# **C-string**

## Câu 1.1: Nguồn DB: Loại bỏ các khoảng trắng thừa trong chuỗi

Viết hàm void process(char \*str, char \*outstr) loại bỏ các khoảng trắng thừa trong chuỗi sao cho không còn hai khoảng trắng liền kề nhau và không có khoảng trắng nào ở đầu và cuối chuỗi.

Đầu vào:

Con trỏ char \* đến phần tử đầu tiên của chuỗi

Đầu ra:

Kết quả được ghi vào chuỗi outstr

**For example:**

|  |  |  |
| --- | --- | --- |
| **Test** | **Input** | **Result** |
| 1 | this is my school | this is my school |

## 

## Câu 1.2: Nguồn DB: Đảo ngược các ký tự trong chuỗi

Viết hàm void process(char \*str, char \*outstr) đảo ngược các ký tự trong chuỗi.

Đầu vào:

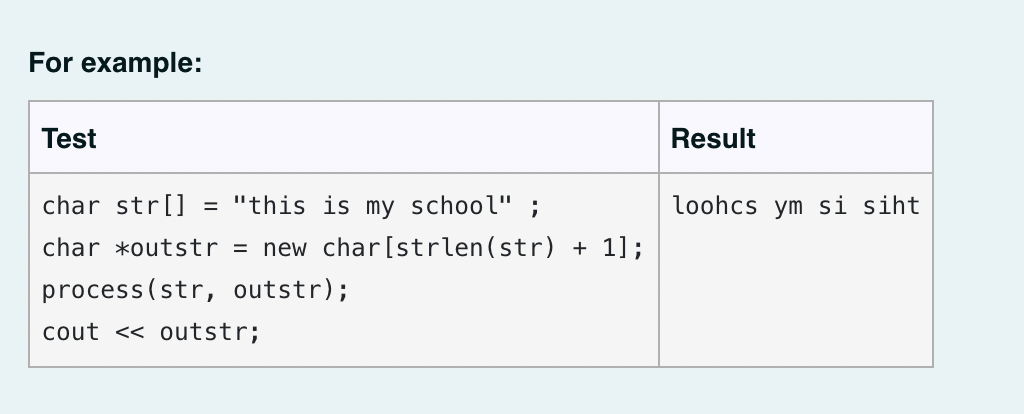
Con trỏ char \* đến phần tử đầu tiên của chuỗi

Đầu ra:

Kết quả được ghi vào chuỗi outstr

**For example:**

|  |  |  |
| --- | --- | --- |
| **Test** | **Input** | **Result** |
| 1 | this is my school | loohcs ym si siht |



## 

## Câu 1.3: Nguồn DB: Chuẩn hóa tên người

Viết hàm void process(char \*name) chuẩn hóa tên người. Quy tắc chuẩn hóa như sau:

1. Chữ cái đầu tiên của mỗi từ phải ghi hoa, các chữ cái còn lại ghi thường.

2. Tên người chỉ bao gồm những ký tự trong bảng chữ cái alphabet.

3. Không có các khoảng trắng thừa. Cụ thể, không có 2 khoảng trắng liền kề nhau và không có khoảng trắng nào ở đầu và cuối chuỗi.

Đầu vào:

Con trỏ char \* đến phần tử đầu tiên của chuỗi

Đầu ra:

Tên người đã được chuẩn hóa theo những nguyên tắc trên

**For example:**

|  |  |  |
| --- | --- | --- |
| **Test** | **Input** | **Result** |
| 1 | ngUYEn V$$AN @#$ thanh\* | Nguyen Van Thanh |

## 

## Câu 1.4: Nguồn DB: Tìm từ đầu tiên trong chuỗi bị lặp lại

Viết hàm void printRepetition(char \*str) in ra từ đầu tiên trong chuỗi bị lặp lại.

Đầu vào:

Con trỏ char \* đến phần tử đầu tiên của chuỗi

Đầu ra:

Từ đầu tiên trong chuỗi bị lặp lại được in trên console. Nếu không có từ nào lặp lại thì in lại chuỗi

**For example:**

|  |  |  |
| --- | --- | --- |
| **Test** | **Input** | **Result** |
| 1 | student teacher student teacher | student |
| 2 | student teacher | student teacher |

## Câu 1.5: Nguồn DB: Tính giá trị của một chuỗi các bit nhị phân

Viết hàm int calc(char \*str) trả về giá trị dạng thập phân của chuỗi các bit nhị phân

Đầu vào:

Con trỏ char \* đến phần tử đầu tiên của chuỗi

Đầu ra:

Giá trị dạng thập phân của chuỗi các bit nhị phân

**For example:**

|  |  |  |
| --- | --- | --- |
| **Test** | **Input** | **Result** |
| 1 | 1001 | 9 |
| 2 | 111 | 7 |

## 

# 

# **Class string**

## Câu 2.1: Nguồn EL : Tìm độ dài lớn nhất của chuỗi con chỉ tạo thành bởi một ký tự trong chuỗi s.

## Viết chương trình đọc vào chuỗi s, tìm tất cả các chuỗi con trong s mà chuỗi con chỉ có duy nhất 1 kí tự, ví dụ : aaaa, bbbb, cc, trả về độ dài của chuỗi con dài nhất.

## 

## Đầu vào:

## Chuỗi s có độ dài ngắn nhất là 1.

## Đầu ra:

## Độ dài của chuỗi con dài nhất mà chỉ có sự xuất hiện của 1 ký tự trong chuỗi con.

## For example:

|  |  |  |
| --- | --- | --- |
| Test | Input | Result |
| 1 | abcabcbb | 2 |

## 

## 

## Câu 2.2: Nguồn EL: Tìm độ dài lớn nhất của chuỗi con không có ký tự nào được lặp lại trong chuỗi s.

## Viết chương trình đọc vào chuỗi s từ bàn phím, tìm và trả về độ dài của chuỗi con dài nhất không có ký tự nào được lặp lại có trong s.

## 

## Đầu vào:

## Chuỗi s.

## Đầu ra:

## Độ dài của chuỗi con dài nhất không có ký tự nào được lặp lại có trong s.

## For example:

|  |  |  |
| --- | --- | --- |
| Test | Input | Result |
| 1 | abcabcbb | 3 |

## 

## Câu 2.3: Nguồn EL : Tìm độ dài lớn nhất của chuỗi con có tính chất đối xứng trong chuỗi s.

## Viết chương trình đọc vào chuỗi s từ bàn phím, tìm và trả về chuỗi con có tính chất đối xứng dài nhất có trong s.

## 

## Đầu vào:

## Chuỗi s.

## Đầu ra:

## Chuỗi con có tính chất đối xứng dài nhất có trong s. Trong trường hợp có nhiều chuỗi con dài nhất có cùng độ dài, trả về chuỗi con có vị trí bắt đầu nhỏ nhất.

## For example:

|  |  |  |
| --- | --- | --- |
| Test | Input | Result |
| 1 | babad | bab |

## 

## Câu 2.4: Nguồn DB: Thay thế sự xuất hiện cuối cùng của chuỗi con s1 thành chuỗi con s2 trong chuỗi s.

## Hiện thực hàm void replaceString(string s, string s1, string s2) để in ra chuỗi s sau khi đã thay thế chuỗi con s1 cuối cùng thành s2 có trong s.

## Gợi ý: Tham khảo hàm string::replace trong thư viện <string>.

## For example:

|  |  |
| --- | --- |
| Test | Result |
| string s = "Truong Dai Hoc Bach Khoa, Sai Gon";  string s1 = "ai";  string s2 = "ong"; ReplaceString(s, s1, s2); | Truong Dai Hoc Bach Khoa, Song Gon |

## 

## Câu 2.5: Nguồn DB: Xóa một từ trong chuỗi.

## Hiện thực hàm deleteWord(string s, string s1) với chức năng in ra chuỗi s sau khi xóa tất cả các chuỗi s1 có trong s.

## For example:

|  |  |
| --- | --- |
| Test | Result |
| **string s = "Truong Dai Hoc Bach Khoa, Sai Gon";**  **string s1 = "ai ";** deleteWord(s, s1); | Truong DHoc Bach Khoa, SGon |

## 

## Câu 2.6: Nguồn NE: Tìm tất cả vị trí xuất hiện của chuỗi con trong chuỗi s

Hiện thực hàm findAllIndex(string s1, string s2) để in ra tất cả các vị trí xuất hiện của kí tự đầu tiên của chuỗi s2 trong chuỗi s1. Nếu không tìm thấy in ra -1. Các vị trí tìm thấy sẽ cách nhau một khoảng trắng, sau vị trí cuối cùng, không in thêm bất kỳ ký tự nào (kể cả khoảng trắng, dấu xuống hàng)

Gợi ý: Sử dụng hàm string::find trong thư viện <string>.

For example:

|  |  |
| --- | --- |
| Test | Result |
| string s1 = “Truong Dai Hoc Bach Khoa.”;  string s2 = “a”;  findAllIndex(s1, s2); | 8 16 23 |

## Câu 2.7: Nguồn NE: In ra một phần của chuỗi.

Hiện thực hàm cutString(string s, int index) để in ra chuỗi con của chuỗi s từ vị trí index đến hết (index tính từ 0).

Gợi ý : Sử dụng hàm string::substr trong thư viện <string>.

For example:

|  |  |
| --- | --- |
| Test | Result |
| string s = “Truong Dai Hoc Bach Khoa.”;  cutString (s, 7); | Dai Hoc Bach Khoa. |

## 

## 

# 

# **Multi-dimensional Array**

## Câu 3.1. [NE]

## Cho mảng 2 chiều chứa các số nguyên, kích thước M x N.

## Hiện thực hàm:

## int findMaxColumn(vector<vector<int>>& arr, int row, int col);

## Trong đó; arr, row và col lần lượt là mảng 2 chiều, số hàng và số cột của mảng. Tìm chỉ số của cột có tổng tất cả các phần tử lớn nhất.

## Lưu ý:

## - Cột đầu tiên được đánh chỉ số 0.

## - Nếu có nhiều hơn một cột có tổng lớn nhất, ta chọn cột có chỉ số lớn nhất.

## Tham khảo ví dụ để hiểu về format đầu ra.

## For example:

|  |  |
| --- | --- |
| Test | Result |
| vector<vector<int>> arr = {{6,7,9,4,8},{3,2,7,4,1},{9,4,5,8,3}};int m = 3, n = 5;cout << findMaxColumn(arr, m, n); | 2 |

## 

## Câu 3.2. [NE]

## Cho mảng 2 chiều chứa các số nguyên, kích thước N x N.

## Hiện thực hàm:

## int diagonalProduct(vector<vector<int>>& arr, int row, int col);

## Trong đó; arr, row và col lần lượt là mảng 2 chiều, số hàng và số cột của mảng. Tìm tích của tất cả các phần tử trong đường chéo chính của mảng này.

## Tham khảo ví dụ để hiểu về format đầu ra.

## For example:

|  |  |
| --- | --- |
| Test | Result |
| vector<vector<int>> arr = {{6,7,9},{3,2,7},{9,4,5}};int m = 3, n = 3;cout << diagonalProduct(arr, m, n); | 60 |

## 

## Câu 3.3. [NE]

## Cho mảng 2 chiều chứa các số nguyên, kích thước N x N.

## Hiện thực hàm:

## bool isSymmetric(vector<vector<int>>& arr, int row, int col);

## Trong đó; arr, row và col lần lượt là mảng 2 chiều, số hàng và số cột của mảng.

## Hàm này dùng để Kiểm tra xem mảng này có phải là một ma trận đối xứng hay không.

Ma trận đối xứng là ma trận luôn có giá trị của phần tử ở hàng i, cột j bằng với giá trị của phần từ ở hàng j cột i với mọi giá trị hợp lệ của i và j.

## Lưu ý:

## - Trả về True nếu đây là ma trận đối xứng, ngược lại trả về False.

## Tham khảo ví dụ để hiểu về format đầu ra.

## For example:

|  |  |
| --- | --- |
| Test | Result |
| vector<vector<int>> arr = {{6,7,9},{7,2,4},{9,4,5}};int m = 3, n = 3;cout << isSymmetric(arr, m, n); | 1 |

## 

## Câu 3.4. [NE]

## Cho mảng 2 chiều chứa các số nguyên, kích thước M x N.

## Hiện thực hàm:

## int ascendingRows(vector<vector<int>>& arr, int row, int col);

## Trong đó; arr, row và col lần lượt là mảng 2 chiều, số hàng và số cột của mảng. Một hàng trong mảng được gọi là “HN1” nếu trong hàng đó, mỗi phần tử đứng trước đều có giá trị không lớn hơn tất cả các phần tử đứng sau nó. Tìm số hàng “HN1” trong mảng.

## Tham khảo ví dụ để hiểu về format đầu ra.

## For example:

|  |  |
| --- | --- |
| Test | Result |
| vector<vector<int>> arr = {{6,7,7},{7,2,4},{9,4,5}};int m = 3, n = 3;cout << ascendingRows (arr, m, n); | 1 |

## 

## Câu 3.5. [NE]

## Cho mảng 2 chiều chứa các số nguyên, kích thước M x N.

## Hiện thực hàm:

## int primeColumns(vector<vector<int>>& arr, int row, int col);

## Trong đó; arr, row và col lần lượt là mảng 2 chiều, số hàng và số cột của mảng. Một cột trong mảng được gọi là “HN2” nếu tổng tất cả các phần tử trong cột đó là một số nguyên tố. Tìm số cột “HN2” trong mảng.

## Tham khảo ví dụ để hiểu về format đầu ra.

## For example:

|  |  |
| --- | --- |
| Test | Result |
| vector<vector<int>> arr = {{6,7,7},{7,2,4},{9,4,5}};int m = 3, n = 3;cout << primeColumns(arr, m, n); | 1 |

## 

## Câu 3.6. [NE]

## Cho mảng 2 chiều chứa các số nguyên, kích thước M x N.

## Hiện thực hàm:

## int specialCell(vector<vector<int>>& arr, int row, int col);

## Trong đó; arr, row và col lần lượt là mảng 2 chiều, số hàng và số cột của mảng. Một phần tử [Tiến: một ô] trong mảng được gọi là “HN3” nếu tổng tất cả các phần tử trong hàng chứa ô đó và tổng các phần tử trong cột chứa ô đó đều là số nguyên tố. Tìm số ô “HN3” trong mảng.

## Tham khảo ví dụ để hiểu về format đầu ra.

## For example:

|  |  |
| --- | --- |
| Test | Result |
| vector<vector<int>> arr = {{6,7,7},{7,2,8},{9,4,5}};int m = 3, n = 3;cout << specialCell (arr, m, n); | 1 |

## 

## Câu 3.7 . [NE]

## Cho mảng 2 chiều chứa các số nguyên, kích thước M x N.

## Hiện thực hàm:

## int diagonalDiff(vector<vector<int>>& arr, int row, int col, int x, int y);

## Trong đó; arr, row, col, x, y lần lượt là mảng 2 chiều, số hàng, số cột của mảng, chỉ số hàng, chỉ số cột. Tìm giá trị tuyệt đối của hiệu giữa các đường chéo chứa ô có chỉ số hàng là x và chỉ số cột là y. (0 <= x < row, 0 <= y < col).

## Tham khảo ví dụ để hiểu về format đầu ra.

## For example:

|  |  |
| --- | --- |
| Test | Result |
| vector<vector<int>> arr = {{6,7,7},{7,2,8},{9,4,5}};int m = 3, n = 3, x = 1, y = 0;cout << diagonalDiff (arr, m, n, x, y); | 3 |

## hiệu giữa các đường chéo là. ……

## Câu 3.8. [NE]

## Cho mảng 2 chiều chứa các số nguyên, kích thước M x N.

## Hiện thực hàm:

## int subMatrix(vector<vector<int>>& arr, int row, int col);

## Trong đó; arr, row và col lần lượt là mảng 2 chiều, số hàng và số cột của mảng. Tìm số mảng con có kích thước 2x2 có tổng tất cả các phần tử là một số lẻ

## Tham khảo ví dụ để hiểu về format đầu ra.

## For example:

|  |  |
| --- | --- |
| Test | Result |
| vector<vector<int>> arr = {{6,7,7},{7,2,8},{9,4,5}};int m = 3, n = 3;cout << subMatrix (arr, m, n, x, y); | 1 |

## 

## 

## 

# 

# **File IO**

## Câu 4.1 [EL]

Viết chương trình đọc vào 3 ký tự và xác định xem 3 ký tự có đúng thứ tự bảng chữ cái (ASCII), nếu đúng xuất ra "true", ngược lại xuất ra "false". Chương trình sẽ lặp cho đến khi 3 ký tự đọc vào là "\*\*\*".

Đầu vào:

Mỗi lần đọc vào 3 ký tự cho đến khi 3 ký tự đọc vào là "\*\*\*".

Đầu ra:

"true" hoặc "false", theo sau bởi dấu xuống dòng, trừ trường hợp đọc vào "\*\*\*" thì kết thúc chương trình.

**For example:**

|  |  |  |
| --- | --- | --- |
| **Test** | **Input** | **Result** |
| 1 | 123  abc  aab  \*\*\* | true  true  true |

## Câu 4.2 [NE]

Viết chương trình tính tổng của các số nguyên không âm được đọc vào từ file input.txt (các số phân cách với nhau bằng một khoảng trắng).

Đầu vào:

File input.txt chứa các số nguyên không âm (các số phân cách với nhau bằng một khoảng trắng).

Đầu ra:

Một số nguyên là tổng của các số nguyên đầu vào.

For example:

|  |  |  |
| --- | --- | --- |
| **Test** | **Input** | **Result** |
| 1 | 0 2 3 5 | 10 |

## Câu 4.3 [NE]

Viết chương trình đọc vào một chuỗi S từ bàn phím, sau đó chuyển tất các ký tự trong chuỗi S thành ký tự viết HOA và ghi chuỗi đã chuyển đổi vào file “output.txt”. Biết chuỗi S chỉ gồm các chữ cái và chữ số.

Chú ý: chỉ thay đổi các chữ cái in thường, các chữ số và chữ cái in hoa sẽ được giữ nguyên.

Đầu vào:

Một chuỗi S được nhập vào từ bàn phím.

Đầu ra:

Một chuỗi chỉ gồm các ký tự IN HOA được xuất ra file output.txt

For example:

|  |  |  |
| --- | --- | --- |
| **Test** | **Input** | **Result** |
| 1 | kythuatLapTrInh212 | KYTHUATLAPTRINH212 |

## Câu 4.4 [EL]

Viết hàm void process(string fileName) đọc vào 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:

Một chuỗi chứa tên file.

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.

**For example:**

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

## Câu 4.5 [NE]

Viết chương trình đọc vào file input.txt, dòng đầu gồm 1 số nguyên dương N, N dòng theo sau mỗi dòng chứa 4 số thực lần lượt là điểm số của các môn NMĐT, KTLT, DSA và PPL của N học sinh.

Điểm trung bình (ĐTB) của sinh viên sẽ là trung bình cộng của 4 cột điểm trên. Sinh viên sẽ được xếp loại dựa trên ĐTB như sau:

* Loại A nếu ĐTB >= 8 và không có môn nào dưới 5.
* Loại B nếu 8 > ĐTB >= 6.5 và không có môn nào dưới 5.
* Loại C nếu 6.5 > ĐTB >= 5 và không có môn nào dưới 5
* Loại D cho các trường hợp còn lại

Xác định số lượng sinh viên mỗi loại và xuất kết quả ra file “output.txt”.

Đầu vào:

File input.txt chứa thông tin về điểm số của sinh viên

Đầu ra:

Số lượng sinh viên mỗi loại được ghi ra file output.txt

**For example:**

|  |  |  |
| --- | --- | --- |
| **Test** | **Input** | **Result** |
| 1 | 5  8 8 8 6  9 9 9 5  10 10 3 10  5 5 6 6  8 9 10 7 | A 2 B 1 C 1 D 1 |

## Câu 4.6 [NE]

Viết chương trình đọc vào 2 file price.txt và buy.txt. Với file price.txt có nội dung cố định.

Trong đó:

* File price.txt chứa thông tin về các sản phẩm, gồm: Dòng đầu tiên chứa số nguyên dương N là số lượng sản phẩm hiện có trong cửa hàng. N dòng tiếp theo mỗi dòng chứa 2 số nguyên dương là ID và giá của các sản phẩm tương ứng

Ví dụ:

3

1 10

2 15

3 12

Có nghĩa là: hiện tại có 3 sản phẩm được bán trong cửa hàng, sản phẩm 1 giá 10

sản phẩm 2 giá 15 và sản phẩm 3 có giá là 12.

* File buy.txt chứa thông tin về việc mua hàng hóa của khách hàng. Dòng đầu tiên chứa số nguyên dương M là số lượng khách hàng mua hàng trong 1 ngày. M dòng tiếp theo mỗi dòng chứa thông tin như sau: sẽ có giá nhiều trị nằm trên một hàng, cách nhau bởi một khoảng trắng. Giá trị đầu tiên sẽ là tên khách hàng (dạng string), các giá trị còn lại sẽ đi theo cặp với nhau, tương ứng là: <ID sản phẩm>\_<số lượng tương ứng>

Ví dụ:

2

A 1 2 2 3

B 1 3 3 2

Có nghĩa là có 2 khách hàng A và B, A mua 2 sản phẩm loại 1 và 3 sản phẩm loại 2; B mua 3 sản phẩm loại 1 và 2 sản phẩm loại 3.

Lưu ý: ID của sản phẩm và tên khách hàng là duy nhất (không lặp lại).

Tính số tiền mỗi khách hàng cần trả, sau đó xuất kết quả ra file “output.txt”.

**For example:**

|  |  |  |
| --- | --- | --- |
| **Test** | **Input** | **Result** |
| 1 | //price.txt  3  1 10  2 15  3 12  //buy.txt  2  A 1 2 2 3  B 1 3 3 2 | A 65  B 54 |

## Câu 4.7 [NE]

Viết chương trình đọc vào 3 file library.txt và book.txt và author.txt.

Trong đó:

* File library.txt chứa thông tin của các thư viện, gồm: dòng đầu tiên chứa số nguyên dương N là số lượng thư viện được khảo sát. N dòng tiếp theo, mỗi dòng chứa 6 giá trị được phân cách nhau bằng dấu khoảng trắng. Cho mỗi dòng, giá trị đầu tiên là tên của Thư viện (tên Thư viện là duy nhất), 5 giá trị còn lại là 5 số nguyên dương, là ID của 5 quyển sách có trong thư viện đó.

Ví dụ:

5

LA 1 2 3 4 5

LB 6 7 8 9 10

LC 1 3 5 7 9

LD 2 4 6 8 10

LE 1 4 7 9 2

* File book.txt chứa thông tin của các quyến sách, gồm: dòng đầu tiên chứa số nguyên dương M là số lượng đầu sách có trong tất cả các thư viện. M dòng tiếp theo mỗi dòng chứa 3 giá trị (phân cách nhau bởi một dấu khoảng trắng) có ý nghĩa như sau: giá trị đầu tiên là một số nguyên dương đại diện cho Mã số sách (ID - ID là duy nhất), giá trị thứ 2 là năm sản xuất và giá trị cuối cùng là thể loại.

Ví dụ:

10

1 2000 A

2 2001 B

3 1993 D

4 1997 A

5 1995 C

6 1990 A

7 1980 D

8 1979 A

9 2015 B

10 2009 C

* File author.txt chứa thông tin của các tác giả, gồm: dòng đầu tiên chứa số nguyên dương P là số lượng tất cả các tác giả của các sách trong các thư viện (giả sử 1 quyển sách chỉ được sáng tác bới 1 tác giả). P dòng tiếp theo mỗi dòng chứa các giá trị như sau (các giá trị được phân cách với nhau bằng 1 dấu khoảng trắng): giá trị đầu tiên là Tên tác giả, các giá trị còn lại là ID của các quyển sách mà người đó đã sáng tác.

Ví dụ:

6

David 1 5

John 3

Henry 2 4

William 6

Tony 10

Brian 8 9

Smith 7

Xác định xem Thư viện L có chứa tác phẩm nào của Tác giả A hay không? Với L và A được nhập vào từ bàn phím.

**For example:**

|  |  |  |
| --- | --- | --- |
| **Test** | **Input** | **Result** |
| 1 | LA Brian | False |

|  |  |  |
| --- | --- | --- |
| **Test** | **Input** | **Result** |
| 2 | LA David | True |