

<b>Trạng thái</b>	Đã xong
<b>Bắt đầu vào lúc</b>	Thứ Sáu, 3 tháng 5 2024, 3:08 PM
<b>Kết thúc lúc</b>	Chủ Nhật, 5 tháng 5 2024, 12:13 AM
<b>Thời gian thực hiện</b>	1 ngày 9 giờ



## Câu hỏi 1

Đúng

Đạt điểm 1,00

Viết hàm **void uppercase(string output)** để đọc vào chuỗi S từ bàn phím, sau đó chuyển tất các ký tự trong chuỗi S thành ký tự viết HOA và xuất kết quả ra file output.

Chú ý: chỉ thay đổi các chữ cái in thường, các kí tự khác sẽ được giữ nguyên.

**Đầu vào:**

Biến "output" chứa tên file dùng để xuất kết quả.

**Đầu ra:**

Hàm đọc chuỗi S từ bàn phím và xử lý chuỗi như mô tả. Sau đó ghi chuỗi đã xử lý vào file có tên được chứa trong biến "output" (hàm không trả về kết quả).

Write a function **void uppercase(string output)** to read the string S from the keyboard, then convert all characters in string S to uppercase characters and output the result to the output file.

Note: change only lowercase letters, other characters will not change.

**Input:**

The variable "output" contains the filename used to output the result.

**Output:**

The function reads the string S from the keyboard and processes the string as described. Then write the processed string to a file whose name is contained in the variable "output" (the function does not return results).

**For example:**

Test	Input	Result
1	heLl010	HELL010

**Answer:** (penalty regime: 0 %)



[Reset answer](#)

```
1
2 void uppercase(string output) {
3     // TODO
4     string input;
5     getline(cin, input);
6
7     for (char& c : input) {
8         if (c >= 'a' && c <= 'z') {
9             c = c - ('a' - 'A');
10        }
11    }
12    ofstream outFile(output);
13    if(outFile.is_open()){
14        outFile << input;
15    }
16    outFile.close();
17 }
```



	Test	Input	Expected	Got	
✓	1	heLl010	HELL010	HELL010	✓

Passed all tests! ✓



## Câu hỏi 2

Đúng

Đạt điểm 1,00

Viết hàm **void process(string fileName)** đọc một file đuôi txt, dòng đầu gồm 2 số N và M cách nhau bởi 1 khoảng trắng.

N dòng theo sau, mỗi dòng gồm M số thực cách nhau bởi 1 khoảng trắng.

Trả về giá trị lớn nhất của mỗi dòng và giá trị lớn nhất trong tất cả các số.

Các thư viện đã được include: *iostream*, *fstream*, *string*.

**Đầu vào:**

Biến "fileName" là tên file chứa dữ liệu đầu vào.

**Đầu ra:**

Giá trị lớn nhất của mỗi dòng và giá trị lớn nhất của tất cả các số, cách nhau bởi ký tự khoảng trắng.

Write a function **void process(string fileName)** that reads a txt file, the first line consists of 2 numbers N and M separated by a space.

N lines follow, each containing M real numbers separated by 1 space.

Print the maximum value of each line and the maximum value of all numbers.

Included libraries: *iostream*, *fstream*, *string*.

**Input:**

The variable "fileName" is the name of the file containing the input data.

**Output:**

The maximum value of each line and the maximum value of all numbers, separated by a space character.

**For example:**

Test	Input	Result
1	3 4 1 2 3 4 4.3 41.2 -4.576 0 -3 -2 -4 -1	4 41.2 -1 41.2



**Answer:** (penalty regime: 0 %)**Reset answer**

```
1 double maxrow( double row[], int m)
2 {   double max = row[0];
3     for(int i = 0; i < m; i++)
4     {
5         if(row[i] > max) max = row[i];
6     }
7     return max;
8 }
9
10 void process(string fileName) {
11     // TODO
12     ifstream inputfile(fileName);
13     if( inputfile.is_open()) {
14         int n , m;
15         inputfile >> n >> m;
16
17         double matrix[n][m];
18         for(int i = 0; i < n; i++)
19         {
20             for(int j = 0; j < m; j++)
21             {
22                 inputfile >> matrix[i][j] ;
23             }
24         }
25         for(int i = 0; i < n; i++)
26         {   cout << maxrow( matrix[i], m) << " " ;       }
27         double MAX = matrix[0][0];
28         for(int i = 0; i < n; i++)
29         {
30             for(int j = 0; j < m; j++)
31             {
32                 if(matrix[i][j] > MAX) MAX = matrix[i][j];
33             }
34         }
35         cout << MAX;
36     }
37 }
```



	Test	Input	Expected	Got	
✓	1	3 4 1 2 3 4 4.3 41.2 -4.576 0 -3 -2 -4 -1	4 41.2 -1 41.2	4 41.2 -1 41.2	✓

Passed all tests! ✓

