**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

**ĐÈ THI CẤP CHỨNG CHỈ LỚP THUẬT TOÁN ỨNG DỤNG SAMSUNG NĂM 2025**

**Thời gian thi: 150** phút (08h15p – 10h45p)

**Ngày thi:** Chủ nhật, ngày 14 tháng 09 năm 2025.

**Lưu ý:**

* Đề thi có 06 trang. 06 bài, mỗi bài 100 điểm.
* Điểm bài thi của sinh viên được tính bằng tổng điểm của 6 bài tập
* Với mỗi bài tập, sinh viên nhận được điểm tương ứng dựa trên số test làm đúng

Ví dụ, bài D, sinh viên AC 4/10 test thì sẽ nhận được 4/10 \* 100 = 40 điểm

* Link Contest tái hiện sau giờ thi: <https://codeforces.com/group/TNf7raZKBu/contest/635423>

**ĐỀ BÀI**

**BÀI A. MA TRẬN 3x3**

Hôm nay trời đẹp, An và Bình nghĩ ra một trò chơi vui để vừa luyện trí thông minh vừa giải trí sau Vòng loại ICPC. Trên bàn có hai tờ giấy:

1. Một tờ đã được kẻ sẵn bảng 3×3.
2. Một tờ khác, An viết ra 6 số nguyên: A, B, C, D, E và F.

Ngoài ra, còn có một cây bút chì và một cục tẩy để có thể điền vào bảng nhiều lần.

An đố Bình: hãy điền các số vào bảng 3×3 sao cho:

* Tổng các số trên hàng 1, 2, 3 lần lượt là A, B và C
* Tổng các số trên cột 1, 2, 3 lần lượt là D, E và F
* Các ô có thể điền bất kỳ số nguyên dương nào vào bảng cũng được

Bình thấy việc liệt kê tất cả các trường hợp quá khó. Bạn hãy giúp Bình tìm số cách điền bảng 3×3 thỏa mãn các điều kiện trên.

**Input.** Một dòng duy nhất gồm 6 số nguyên A, B, C, D, E và F. Mỗi số cách nhau 1 dấu cách

**Output.** In ra số cách điền thoả mãn vào bảng 3x3. Nếu không có cách điền nào thì ghi 0.

**Subtask.**

* **Subtask 1 (30%):** 1 ≤ A, B, C, D, E, F ≤ 10
* **Subtask 2 (70%):** 1 ≤ A, B, C, D, E, F ≤ 30

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 3 3 3 3 3 3 | 1 |
| 3 3 3 2 4 3 | 0 |
| 5 13 10 6 13 9 | 120 |

**Giải thích test:**

* **Test 1:** Ma trận cần tìm gồm 9 số 1
* **Test 2:** Mỗi cột là tổng của 3 số dương, như vậy tổng tối thiểu mỗi cột là 3. Nhưng đề bài yêu cầu tổng cột 1 là 2 🡪 Không tồn tại

**BÀI B. XÂU KHÔNG CÂN BẰNG**

Cho một xâu S và một số K. Biết rằng xâu S chỉ gồm 3 loại ký tự: Dấu hỏi chấm (?), A và B. Hãy đếm số cách thay các dấu ? bằng 1 trong 2 ký tự A và B, sao cho xâu thu được không có xâu con liên tiếp đối xứng nào độ dài K. Vì đáp án có thể rất lớn, hãy chia dư kết quả tính được cho 109 + 7.

**Input.**

* Dòng đầu tiên gồm 2 số N và K. (2 ≤ K ≤ 10)
* Dòng thứ hai là N ký tự của xâu S, là 1 trong 3 ký tự: A, B, ?

**Output.** In ra số cách thay thế các dấu ? theo modulo 109 + 7

**Subtask**

* **Subtask 1 (40%):** N≤ 20
* **Subtask 2 (60%):** N≤ 1000

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 6 3  AB?A?B | 1 |
| 6 3  ABA??? | 0 |
| 10 6  ??A??????? | 254 |

**Giải thích test:**

* **Test 1:**
* AB**A**A**A**B. Xét **ABA**AAB: S[1 .. 3] = ABA, đối xứng độ dài 3, loại
* AB**A**A**B**B. Xét **ABA**ABB: S[1 .. 3] = ABA, đối xứng độ dài 3, loại
* AB**B**A**A**B. Các xâu con liên tiếp độ dài 3: ABB, BBA, BAA, AAB đều không đối xứng
* AB**B**A**B**B. Xét AB**BAB**B. S[3 .. 5] = BAB, đối xứng độ dài 3, loại

Vậy chỉ có 1 cách thay thỏa mãn

* **Test 2:**

**ABA**???. S[1 … 3] = ABA, đối xứng độ dài 3

Đây là phần không thể thay đổi, vi phạm vào điều kiện không tồn tại xâu đối xứng độ dài 3, bất kể cách thay là gì. Do đó, không có cách thay nào

**BÀI C. DI CHUYỂN TRONG MÊ CUNG - 1**

Một robot đang di chuyển trong một mê cung N hàng × M cột. Mỗi ô trong mê cung là ô trống, robot có thể đi vào, hoặc là chướng ngại vật, robot không thể bước qua. Robot có một điểm xuất phát và một đích đến. Mỗi bước, robot có thể di chuyển lên, xuống, trái hoặc phải một ô liền kề, miễn là không ra ngoài mê cung và không đi vào ô chướng ngại vật. Robot có thể bị kẹt nếu tất cả các đường đi bị chặn. Nhiệm vụ của bạn là tìm số bước ít nhất để robot từ xuất phát đến đích. Nếu robot không đi được do moi đường đi bị chặn, in -1.

**Input:**

* Dòng đầu tiên gồm hai số nguyên N và M là số hàng và số cột của mê cung
* Dòng thứ hai là hai cặp tọa độ (x, y) và (z, t) là xuất phát và đích đến
* N dòng tiếp theo: mỗi dòng gồm M ký tự, thuộc 1 trong 2 loại:
  + Dấu . : Biểu thị ô tự do
  + Dấu #: Biểu thị ô có vật cản
* 1 ≤ N, M ≤ 1000; 1 ≤ x < z ≤ N; 1 ≤ y < t ≤ M

**Output:**

Một số nguyên duy nhất: số bước ít nhất từ xuất phát đến đích, hoặc -1 nếu robot không có đường nào để đi

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 4 4  1 1 4 4  ..#.  ...#  .#..  .#.. | 6 |
| 3 3  1 1 3 2  .#.  ###  ..# | -1 |

**Giải thích test**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test 1:**   |  |  |  |  | | --- | --- | --- | --- | | **ST** |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  | **EN** |   Đường đi ngắn nhất: (1, 1) 🡪 (1, 2) 🡪 (2, 2) 🡪 (2, 3) 🡪 (3, 3) 🡪 (3, 4) 🡪 (4, 4)  Tổng cộng cần 6 bước đi | **Test 2:**   |  |  |  | | --- | --- | --- | | **ST** |  |  | |  |  |  | |  | **EN** |  |   Dễ thấy rằng từ ô xuất phát, cả 4 hướng đều đã bị chặn nên không đi được ngay từ điểm xuất phát. Vậy đáp án là -1 |

**BÀI D. TRUY VẤN VỚI DÃY NGOẶC ĐÚNG**

Dãy ngoặc đúng được định nghĩa như sau:

* Một xâu rỗng được coi là một dãy ngoặc đúng.
* Nếu A là dãy ngoặc đúng thì (A) cũng là một dãy ngoặc đúng.
* Nếu A và B là dãy ngoặc đúng thì xâu AB cũng là một dãy ngoặc đúng.

Cho xâu S dài N ký tự, chỉ có 1 trong 2 loại ký tự: ) và (. Thực hiện 2 loại truy vấn:

* **Loại 1: 1 x y:** Hoán vị S[x] thành S[y] và ngược lại
* **Loại 2:** **2 x y:** Hỏi đoạn con S[x] S[x + 1] … S[y] có là dãy ngoặc đúng hay không

**Input**

* Dòng đầu tiên gồm 2 số N và Q, lần lượt là độ dài xâu S và số truy vấn
* Dòng thứ hai là N ký tự của xâu S, là 1 trong 2 loại ) hoặc (
* Q dòng tiếp theo, mỗi dòng gồm 3 số t, x và y. Trong đó:
  + t là loại truy vấn, là 1 hoặc 2
  + x và y là vị trí trong xâu, tính từ 1

**Output**. Với mỗi truy vấn loại 2, hãy in kết quả kiểm tra trên 1 dòng. Nếu xâu con từ x đến y là đối xứng, hãy in ra YES, ngược lại, in ra NO

**Subtask**

* **Subtask 1 (40%):** N, Q ≤ 103
* **Subtask 2 (60%):** N, Q ≤ 2.105

**Ví dụ.**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 5 6  (())(  2 1 4  2 1 2  1 1 4  2 1 4  1 1 5  2 2 5 | YES  NO  NO  YES |

**Giải thích test:**

* **Ban đầu:** (())(
* **Truy vấn 1:** S[1 … 4]: **(())**(. Dãy (()) là dãy ngoặc đúng
* **Truy vấn 2:** S[1 … 2]: **((**))(. Dãy (( là dãy ngoặc sai
* **Truy vấn 3:** Hoán vị S[1] và S[4]: **(**()**)**( 🡪 **)**()**(**(
* **Truy vấn 4:** S[1 … 4]: **)()(**(. Dãy )()( là dãy ngoặc sai
* **Truy vấn 5:** Hoán vị S[1] và S[5]: **)**()(**(** 🡪 **(**()(**)**
* **Truy vấn 6:** S[2 … 5]: (**()()**. Dãy ()() là dãy ngoặc đúng

**BÀI E. THAO TÁC XOÁ XÂU**

Cho xâu S có N ký tự. Mỗi lần, bạn được phép chọn 1 xâu con liên tiếp S[l] S[l + 1] … S[r] gồm các ký tự y hệt nhau rồi xoá chúng. Hỏi sau bao nhiêu lần thì ta sẽ xoá hết sạch xâu?

**Input.**

* Dòng đầu tiên là số N
* Dòng thứ hai là N ký tự của xâu S

**Output.** Số bước ít nhất để xoá hết N ký tự trong xâu S

**Subtask**

* **Subtask 1 (40%):** N ≤ 20
* **Subtask 2 (60%):** N ≤ 500

**Ví dụ.**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 8  abcddcba | 4 |
| 5  aeaea | 3 |

**Giải thích test:**

* **Test 1:**

Bước 1: abc**dd**cba: Xóa đoạn [4; 5] đi, còn lại abccba

Bước 2: ab**cc**ba: Xóa đoạn [3; 4] đi, còn lại abba

Bước 3: a**bb**a: Xóa đoạn [2; 3] đi, còn lại aa

Bước 4: **aa**: Cả xâu giống nhau, nên xóa đi ta được xâu rỗng

* **Test 2:**

Bước 1: ae**a**ea: Xóa S[3] đi, còn lại aeea

Bước 2: a**ee**a: Xóa đoạn [2; 3] đi, còn lại aa

Bước 3: **aa**: Cả xâu giống nhau, nên xóa đi ta được xâu rỗng

**BÀI F. DI CHUYỂN TRONG MÊ CUNG - 2**

Một robot đang di chuyển trong một mê cung N hàng × M cột. Mỗi ô trong mê cung là ô trống, robot có thể đi vào, hoặc là chướng ngại vật, robot không thể bước qua. Robot có một điểm xuất phát và một đích đến. Luật di chuyển được quy định như sau:

* Mỗi bước, robot có thể di chuyển lên, xuống, trái hoặc phải một ô liền kề
* Xét hình vuông cạnh 5x5 mà ô (i, j) làm tâm. Robot được quyền đến bất kỳ ô nào trong phạm vi hình vuông này với số bước là 2.
* Không ra ngoài mê cung và không đi vào ô chướng ngại vật.

Nhiệm vụ của bạn là tìm số bước ít nhất để robot từ xuất phát đến đích. Nếu robot không đi được do mọi đường đi bị chặn, in -1.

**Input:**

* Dòng đầu tiên gồm hai số nguyên N và M là số hàng và số cột của mê cung
* Dòng thứ hai là hai cặp tọa độ (x, y) và (z, t) là xuất phát và đích đến
* N dòng tiếp theo: mỗi dòng gồm M ký tự, thuộc 1 trong 2 loại:
  + Dấu . : Biểu thị ô tự do
  + Dấu #: Biểu thị ô có vật cản
* 1 ≤ N, M ≤ 1000; 1 ≤ x < z ≤ N; 1 ≤ y < t ≤ M

**Output:**

Một số nguyên duy nhất: số bước ít nhất từ xuất phát đến đích, hoặc -1 nếu robot không có đường nào để đi

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 4 4  1 1 4 4  ..#.  ...#  .#..  .#.. | 4 |
| 3 3  1 1 3 2  .#.  ###  ..# | -1 |

**Giải thích test**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test 1:**   |  |  |  |  | | --- | --- | --- | --- | | **ST** |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  | **EN** |   1 trong số các đường đi ngắn nhất là : (1, 1) 🡪 (2, 1) (1 bước) 🡪 (4, 3) (2 bước) 🡪 (4, 4) (1 bước)  Tổng: 1 + 2 + 1 = 4 bước | **Test 2:**   |  |  |  | | --- | --- | --- | | **ST** |  |  | |  |  |  | |  | **EN** |  |   Dễ thấy rằng từ ô xuất phát, cả 4 hướng đều đã bị chặn nên không đi được ngay từ điểm xuất phát. Vậy đáp án là -1 |

**HẾT**