# Đếm chữ cái Tên file: CHAR.CPP hoặc CHAR.PAS

Nhập xâu. Không phân biệt viết hoa hay viết thường, hãy in ra các kí tự chữ cái có mặt trong xâu và số lần xuất hiện của nó (ví dụ xâu “Trach - Van - Doanh” có chữ a xuất hiện 3 lần, c(1), d(1), h(2), n(2), o(1), r(1), t(1), v(1).

INPUT: CHAR.INP

* xâu s.

OUTPUT: CHAR.OUT

- gồm nhiều dòng, mỗi dòng chứa 1 ký tự x - ký tự có mặt trong xâu s - và một số nguyên dương k - số lần xuất hiện của x trong s, x và k cách nhau 1 dấu trắng.

- Các ký tự được hiện theo thứ tự tăng dần và là các chữ cái viết thường

Ví dụ

|  |  |
| --- | --- |
| **INPUT** | **OUTPUT** |
| Trach Van Doanh | a 3  c 1  d 1  h 2  n 2  o 1  r 1  t 1  v 1 |

# Điệp viên tên file: SPY.CPP hoặc SPY.PAS

Bộ phim mới Thành Long (diễn viên nổi tiếng của Hollywood) vào vai một điệp viên. Trong phim anh được tổ chức giao nhiệm vụ theo dõi một tên trùm Mafia. Theo tìm hiểu của anh, tên trùm Mafia thường vừa đi vừa trao đổi điện thoại trên xe ô tô trong thị trấn. Vì vậy, anh quyết định ngồi ở một vị trí nào đó và nghe lỏm thông tin trao đổi của hắn.

Trị trấn được biểu diễn như là một mặt phẳng hệ trục tọa độ. Trùm Mafia đi từ vị trí (0,0), và mỗi vị trí hắn có thể di chuyển 1 đơn vị theo một trong 4 hướng: phải, trái, lên hoặc xuống.

Chàng điệp viên đang ngồi ở vị trí (x,y), và anh chỉ nghe lỏm được về thông tin của tên trùm nếu vị trí của anh trùng vị trí của tên trùm hoặc vị trí tên trùm thuộc 1 trong 8 vị trí liền kề với vị trí của anh.

Nhiệm vụ của bạn là viết chương trình tìm các thời điểm mà Chàng điệp viên đã nghe được cuộc trò chuyện.

Dữ liệu vào: SPY.INP

* Dòng đầu chứa hai số nguyên x và y, -104 ≤ x, y ≤ 104 là vị trí của chàng điệp viên
* Dòng tiếp theo chứa số nguyên k, 1 ≤ k ≤ 105, là độ dài đường di chuyển của trên trùm Mafia
* Dòng tiếp theo chứa K ký tự chỉ cho ta biết con đường mà tên trùm đi hôm đó. Trong đó ký tự ‘R’ – Phải, ‘U’ – Trên, ‘L’ – trái, và ‘D’ – dưới.

Kết quả ra: SPY.OUT

* Nếu không thể nghe được cuộc trò chuyện nào thì ghi ra số -1.
* Nếu nghe được các cuột trò chuyện thì ghi ra các thời điểm nghe được cuộc trò chuyện. Mỗi thời điểm trên một dòng.

Ví dụ:

|  |  |  |  |
| --- | --- | --- | --- |
| **INPUT** | **OUTPUT** | **INPUT** | **OUTPUT** |
| 2 2  3  RRR | -1 | -1 0  8  DDLLUURU | 0  1  5  6  7  8 |

# Dãy ngoặc Tên file: NGOAC.CPP hoặc NGOAC.PAS

Người ta định nghĩa một dãy ngoặc đúng theo đệ quy như sau:

- Xâu A là xâu rỗng là một dãy ngoặc đúng

- Nếu xâu A là dãy ngoặc đúng thì (A) cũng là dãy ngoặc đúng

- Nếu xâu A và xâu B là dãy ngoặc đúng thì AB cũng là dãy ngoặc đúng.

Còn những xâu chứa những ký tự khác "(" và ")" được gọi là xâu không hợp lệ.

Ví dụ:

S = "(A()B)" là dãy "KHONG HOP LE" vì chứa chữ cái A và B

S = "( )( )(( ))" là dãy ngoặc "DUNG"

S = "( )( ))" là dãy ngoặc "KHONG DUNG"

Cho xâu S. Hãy kiểm tra xâu S là dãy ngoặc "DUNG", "KHONG DUNG" hay là xâu "KHONG HOP LE"

INPUT: NGOAC.INP:

* Một xâu S chứa tối đa 106 phần tử

OUTPUT: NGOAC.OUT:

* Thông báo "KHONG HOP LE" nếu xâu không hợp lệ
* Thông báo "DUNG" nếu xâu đúng
* Thông báo "KHONG DUNG" nếu xâu không đúng

Ví dụ:

|  |  |
| --- | --- |
| **INPUT** | **OUTPUT** |
| (A()B) | KHONG HOP LE |
| ()()(()) | DUNG |
| ((()) | KHONGDUNG |

# Độ đo Tên file: AD.CPP hoặc AD.PAS

Hai xâu ký tự được gọi là đảo của nhau nếu ta có thể hoán vị các ký tự của xâu để được xâu còn lại. Ví dụ: xâu “occurs” là đảo của xâu “succor”, tuy nhiên “dear” không phải là đảo của xâu “daerd” (vì chữ “d” xuất hiện 2 lần trong “deard” còn “dear” thì chỉ xuất hiện 1 lần).

Độ đo giữa 2 xâu ký tự là số ký tự ít nhất cần phải xóa (trên cả 2 xâu) để 2 xâu còn lại là đảo của nhau. Ví dụ: độ đo giữa hai xâu “sleep” và “leap” là 3, độ đo giữa hai xâu “dog” và “cat” là 6.

**Yêu cầu:** Hãy tìm độ đo giữa hai xâu cho trước.

**Dữ liệu vào:** file **AD.INP:**

- Gồm 2 dòng, mỗi dòng chứa một xâu ký tự chỉ gồm các chữ cái tiếng Anh thường, mỗi dòng chứa không quá 1 triệu ký tự.

**Kết quả:** file **AD.OUT:**

- Một số nguyên duy nhất là độ đo giữa hai xâu cho trước.

Ví du:

|  |  |
| --- | --- |
| **AD.INP** | **AD.OUT** |
| begin  end | 4 |

# ROBOT Tên file: ROBOT.CPP hoặc ROBOT.PAS

Một robot xuất phát từ vị trí (0,0) mặt quay về hướng Bắc (hướng lên trên). Mỗi lần chỉ có một trong 4 lệnh chuyển động là G, L, R, B tương ứng là tiến lên trên phía trước mặt, tiến sang trái, tiến sang phải, quay lại phía sau lưng một đơn vị. Cho dãy lệnh chuyển động. Hãy tìm xem vị trí cuối cùng của robot là vị trí nào?

Input:ROBOT.INP

+Dòng đầu tiên ghi n (n≤100) là số lệnh robot cần thực hiện.

+Dòng thứ hai là dãy n ký tự mô tả dãy lệnh robot thực hiện

Output:ROBOT.OUT

- Hai số nguyên là tọa độ (x,y) của vị trí cuối cùng robot.

Ví dụ:

|  |  |
| --- | --- |
| **INPUT** | **OUTPUT** |
| 4  BLGR | 2 -2 |

# KHỐI LẬP PHƯƠNG *Tên file:* ABBAA.CPP hoặc ABBAA.PAS

Jim là một chú bé, bình thường như bao chú bé khác. Jim còn rất bé nên chưa biết đếm. Một hôm, bố mẹ mang về cho chú bộ đồ chơi là các khối hộp kích thước 2×1×1 có in chữ cái trên các mặt của khối lập phương đơn vị. Jim còn quá bé, bố mẹ không muốn làm chú rối trí nên chỉ chọn các khối hộp chỉ ghi 2 chữ cái đầu tiên là ***A*** và ***B***.

Chú rất khoái chí và bắt đầu chơi, nhưng chơi không thật đúng cách: bẻ một số khối thành các hình lập phương để xếp nhà, sau đó, chú không thích các khối lập phương có chữ B và làm mất tất cả các khối này. Như vậy chú chỉ còn các khối hộp có ghi ***AB*** hoặc ***BA*** và một số khối lập phương có ghi chữ ***A***. Lúc bố mẹ đi vắng cậu tìm được lọ keo dán. Là một cậu bé sáng ý, Jim nhanh chóng hiểu được tác dụng của vật này và dán các khối hộp cùng các khối lập phương thành các băng, băng thứ ***i*** có kích thước 1×1×***mi***.

Khi bố mẹ về nhà Jim tự hào khoe kết quả lao động của mình. Qua mô tả rắm rối của Jim bố mẹ hiểu rằng Jim đã dán được ***n*** băng, băng thứ ***i*** biểu diễn xâu ***si*** (chỉ chứa các ký tự ***A*** và ***B***). Bố mẹ Jim muốn biết mình đã hiểu đúng những gì Jim mô tả hay chưa, cụ thể là liệu có tạo được xâu ***si*** từ những gì mà Jim có hay không.

***Yêu cầu:*** Cho ***n*** (1 ≤ ***n*** ≤ 10) và các xâu ***si*** (***i*** = 1 ÷ ***n***), tổng độ dài các xâu không vượt quá 105. Với mỗi xâu hãy xác định là có thể tồn tại hay không và đưa ra câu trả lời “***YES***” hoặc “***NO***”.

***Dữ liệu:*** Vào từ file văn bản ABBAA.INP:

* Dòng đầu tiên chứ a số nguyên ***n***,
* Dòng thứ ***i*** trong ***n*** dòng sau chứa xâu ***si***.

***Kết quả:*** Đưa ra file văn bản ABBAA.OUT các câu trả lời “***YES***” hoặc “***NO***”, mỗi câu trên một dòng.

***Ví dụ:***

|  |  |  |
| --- | --- | --- |
| ABBAA.INP |  | ABBAA.OUT |
| **5**  **A**  **B**  **ABBA**  **BABBA**  **ABBAA** |  | **YES**  **NO**  **YES**  **NO**  **YES** |
|  |

# Duyệt binh Tên file: ARMY.CPP hoặc ARMY.PAS

Vào năm 1945, Liên Xô đang đánh nhau với phát xít Đức hết sức ác liệt. Hàng triệu thanh niên Liên Xô phải lên đường nhập ngũ. Một cuộc duyệt binh diễn ra, các tân binh không biết đứng quay mặt về bên nào liền xếp tùy ý, vị tổng chỉ huy thấy thể liền ra lệnh: “Nếu hai tân binh liên tiếp và đối mặt với nhau thì ngay lập phải quay ngược lại(180 độ), động tác này diễn ra trong vòng 1s!”. Người tổng chỉ huy muốn biết sau bao lâu thì thì đội hình sẽ ngừng quay?

Input: ARMY.INP

* Dòng đầu ghi số nguyên N là số tân binh.
* Dòng thứ hai gồm đúng N kí tự ‘<’, ‘>’ thể hiện cách đứng của các tân binh. Nếu hai tân binh liên tiếp quay mặt vào nhau thì sẽ được biểu diễn bởi ‘><’.

Output: ARMY.OUT

* Gồm một số duy nhất ghi thời gian ít nhất để đội hình ngừng quay.

Giới hạn:

Kích thước: 1 ≤ n ≤ 1000000

Thời gian: 1 s/test

Bộ nhớ: 2 MB

Ví dụ:

|  |  |
| --- | --- |
| ARMY.INP | ARMY.OUT |
| 4  <><> | 1 |

Giải thích:

Tại thời điểm 0: <><>

Tại thời điểm 1: <<>>

# Clumsy Cows Tên file: CLUMSY.CPP hoặc CLUMSY.PAS

Bessie là cô bò đang cố gắng gõ một chuỗi ngoặc đơn cân bằng vào máy tính xách tay của cô ta, nhưng cô ta rất vụng về (do móng guốc khá lớn) nên cô ta hay gõ mất các kí tự. Hãy giúp cô ta tính xem có bao nhiêu kí tự trong chuỗi ngoặc đơn cần phải được đổi chiều (có nghĩa là đổi dấu mở ngoặc đơn thành dấu đóng ngoặc đơn, và ngược lại) để chuỗi ban đầu thành chuỗi cân bằng.

Có nhiều cách để định nghĩa một chuỗi ngoặc là “cân bằng.” Cách dễ nhất là số lượng dấu mở ngoặc ‘(‘ bằng số lượng dấu đóng ngoặc ‘)’, và với bất kì chuỗi tiền tố nào, số lượng dấu mở ngoặc đơn ‘(‘ phải lớn hơn hoặc bằng số lượng dấu đóng ngoặc đơn ‘)’. Trong những ví dụ sau đây, những chuỗi ở bên dưới là chuỗi cân bằng:

()

(())

()(()())

Những chuỗi sau đây là chuỗi không cân bằng:

)(

())(

((())))

INPUT: CLUMSY.INP

* Một chuỗi dấu ngoặc đơn có độ dài tối đa là 100.000 kí tự.

OUTPUT: CLUMSY.OUT

* Một số tự nhiên duy nhất là số lượng dấu ngoặc đơn nhỏ nhất cần phải được đổi chiều để biến chuỗi ban đầu thành chuỗi cân bằng.

Ví dụ:

|  |  |
| --- | --- |
| CLUMSY.INP | CLUMSY.OUT |
| ())( | 2 |

# Lớp học khiêu vũ Tên file: DANCE.CPP hoặc DANCE.PAS

Lớp học múa khiêu vũ dạ hội của giáo sư Padegras có ***n*** học sinh nam và nữ ghi tên. Giáo sư cho tất cả học sinh xếp thành một hàng dọc và chọn một nhóm các học sinh liên tiếp nhau cho buổi học đầu tiên với yêu cầu là số học sinh nam và nữ phải bằng nhau.

Hãy xác định, giáo sư Padegras có bao nhiêu cách lựa chọn khác nhau cho buổi học đầu tiên.

**Input:**

* Dòng đầu tiên chứa số nguyên ***n*** (1 ≤ ***n*** ≤ 106),
* Dòng thứ 2 chứa xâu độ dài ***n*** bao gồm các ký tự từ tập {***a***, ***b***} xác định dòng xếp hàng, ***a*** là nam, ***b*** – nữ.

**Output:** Một số nguyên – số cách lựa chọn.

**Ví dụ**

|  |  |
| --- | --- |
| **DANCE.INP** | **DANCE.OUT** |
| 8  abbababa | 13 |

**\* Ràng buộc:**

* 30% số test tương ứng với ***n*** ≤ 500
* 40% số test tiếp theo tương ứng với 500 < ***n*** ≤ 10000
* 30% số test cuối cùng tương ứng với 10000 < ***n*** ≤ 106

# Xâu con Tên file: SUBSTR.CPP hoặc SUBSTR.PAS

Một xâu gọi là xâu nhị phân nếu chỉ chứa hai ký tự "0" hoặc "1".

Xâu v gọi là xâu con của w nếu xâu v có độ dài khác 0 và gồm các ký tự liên tiếp trong xâu w. Ví dụ: xâu "010" có các xâu con là "0", "1", "0", "01", "10", "010".

**Yêu cầu:**Cho trước một giá trị k, hãy đếm xem có bao nhiêu xâu con chứa đúng k ký tự "1".

**Dữ liệu vào:** SUBSTR.INP

* Dòng 1: chứa một số nguyên k (0 ≤ k ≤ 106)
* Dòng 2: chứa một xâu nhị phân có độ dài ≤ 106

**Dữ liệu ra: SUBSTR.OUT**

* Ghi ra một số nguyên duy nhất là kết quả tìm được.

Ví dụ

|  |  |
| --- | --- |
| SUBSTR.INP | SUBSTR.OUT |
| 2  01010 | 4 |

***Giải thích:***có 4 xâu con chứa 2 ký tự 1 là: "101", "0101", "1010", "01010"

**Ràng buộc:** gọi ***n*** là độ dài xâu​

          - Có 30% số test ứng với 30% số điểm của bài có ***n*** ≤ 100.

          - Có 30% số test ứng với 30% số điểm của bài có 100 < ***n*** ≤ 1000.

          - Có 40% số test ứng với 40% số điểm của bài có 1000 < ***n*** ≤ 1.000.000.

# GIẢI MÃ CHỮ VIẾT CỦA NGƯỜI MAIA Tên file: WRITING.CPP hoặc WRITING.PAS

Công việc giải mã chữ viết của người MAIA là khó khăn hơn người ta tưởng nhiều. Trải qua hơn 200 năm mà người ta vẫn hiểu rất ít về các chữ viết này. Chỉ trong 3 thập niên gần đây do công nghệ phát triển việc giải mã này mới có nhiều tiến bộ.

Chữ viết Maia dựa trên các ký hiệu nhỏ gọi là nét vẽ, mỗi nét vẽ tương ứng với một âm giọng nói. Mỗi từ trong chữ viết Maia sẽ bao gồm một tập hợp các nét vẽ như vậy kết hợp lại với nhiều kiểu dáng khác nhau. Mỗi nét vẽ có thể hiểu là một ký tự ta hiểu ngày nay.

Một trong những vấn đề lớn khi giải mã chữ Maia là thứ tự đọc các nét vẽ. Do người Maia trình bày các nét vẽ này không theo thứ tự phát âm, mà theo cách thể hiện của chúng. Do vậy nhiều khi đã biết hết các nét vẽ của một từ rồi nhưng vẫn không thể tìm ra được chính xác cách ghi và đọc của từ này.

Các nhà khảo cổ đang đi tìm kiếm một từ đặc biệt *W*. Họ đã biết rõ tất cả các nét vẽ của từ này nhưng vẫn chưa biết các cách viết ra của từ này. Vì họ biết có các thí sinh IOI'06 sẽ đến nên muốn sự trợ giúp của các sinh viên này. Họ sẽ đưa ra toàn bộ *g* nét vẽ của từ *W* và dãy *S* tất cả các nét vẽ có trong hang đá cổ. Bạn hãy giúp các nhà khảo cổ tính xem có bao nhiêu khả năng xuất hiện từ *W* trong hang đá.

**Yêu cầu:** Hãy viết chương trình, cho trước các ký tự của từ *W* và dãy *S* các nét vẽ trong hang đá, tính tổng số khả năng xuất hiện của từ *W* trong dãy *S*, nghĩa là số lần xuất hiện một hoán vị các ký tự của dãy *g* ký tự trong *S*.

Các ràng buộc

* *1 ≤ g ≤ 3.000*, số nét vẽ trong *W*
* *g ≤ |S| ≤ 3.000.000*, *|S|* là số các nét vẽ của dãy *S*

**INPUT**: WRITING.INP:

* Dòng 1: chứa 2 số *g* và *|S|* cách nhau bởi dấu cách.
* Dòng 2: chứa *g* ký tự liền nhau là các nét vẽ của từ *W*. Các ký tư hợp lệ là 'a'-'z' và 'A'-'Z'. Các chữ in hoa và in thường là khác nhau.
* Dòng 3: Chứa *|S|* ký tự là dãy các nét vẽ tìm thấy trong hang. Các ký tư hợp lệ là 'a'-'z' và 'A'-'Z'. Các chữ in hoa và in thường là khác nhau.

**OUTPUT:** WRITING.OUT:

* Chứa đúng 1 số là khả năng xuất hiện của từ *W* trong dãy *S*.

Ví dụ:

|  |  |
| --- | --- |
| **WRITING.INP** | **WRITING.OUT** |
| 4 11  cAda  AbrAcadAbRa | 2 |