SPRING CONTEST 2021

Lưu ý: Các thí sinh có thể nộp bằng ngôn ngữ Pascal, C++ hoặc Java tùy theo ngôn ngữ nào thí sinh thấy thuận tiện. Tên các bài là Tenbai.*, trong đó * tương ứng với pas, cpp hay java. Tenbai là tên của bài sẽ được quy định trong cụ thể từng bài.

Bài 4 (SPC4.*):

Trong hóa học, "chất vô cơ" là một khái niệm rất quan trọng. Trong các chất vô cơ, những nguyên tố có thể cấu tạo nên một chất vô cơ rất nhiều, là Cacbon (C), Hidro (H), Sắt (Fe), Maggie (Mg), ... "Số ôxy hóa" của một nguyên tố trong phân tử là điện tích của nguyên tử trong nguyên tố đó.

Ta có một quy tắc như sau: Tổng các số ôxy hóa của các nguyên tử = 0.

Ví dụ:

- Xét chất HNO3 (axit nitric), số ôxy hóa của H là +1, của O là -2 và của N là +5. Ta thấy rằng tổng số ôxy hóa của các nguyên tố như sau:

  Phân tử HNO3, gồm 1H, 1N và 3O. Tổng số ôxy hóa là: 1 * 1 + 5 * 1 + (-2) * 3 = 0

- Xét chất KMnO4 (Kali Pemanganat), số ôxy hóa của K là +1, của O là -2 và của Mn là +7. Ta thấy tổng số ôxy hóa của các nguyên tố như sau:

  Phân tử KMnO4 gồm 1H, 1Mn và 4O. Tổng số oxy hóa là: 1 * 1 + 7 * 1 + (-2) * 4 = 0

- Phân tử Fe(NO3)3, số oxy hóa của Fe là +3, của N là +5 và O là -2. Ta thấy rằng tổng số oxy hóa của các nguyên tố như sau:

  Phân tử Fe(NO3)3 gồm 1Fe, 3N và 9O. Tổng số oxy hóa là: 3 * 1 + 5 * 3 + (-2) * 9 = 0.

Đạt sau khi thi học sinh giỏi quốc gia tin học, đã thực hiện lao đầu vào cày Toán - Lý - Hóa để thi đại học, nhưng hỡi ôi môn Hóa, Đạt không còn nhớ gì cả. Trong hợp chất đang xét, có K nguyên tố, Đạt đã tìm ra được số oxy hóa của K - 1 nguyên tố. Còn lại một nguyên tố cuối cùng, Đạt muốn tìm ra được số oxy hóa của nguyên tố đó.

Các bạn hãy giúp Đạt nhé, nếu Đạt không đậu đại học, Đạt sẽ ngủ ngoài đường mất !!!!!

- **Dữ liệu đầu vào (SPC4.INP):**
  - Đọc từ file SPC4.INP
  - Dòng đầu tiên gồm một số nguyên dương T ($1 \le T \le 50$) là số lượng bộ dữ liệu của đề bài.
  - T bộ dữ liệu tiếp theo, mỗi bộ dữ liệu được tổ chức như sau:
    - Dòng đầu tiên gồm 1 chuỗi ký tự, là hợp chất vô cơ tương ứng. Dữ liệu đảm bảo hợp chất vô cơ có từ 2 nguyên tố trở lên và đây là những chất vô cơ có thật trong thực tế. Dữ liệu sẽ chỉ chứa các ký tự chữ cái Latin in hoa, in thường, ký tự số và 2 ký tự '(' và ')' (nếu có). Những nguyên tố trong chất, sẽ được xác định bằng 1 chuỗi gồm 1 hoặc 2 ký tự. Nếu chuỗi chỉ có 1 ký tự, thì ký tự đó sẽ được viết hoa (Ví dụ: H, F, O, C,....). Nếu chuỗi có 2 ký tự, thì ký tự đầu tiên sẽ viết hoa và ký tự thứ 2 sẽ viết thường (Ví dụ: Mg, Mn, Fe, Ca, Ba, ...). Đứng sau mỗi nguyên tố sẽ là số

lượng nguyên tử của nguyên tố đó. Nếu chỉ có 1 nguyên tử thì ta không cần ghi số 1 đằng sau nguyên tố đó. Nếu như trong chuỗi có một cặp ngoặc '(' và ')', thì trong cặp ngoặc trên là một nhóm các nguyên tố tương ứng.

- Nếu chất có K nguyên tố, thì K - 1 dòng tiếp theo, mỗi dòng được tổ chức như sau: "nt: val", trong đó nt là nguyên tố và val là một số nguyên tương ứng là số oxy hóa của nguyên tố đó có trong chất. Dữ liệu bảo đảm rằng không có nguyên tố nào nằm ngoài các nguyên tố trong hợp chất kể trên, và các nguyên tố trong K - 1 dòng là khác nhau.

- **Dữ liệu đầu ra (SPC4.OUT):**
  - Ghi ra file SPC4.OUT
  - Gồm T dòng, mỗi dòng gồm một số nguyên duy nhất là số oxy hóa của nguyên tố còn lại trong K nguyên tố kể trên.
- **Ví dụ:**

SPC4.INP	SPC4.OUT
5 H2O O: -2 H2SO4 H: 1 S: 6 HCl Cl: -1 Fe(NO3)3 N: 5 O: -2 MgCl2 Mg: 2	1 -2 1 3 -1

- **Giải thích test ví dụ:**
  - Với ví dụ 1: Một chất rất quen thuộc là nước. Ta biết được số oxy hóa của Oxy là -2, do đó số oxy hóa của Hidro phải là 1.
  - Với ví dụ 2: Axit sunfuric (H2SO4) là một hợp chất khá nguy hiểm. Ta biết được số oxy hóa của H là +1 và S là +6, như vậy Oxy sẽ là -2
  - Những ví dụ còn lại, bạn đọc có thể tự kiểm chứng.
- **Bài toán sẽ chỉ có 1 Dataset duy nhất:**
  - Độ dài các chất hóa học sẽ không quá 15 ký tự, và số lượng nguyên tố không vượt quá 3.
- **Giới hạn thời gian và bộ nhớ:**
  - 2s / Dataset.
  - 512Mb / Dataset.

Bài 5 (SPC5.*):

😣😣😣, chúng ta chuẩn bị bước qua tháng 1, là 1 tháng có tiết trời rất mát mẻ, dễ chịu, vui tươi 😣😣, chính vì vậy mà chúng ta có Spring Contest 2021. Nhưng vẫn luôn có một người rất hay buồn 😣, là Tú, cậu ta luôn buồn 😣 trong mọi lúc mọi nơi, trong mọi comment 😣😣😣

đến mọi status 😣 😣. Vì vậy, hãy giúp Tú toại nguyện được sống trong những nỗi buồn nhé 😣 😣.

Một mặt buồn 😣 được tạo ra bằng việc ghép 1 chữ T ('T') đứng đầu, 1 hoặc nhiều dấu gạch dưới ('_') và kết thúc bằng một chữ T ('T'). Ví dụ một số mặt buồn như sau : T_T , T___T, T_____T , T_____T , T_____T

Bạn được cho một chuỗi s gồm các ký tự 'T' và '_'. Một mặt buồn được tạo thành từ chuỗi s sẽ được định nghĩa như sau:

- Chuỗi tạo nên mặt buồn là một chuỗi con KHÔNG LIÊN TỤC của chuỗi s
- Các ký tự tạo nên phải đúng theo quy tắc của mặt buồn.

Một chuỗi con KHÔNG LIÊN TỤC của chuỗi s là một chuỗi b được tạo nên từ s bằng việc xóa 0, 1 hoặc nhiều ký tự trong chuỗi s.

Ví dụ: Chuỗi s = "aebcde" thì một số chuỗi b là chuỗi con KHÔNG LIÊN TỤC của chuỗi s là các chuỗi "ae", "ade", "bd", ...

Chuỗi "ae" là chuỗi con của chuỗi s vì "ae" được tạo nên bằng việc xóa các ký tự 'b', 'c', 'd', 'e' khỏi chuỗi s.

Chuỗi "abd" là chuỗi con của chuỗi s vì "abd" được tạo nên bằng việc xóa các ký tự 'e', 'c', 'e' khỏi chuỗi s.

Chuỗi "acb" không phải là chuỗi con của chuỗi s vì ta không thể tạo ra được chuỗi "acb" bằng việc xóa những ký tự trong chuỗi s.

Bạn được cho chuỗi s, hãy giúp Tú sống thỏa thích trong nỗi buồn bằng việc đếm có bao nhiêu cách chia chuỗi s thành những chuỗi sao cho mỗi chuỗi đều là mặt buồn hợp lệ nhé !!! Vì số cách có thể rất lớn, bạn cần in ra phần dư của số cách khi chia cho 1000000007. Biết đâu khi giải được bài này, Tú sẽ hết buồn 😣😣😣😣.

- **Dữ liệu đầu vào (SPC5.INP):**
  - Đọc từ file SPC5.INP.
  - Dòng đầu tiên gồm một số nguyên dương T ($1 \le T \le 40$) là số lượng bộ dữ liệu cần xử lý
  - T bộ dữ liệu tiếp theo, mỗi bộ dữ liệu chỉ chứa 1 dòng duy nhất là chuỗi s. Chuỗi s đảm bảo rằng chỉ gồm 2 ký tự là 'T' và '_'. Đồng thời bảo đảm rằng số lượng ký tự 'T' là số chẵn.
- **Dữ liệu dầu ra (SPC5.OUT):**
  - Ghi vào file SPC5.OUT.
  - Kết quả trả về cho T bộ test, mỗi bộ test gồm 1 số duy nhất là phần dư của số cách chia chuỗi thành những mặt buồn hợp lệ cho 1000000007
- **Ví dụ:**

SPC5.INP	SPC5.OUT

5 T_____T T_TT_____T _T_____T TTT___TTT T_TT__T_TT	1 2 0 36 52

- **Giải thích test ví dụ:**
  - Ví dụ 1: Chỉ có đúng 1 cách duy nhất để chia mặt buồn đó là chọn toàn bộ ký tự trong cả dãy.
  - Ví dụ 2: Có 2 cách chia như sau:
    - Cách 1:

      T_T

      T_____T
    - Cách 2:

      T_ T

      T _____T
  - Ví dụ 3: Vì ký tự đầu tiên của chuỗi là '_' nên không thể nào tạo được một mặt buồn nào thỏa mãn.
  - Ví dụ 4: Có 3 ký tự đầu là 'T', 3 ký tự giữa là '_' và 3 ký tự cuối là 'T'. Dù ta chọn bất kỳ chữ T nào trong tập 3 chữ T đầu, ký tự '_' nào trong 3 ký tự giữa và 'T' nào trong 3 ký tự cuối, ta cũng sẽ được 1 mặt cười.

    Do đó, giả sử mặt cười thứ 1 có chữ T thứ 1. Số cách chọn một ký tự '_' trong tập giữa là 3, số cách chọn ký tự 'T' trong tập cuối là 3. Ta có tổng cộng 3 * 3 = 9 cách tạo mặt cười thứ 1.

    Giả sử mặt cười thứ 2 có chữ T thứ 2. Số cách chọn một ký tự '_' trong tập giữa còn lại là 2, số cách chọn chữ 'T' trong tập 2 chữ 'T' cuối còn lại là 2. Ta được 2 * 2 = 4 cách tạo ra mặt cười thứ 2. Các ký tự còn lại tạo nên mặt cười thứ 3. Số cách để tách thành 3 mặt cười là 9 * 4 = 36 cách.
  - Ví dụ 5: Nhằm mục đích để bạn đọc có thể tự kiểm tra với đáp án của mình.
- **Bài toán có 2 Dataset:**
  - Small Dataset: Chuỗi s chỉ có thể tách được tối đa 2 mặt buồn.
  - Large Dataset: Không có điều kiện về số lượng mặt buồn.
  - $1 \leq len(s) \leq 50$, len(s) là độ dài của chuỗi s. Dữ liệu luôn bảo đảm số lượng ký tự 'T' là số chẵn.
- **Giới hạn dữ liệu:**
  - 5s / Dataset.
  - 512Mb / Dataset.

Bài 6 (SPC6.*):

Lập trình Robot là một bộ môn tạo được rất nhiều hứng thú cho các bạn trẻ đam mê tin học nói riêng và đam mê khoa học nói chung. Trong một cuộc thi Robot, 2 chú Robot được đặt vào đấu trường, tại 2 địa điểm bất kì và cần di chuyển đến đích. Chú Robot A cần đến đích dành cho chú A và chú Robot B cần đến đích dành riêng cho chú B. Bản đồ của đấu trường được tổ chức như một ma trận kích thước N * M, gồm N dòng và M cột. Các dòng được đánh số từ 1 đến N và các cột được đánh số theo thứ tự từ 1 đến M. Trên đấu trường có những ô có thể đi được và có những ô không thể đến được. Một chú Robot đang đứng tại ô (x, y), có thể di chuyển qua những ô kề cạnh, là 'Lên', 'Xuống', 'Trái' hoặc 'Phải'. Bạn cần lập trình cho cả 2 chú Robot sao cho 2 chú đến được đích cùng một lúc và trong quá trình di chuyển, không có một thời điểm nào 2 chú Robot đứng cùng một ô. Biết rằng 2 chú Robot không thể đi qua nhau (Xem test ví dụ số 4 sẽ rõ).

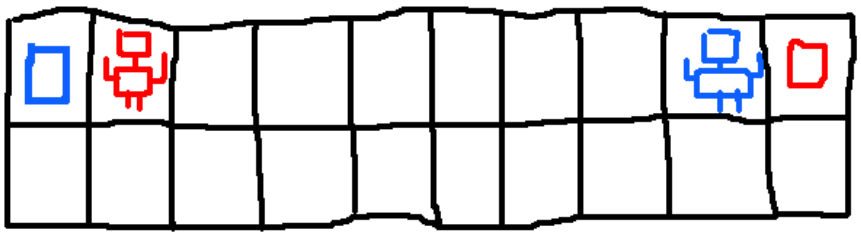
- **Dữ liệu đầu vào (SPC6.INP):**
  - Dòng đầu tiên gồm một số nguyên dương T ($1 \le T \le 40$), là số lượng bộ dữ liệu.
  - T bộ dữ liệu tiếp theo, mỗi bộ dữ liệu được tổ chức dưới dạng như sau:
    - Dòng đầu tiên gồm hai số nguyên dương N và M, tương ứng là số lượng hàng và cột của đấu trường.
    - N dòng tiếp theo, mỗi dòng gồm M ký tự. Các ký tự có thể là 'A', 'a', 'B', 'b', '.' hoặc '#'. Trong đó:
      - Nếu ký tự là 'A', thì ở vị trí tương ứng là vị trí xuất phát của Robot A. Dữ liệu đảm bảo chỉ có duy nhất 1 ký tự 'A' trong bảng.
      - Nếu ký tự là 'B', thì ở vị trí tương ứng là vị trí xuất phát của Robot B. Dữ liệu đảm bảo chỉ có duy nhất 1 ký tự 'B' trong bảng.
      - Nếu ký tự là 'a', thì ở vị trí tương ứng là vị trí cần đến của Robot A. Dữ liệu đảm bảo chỉ có duy nhất 1 ký tự 'a' trong bảng.
      - Nếu ký tự là 'b', thì ở vị trí tương ứng là vị trí xuất phát của Robot B. Dữ liệu đảm bảo chỉ có duy nhất 1 ký tự 'b' trong bảng.
      - Nếu ký tự là '.', đó là vị trí mà 2 chú Robot có thể đến được.
      - Nếu ký tự là '#', đó là vị trí mà Robot không thể đến.
- **Dữ liệu đầu ra (SPC6.OUT):**
  - Kết quả cho T bộ dữ liệu. Nếu ở dữ liệu thứ i, không tồn tại cách để 2 chú Robot có thể đến được đích trong cùng 1 thời điểm, in ra 1 dòng NO SOLUTION. Nếu tồn tại cách, ta in ra một số nguyên dương là thời điểm ít nhất mà cả 2 chú Robot cùng đến đích.

Ví dụ:

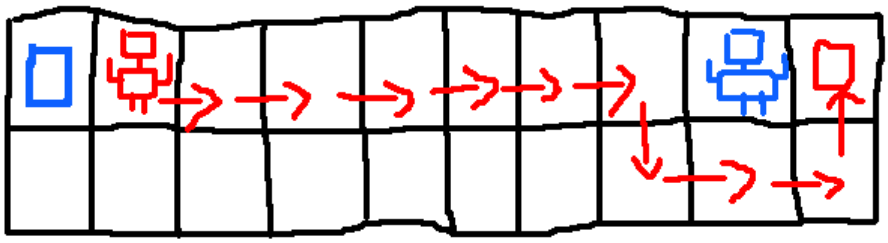
SPC6.INP	SPC6.OUT
5 1 4 ABab 5 3 ... #a# A.b #B# #.# 4 11 AB.....	2 4 NO SOLUTION NO SOLUTION 10

....b.a 1 10 bA.....Ba 2 10 bA.....Ba	

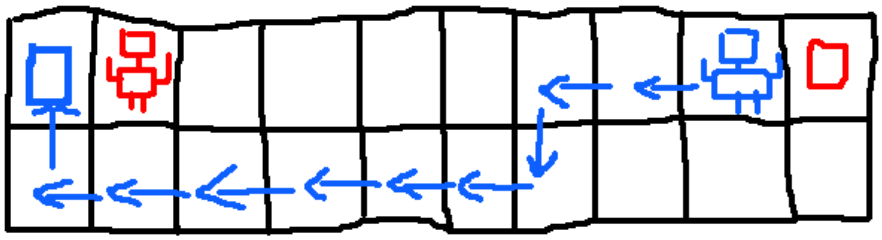
- **Giải thích test ví dụ:**
  - Với test ví dụ 1: Cả chú Robot A và B cùng tiến qua phải 2 bước thì sẽ đến được đích.
  - Với test ví dụ 2: Chú A sẽ đi như sau: Phải - lên - lên - xuống, Chú B sẽ đi như sau: xuống - lên - lên - phải. Lưu ý: Kết quả không thể là 2 được, vì nếu kết quả là 2, chú A sẽ phải đi như sau: phải - lên, chú B sẽ đi như sau: lên - phải. Tại thời điểm thứ 1, cả chú A và chú B sẽ cùng đứng tại một ô là ô (3, 2). Điều này vi phạm luật.
  - Với test ví dụ 3: Tuy đường rộng thênh thang, không có chướng ngại vật nhưng cả 2 chú A và B sẽ không bao giờ đến được đích cùng một lúc cả.
  - Với test ví dụ 4: Chú robot A và B không được phép đi qua nhau, tại thời điểm số 3, chú A đang đứng ở ô (1, 5), chú B đang đứng ở ô (1, 6). Và tại thời điểm sau đó, nếu chú A vẫn muốn tiến qua phải, chú A sẽ đứng ở ô (1, 6), còn chú B muốn tiến qua trái, chú B sẽ đứng ở ô (1, 5) và điều này không được phép xảy ra.
  - Với test ví dụ 5: Ban đầu, trạng thái của các chú Robot sẽ như sau:



Hành trình di chuyển của chú Robot A (màu đỏ)



Hành trình di chuyển của chú Robot B (màu xanh)



- **Bài toán sẽ được chia làm 2 Dataset:**
  - Small Dataset: Không tồn tại một ký tự '#' nào trong bảng, đồng thời vị trí xuất phát của Robot A và Robot B lần lượt ở 2 vị trí (1, 1) và (1, 2). Còn vị trí của 2 đích của Robot a và Robot b lần lượt là (N, M - 1) và (N, M).
  - Large Dataset: Không ràng buộc gì thêm.
  - Trong mọi Dataset, ta luôn có N và M không vượt quá 40.
- **Giới hạn dữ liệu:**
  - 10s / Dataset.
  - 512Mb / Dataset.

----------Chúc các bạn làm bài tốt <3 <3 <3 ----------