)
{
    // 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  /**
2   * Tìm ước chung lớn nhất (GCD) của hai số bằng phương pháp đệ quy
3   * Sử dụng thuật toán Euclid: gcd(p, q) = gcd(q, p % q)
4   * @param p Số nguyên dương thứ nhất
5   * @param q Số nguyên dương thứ hai
6   * @return Ước chung lớn nhất của p và q
7   */
8  int gcdRecursion(int p, int q)
9  {
10     // BEGIN YOUR IMPLEMENTATION [1]
11     // Trường hợp cơ sở: nếu q = 0, ước chung lớn nhất là p
12     if (q == 0) {
13         return p;
14     }
15
16     // Gọi đệ quy với tham số q và p % q
17     return gcdRecursion(q, p % q);
18     // END YOUR IMPLEMENTATION [1]
19 }
20
21 /**
22 * Tìm ước chung lớn nhất (GCD) của hai số bằng phương pháp vòng lặp
23 * Sử dụng thuật toán Euclid: gcd(p, q) = gcd(q, p % q)
24 * @param p Số nguyên dương thứ nhất
25 * @param q Số nguyên dương thứ hai
26 * @return Ước chung lớn nhất của p và q
27 */
28 int gcdIteration(int p, int q)
29 {
30     // BEGIN YOUR IMPLEMENTATION [2]
31     // Tránh trường hợp p = 0 hoặc q = 0
32     if (p == 0) return q;
33     if (q == 0) return p;
34
35     // Vòng lặp thực hiện thuật toán Euclid
36     while (q != 0) {
37         // Lưu giá trị hiện tại của q
38         int temp = q;
39         // Cập nhật q = p % q
40         q = p % q;
41         // Cập nhật p = temp
42         p = temp;
43     }
44
45     // Kết quả là giá trị p sau khi kết thúc vòng lặp
46     return p;
47     // END YOUR IMPLEMENTATION [2]
48 }

```

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

Passed all tests! ✓

Câu hỏi 3

Đúng

[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
char str[] = "Truong DH Bach Khoa"; cout << strLen(str);	19

Answer: (penalty regime: 0 %)

Reset answer

```
1  /**
2   * Hàm tính độ dài của chuỗi sử dụng đệ quy
3   * @param str Con trỏ đến chuỗi cần tính độ dài
4   * @return Độ dài của chuỗi
5   */
6  int strLen(char* str)
7  {
8      // Trường hợp cơ sở: nếu gặp ký tự kết thúc chuỗi '\0'
9      if (*str == '\0') {
10         return 0;
11     }
12
13     // Trường hợp đệ quy: tính độ dài của phần chuỗi còn lại và cộng thêm 1
14     return 1 + strLen(str + 1);
15 }
```

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

Passed all tests! ✓

