

Status	Finished
Started	Sunday, 12 January 2025, 3:25 AM
Completed	Sunday, 12 January 2025, 3:25 AM
Duration	5 secs
Marks	0.00/80.00
Grade	0.00 out of 10.00 (0%)

Question 1

Not answered

Mark 0.00 out of 10.00

[File - EasyQuiz]

Viết chương trình nhận vào một [xâu kí tự](#) là đường dẫn đến một tệp. Thử xem có mở được tệp đó không, nếu có in ra *YES* ngược lại in ra *NO*. Mở xong nhớ đóng tệp lại.

For example:

Input	Result
1.txt	YES

Answer: (penalty regime: 0 %)

1	
---	--

Question 2

Not answered

Mark 0.00 out of 10.00

[HiddenClue]

Trong trò chơi tìm kho báu, để tăng độ khó của trò chơi, người quản trò quyết định lưu gợi ý tiếp theo vào trong một tệp và cung cấp cho người chơi đường dẫn đến nơi lưu trữ tệp đó.

Với vai trò là một người chơi, bạn sẽ nhận được đường dẫn đến tệp lưu gợi ý, hãy tìm và hiển thị gợi ý đó cho mọi người trong đội của bạn cùng biết.

Viết hàm `void showClue (char* path)` nhận đầu vào là đường dẫn đến nơi lưu tệp. Hàm in ra toàn bộ gợi ý để đi đến địa điểm kế tiếp (độ dài tối đa của gợi ý là 10000 kí tự).

For example:

Input	Result
first_clue.txt	Go straight ahead then turn right on the 3th street.

Answer: (penalty regime: 0 %)

1	
---	--

Question 3

Not answered

Mark 0.00 out of 10.00

[CountCharactersFromFile]

[Thống kê tần suất](#) xuất hiện của một kí tự trong bảng chữ cái của một ngôn ngữ trong đời sống hàng ngày là một việc làm có nhiều giá trị. Chẳng hạn, trong mã hóa thông tin, người ta sẽ gán cho kí tự xuất hiện nhiều một mã có độ dài ngắn.

Giả sử bạn được giao phó trọng trách [thống kê](#) này. Bạn được cấp hàng loạt các tệp văn bản lưu lại thơ, văn, cuộc nói chuyện hàng ngày của một ngôn ngữ. Việc đầu tiên bạn cần làm là viết một hàm `int countCharacter(const char* path, char letter)` nhận tham số đầu vào là đường dẫn đến tệp văn bản và kí tự cần đếm số lần xuất hiện, hàm trả về số lần xuất hiện của kí tự truyền vào.

Lưu ý: bạn cần đếm số lần xuất hiện của kí tự mà không phân biệt chữ hoa chữ thường. Ví dụ đoạn văn bản sau có 7 kí tự *e*.

```
Everyone deserves a second shot.
```

Answer: (penalty regime: 0 %)

1	
---	--

Question 4

Not answered

Mark 0.00 out of 10.00

[CharacterCounter]

Viết hàm `int count(char sentence[], char character)`, nhận tham số truyền vào là một chuỗi kí tự `sentence` và một kí tự `character`, trả về số lần xuất hiện của kí tự `character` trong chuỗi `sentence`.

Lưu ý: bắt buộc phải viết hàm yêu cầu trong đề bài, không cần viết hàm `main()`, không cần thêm các dòng `#include` và có thể viết các hàm thêm tùy ý.

For example:

Input	Result
"We accept the love we think we deserve." e	9

Answer: (penalty regime: 0 %)

1	
---	--

Question 5

Not answered

Mark 0.00 out of 10.00

[toLowerCase]

Viết hàm `string toLower (string s)` chuyển đổi xâu kí tự s thành xâu kí tự có tất cả các kí tự là chữ cái in thường. Hàm nhận đầu vào là một xâu s bất kỳ và trả về xâu đã được chuyển về dạng chữ thường (lowercase).

Chú ý: Bạn chỉ phải viết đúng hàm theo yêu cầu. Hàm ***main()*** và thư viện đã có sẵn.

For example:

Test	Input	Result
toLowerCase(s)	Convert all the StrIng to LowerCase.	convert all the string to lowercase.

Answer: (penalty regime: 0 %)

1	
---	--

Question 6

Not answered

Mark 0.00 out of 10.00

[NumberOfLines]

Viết chương trình nhận vào một chuỗi kí tự chứa đường dẫn đến một tệp. Thử mở tệp đó, nếu không mở được thì in ra dòng "**Mission failed**", nếu mở được thì đếm xem tệp có bao nhiêu dòng. Biết rằng tệp là một tệp văn bản, mỗi dòng có thể chứa chuỗi kí tự, số thực, số nguyên hoặc dòng trống. Mở xong nhớ đóng tệp.

Gợi ý: Sử dụng kí tự kết thúc tệp.

For example:

Input	Result
1.txt	6

Answer: (penalty regime: 0 %)

1	
---	--

Question 7

Not answered

Mark 0.00 out of 10.00

Harry Potter là tên của bộ truyện (gồm bảy phần) của nữ nhà văn nước Anh J. K. Rowling. Bộ truyện viết về những cuộc phiêu lưu phù thủy của cậu bé Harry Potter và những người bạn Ronald Weasley, Hermione Granger, lấy bối cảnh tại [Trường Phù thủy và Pháp sư Hogwarts](#) nước Anh. Những cuộc phiêu lưu tập trung vào cuộc chiến của Harry Potter trong việc chống lại tên Chúa tể hắc ám Voldemort - người có tham vọng muốn trở nên bất tử, thống trị thế giới phù thủy, nô dịch hóa những người phi pháp thuật và tiêu diệt những ai cản đường hắn đặc biệt là Harry Potter.

Trong bộ truyện, người ta kể lại rằng Tom Riddle, một chàng trai dị thường người mà sau đó trở thành phù thủy đáng sợ nhất mọi thời đại, đã sáng tạo ra biệt danh xấu xa của mình như một kiểu đảo chữ (anagram). Tom Marvolo Riddle trở thành "I Am Lord Voldemort".

Hai chuỗi là anagram của nhau nếu các chữ cái của chuỗi này có thể được sắp xếp để tạo thành chuỗi kia. Hãy viết hàm `bool areAnagrams(string str1, string str2)` trả về `true` nếu hai chuỗi đầu vào là anagram của nhau ngược lại `false`. Hàm không phân biệt chữ hoa chữ thường và không quan tâm đến các dấu câu và dấu cách.

Lưu ý: chỉ cần viết hàm như đề bài yêu cầu, không cần viết hàm `main()`, không cần viết các câu lệnh `#include`, `using namespace std`

For example:

Test	Input	Result
<pre>string str1, str2; getline(cin >> ws, str1); getline(cin >> ws, str2); cout << boolalpha << areAnagrams(str1, str2);</pre>	<pre>Tom Marvolo Riddle I Am Lord Voldemort.</pre>	true

Answer: (penalty regime: 0 %)

1	
---	--

Mark 0.00 out of 10.00

[Back to Course](#)