

Trạng thái	Đã xong
Bắt đầu vào lúc	Thứ Tư, 7 tháng 5 2025, 1:05 PM
Kết thúc lúc	Thứ Tư, 7 tháng 5 2025, 1:17 PM
Thời gian thực hiện	11 phút 47 giây

Câu hỏi 1

Đúng

Đạt điểm 1,00

[Tiếng Việt]

Viết hàm tìm chữ cái xuất hiện nhiều lần nhất trong một chuỗi và số lần xuất hiện của chữ cái đó. Nếu có nhiều chữ cái thỏa mãn, xác định chữ cái có thứ tự alphabet nhỏ nhất.

Đầu vào:

- const char* str: chuỗi ký tự đang xét, chỉ bao gồm chữ cái thường và hoa

Đầu ra:

- char& res: chữ cái xuất hiện nhiều lần nhất trong chuỗi str
- int& freq: số lần xuất hiện của chữ cái đó

Chú ý: tham số res là chữ cái thường. Chữ cái viết hoa trong str cũng được tính là giống chữ cái viết thường (VD "aA" chứa hai ký tự 'a')

Lưu ý: Sinh viên chỉ có 5 lần nộp không tính penalty, ở lần nộp thứ 6 trở đi bài làm sẽ được tính là 0 điểm.

[English]

Write a function that finds the most frequent character in a given string and how many times that character appears. If there are multiple characters satisfy the condition, find the one with the lowest alphabetical order.

Input:

- const char* str: the input string, contains only lowercase and uppercase ASCII letters

Output:

- char& res: the most frequent character in str
- int& freq: the times that the result character appears in the string

Note: return res as a lowercase letter. Uppercase letters in str is also counted as lowercase letters (e.g the string "aA" contains two letter 'a')

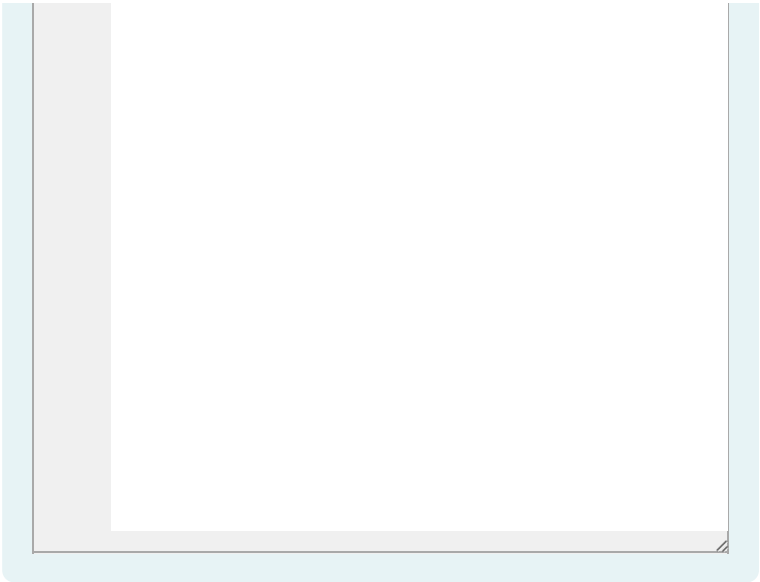
For example:

Test	Input	Result
<pre>int n; cin >> n; char* str = new char[n+1]; for(int i = 0; i < n; i++) cin >> str[i]; str[n] = 0; char res = 0; int freq = 0; mostFrequentCharacter(str, res, freq); cout << res << ' ' << freq; delete[] str;</pre>	<pre>7 aCxAboc</pre>	<pre>a 2</pre>
<pre>int n; cin >> n; char* str = new char[n+1]; for(int i = 0; i < n; i++) cin >> str[i]; str[n] = 0; char res = 0; int freq = 0; mostFrequentCharacter(str, res, freq); cout << res << ' ' << freq; delete[] str;</pre>	<pre>6 OmnPqB</pre>	<pre>b 1</pre>

Answer: (penalty regime: 0, 0, 0, 0, 0, 0, 0, 0, 50, 100 %)

Reset answer

```
1 void mostFrequentCharacter(char* str, char& res,
2 // Write your code
3 int count[26] = {0};
4
5 for(int i = 0; str[i] != '\0'; ++i){
6     char c = str[i];
7     if(c >= 'A' && c <= 'Z'){
8         c = c - 'A' + 'a';
9     }
10
11     if(c >= 'a' && c <= 'z'){
12         count[c - 'a']++;
13     }
14 }
15 freq = 0;
16 res = 'a';
17 for(int i = 0; i < 26; ++i){
18     if(count[i] > freq || (count[i] == freq && i < res - 'a')){
19         {
20             freq = count[i];
21             res = i + 'a';
22         }
23     }
24 }
```



	Test	Input	Expected	Got
✓	<pre>int n; cin >> n; char* str = new char[n+1]; for(int i = 0; i < n; i++) cin >> str[i]; str[n] = 0; char res = 0; int freq = 0; mostFrequentCharacter(str, res, freq); cout << res << ' ' << freq; delete[] str;</pre>	7 aCxAboc	a 2	a 2
✓	<pre>int n; cin >> n; char* str = new char[n+1]; for(int i = 0; i < n; i++) cin >> str[i]; str[n] = 0; char res = 0; int freq = 0; mostFrequentCharacter(str, res, freq); cout << res << ' ' << freq; delete[] str;</pre>	6 OmnPqB	b 1	b 1

Passed all tests! ✓

Câu hỏi 2

Đúng

Đạt điểm 1,00

[Tiếng Việt]

Cho hai số tự nhiên n và m . Hãy viết một hàm tính toán và trả về giá trị của n trong hệ m phân, biết giá trị m thuộc $\{2, 4, 8\}$. Sinh viên cần thay đổi tham số m để có giá trị mặc định là 2 trong định nghĩa hàm.

Đầu vào:

int n : số tự nhiên n cần được chuyển đổi sang hệ m phân

int m : hệ cơ số cần chuyển đổi. m phải có đối số mặc định là 2 khi định nghĩa hàm.

Đầu ra:

int: giá trị của n trong hệ m phân

Chú ý: đầu vào sẽ đảm bảo giá trị trả về không bị tràn số

[English]

Given two natural numbers n and m. Write a function that converts n to base m and returns that value, given that m is in {2, 4, 8}. Student must provide a default argument of value 2 for the second parameter(m).

Input:

int n: a natural number n that needs to be converted

int m: the new base which n is converted to. The default argument of this parameter must be 2 when defining the function.

Output:

int: the value of n in base m

Note: the input guarantees that the return value will not cause integer overflow

For example:

Test	Input	Result
int n; cin >> n; cout << convertToBaseM(n, 4) << '\n'; cout << convertToBaseM(n, 8) << '\n'; cout << convertToBaseM(n);	16	100 20 10000
int n; cin >> n; cout << convertToBaseM(n, 4) << '\n'; cout << convertToBaseM(n, 8) << '\n'; cout << convertToBaseM(n);	28	130 34 11100

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

Reset answer

```
1 int convertToBaseM(int n, int m = 2) {  
2     // Write your code  
3     int res = 0;  
4     int place = 1;  
5  
6     while(n > 0){  
7         res += (n % m) * place;  
8         n /= m;  
9         place *= 10;  
10    }  
11    return res;  
12 }
```



	Test	Input	Expected	Got	
✓	<pre>int n; cin >> n; cout << convertToBaseM(n, 4) << '\n'; cout << convertToBaseM(n, 8) << '\n'; cout << convertToBaseM(n);</pre>	16	100 20 10000	100 20 10000	✓
✓	<pre>int n; cin >> n; cout << convertToBaseM(n, 4) << '\n'; cout << convertToBaseM(n, 8) << '\n'; cout << convertToBaseM(n);</pre>	28	130 34 11100	130 34 11100	✓

Passed all tests! ✓