**BÀI 91. TRAKTOR**

Mirko mới sắm được một chiếc máy thu hoạch cực “cool”, nó có thể hái nấm giúp cho Mirko trong dịp giáng sinh này. Nấm của mọc trên một thảo nguyên rộng lớn hình vuông, có thể đặt lên một mặt phẳng tọa độ với điểm trái dưới là (1, 1) và phải trên (105, 105).

Có **N** cây nấm mọc dần dần theo từng giây, mỗi giây, một cây nấm sẽ mọc lên trên một ô trống.

Mirko muốn để chiếc máy thu hoạch đúng một lần duy nhất với **K** cây nấm. Mirko có thể đặt chiếc máy ở một ô bất kì. Nó có thể di chuyển theo các hướng song song với trục tọa độ hoặc theo hướng chéo của các ô. Vì đây là một chiếc máy có tốc độ rất tốt nên trên đường đi nó không thể rẽ.

Hãy giúp Mirko tính toán xem thời điểm mà anh ấy có thể thu hoạch được đúng **K** cây nấm. Cây nấm đầu tiên mọc tại thời điểm 1.

**INPUT:** Nhập vào từ file **INP.TXT**

* Dòng thứ nhất là hai số nguyên **N**, **K** (2 ≤ **K** ≤ **N** ≤ 106).
* **N** dòng tiếp theo, mỗi dòng chứa hai số nguyên **Xi** , **Yi** là tọa độ mà cây nấm thứ .

**OUTPUT:** Đưa ra file **TRAKTOR.OUT** một số duy nhất là thời điểm mà Mirko thu được đủ **K** cây nấm. Nếu Mirko không thể thu đủ K cây nấm, in ra **-1**.

**SAMPLE TEST:**

|  |  |  |
| --- | --- | --- |
| **INPUT**  4 3  1 2  3 4  3 2  4 5  **OUTPUT**  4 | **INPUT**  7 4  3 1  2 2  4 1  3 2  2 3  1 4  1 3  **OUTPUT**  6 | **INPUT**  5 2  1 1  2 1  1 2  1 3  1 4  **OUTPUT**  2 |

**BÀI 92. PAPRIKA**

Marin là một trợ lý bếp của nhà hàng **Plavi 9**. Công việc hàng ngày của anh là chuẩn bị nguyên liệu cho bếp trưởng. Hôm nay, anh phải chuẩn bị một nguyên liệu khá đơn giản, đó là hạt tiêu. Mỗi đầu bếp đều biết rằng, hạt tiêu được chọn để cho vào chế biến phải là hạt con non. Vì vậy anh quyết định chọn hạt không già hơn **X** ngày. Mỗi hạt tiêu được dán một ID card, từ đó, Marin dễ dàng biết được tuổi của hạt đó.

Những hạt tiêu không hề biết về số **X** của Marin, chúng quyết định đổi các ID card cho nhau theo thứ tự sau: hạt thứ nhất sẽ đổi ID cho hạt thứ hai, hạt thứ hai sẽ đổi ID cho hạt thứ ba, … hạt thứ **N – 1** sẽ đổi ID cho hạt thứ **N**. Hai hạt chỉ đổi ID cho nhau khi hạt có ID lớn hơn có mong muốn được dùng làm nguyên liệu, nhưng với một hạt tương tự có ID nhỏ hơn thì không đổi. Hai hạt có cùng ID thì không đổi ID cho nhau (đổi cũng như không =)) ).

**INPUT:** Nhập vòa từ file **INP.TXT**

* Dòng thứ nhất là hai số nguyên **N** và **X** (1 ≤ **X**, **N** ≤ 1000) là số hạt tiêu hiện có và mốc ngày mà Marin đã chọn.
* **N** dòng tiếp theo, dòng thứ i chứa hai số nguyên **ai** và **bi** (1 ≤ **ai** ≤ 1000, **bi** bằng **1** hoặc **0**) là ID của hạt tiêu thứ **i** và mong muốn của nó (**0** nếu nó mong được bảo quản và **1** nếu nó mong được dùng làm nguyên liệu).

**OUTPUT:** Đưa ra file **OUT.TXT** một số nguyên duy nhất là số lượng hạt tiêu thực hiên được ước muốn của nó.

**SAMPLE TEST:**

|  |  |  |
| --- | --- | --- |
| **INPUT**  4 5  2 0  3 0  4 0  5 0  **OUTPUT**  0 | **INPUT**  5 5  3 1  2 0  13 1  2 0  10 1  **OUTPUT**  5 | **INPUT**  6 10  15 1  12 1  8 0  10 1  3 0  1 1  **OUTPUT**  4 |

**BÀI 93. NIKO**

Một đội hình của một đội bóng được xác định bởi vị trí chơi của các cầu thủ. Đội hình được biểu diễn bởi 3 số nguyên **O**, **V** và **N**. **O** là số lượng cầu thủ ở vị trí phòng ngự, **V** là số lượng cầu thủ chơi ở vị trí giữa sân và **N** là số lượng cầu thủ chơi ở vị trí tấn công. Nó phải đảm bảo **O** + **V** + **N** = 10.

Sau khi chọn đội hình, huấn luyện viên phải chọn cầu thủ thích hợp vào vị trí đó bởi không phải cầu thủ nào cũng chơi được ở mọi vị trí. Huấn luyện viên biết rất rõ về các cầu thủ của mình. Ông ấy có một danh sách các cầu thủ. Hãy giúp ông ấy sắp xếp xem các đội hình mà ông ấy đưa ra có thực hiên được không.

**INPUT:** Nhập vào từ file **INP.TXT**

* Dòng thứ nhất là số nguyên **n** (1 ≤ **n** ≤ 10) là số đội hình mà huấn luyện viên muốn nhờ bạn kiểm tra.
* **n** dòng tiếp, mỗi dòng là một đội hình theo mẫu **O**-**V**-**N**, trong đó **O**, **V**, **N** là các số nguyên thỏa mãn **O** + **V** + **N** = 10.
* Dòng tiếp theo là số nguyên **m** (10 ≤ **m** ≤ 22) là số lượng cầu thủ muốn vào đội.  
  **m** dòng kế tiếp, dòng thứ i là các xâu chứa các kí tự **O**, **V**, **N** là các vị trí chơi mà cầu thủ thứ i có thể chơi.

**OUTPUT:** Đưa ra file **OUT.TXT** N dòng, mỗi dòng là kết quả: “**NE**” nếu không thể xếp đội hình, “**DA**” nếu có thể xếp đội hình.

**SAMPLE TEST:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **INP.TXT** | **OUT.TXT** |  | **INP.TXT** | **OUT.TXT** |
| 2  4-4-2  10-0-0  10  O  O  O  O  O  O  O  O  ON  NO | NE DA |  | 3  4-4-2  3-5-2  4-3-3  11  OV  OV  OVN  OV  OV  V  V  N  O  O  O | DA  DA  NE |

**BÀI 94. METEOR**

Một nhà nhiếp ảnh chuyển nghiệp đã chụp được bức ảnh khi thiên thạch đang ở trên cao xuống rơi xuống và một bức khi nó vừa mới chạm đất (chưa bị vỡ). Sau chụp, anh ta kiểm tra lại thì thấy hỏng một tấm - tấm ảnh khi thiên thạch chạm đất.

Nhiếp ảnh gia quyết định “Photoshop” lại tấm hình đã mất từ tấm còn lại. Tấm hình còn lại rất đơn giản, là một lưới ô vuông, gồm các kí tự:

* ‘**X**’ biểu diễn phần thiên thạch.
* ‘**#**’ biểu diễn phần mặt đất.
* ‘**.**’ biểu diễn phần không khí.

Vì nhiếp ảnh gia vẫn còn nghiệp dư, nên bạn hãy giúp anh ấy khôi phục tấm hình mà anh ấy đã làm mất nhé.

**INPUT:** Nhập vào từ file **INP.TXT**

Dòng thứ nhất là hai số nguyên **R**, **S** là kích thước bức ảnh (1 ≤ **R**, **S** ≤ 3000).  
**R** dòng kế tiếp, mỗi dòng là **S** kí tự (‘**X**’, ‘**#**’ hoặc ‘**.**’) là các kí tự mô tả nội dung bức ảnh. Đảm bảo rằng thiên thạch luôn nằm ở trên. Mặt đất và Thiên thạch là các phần tử liền nhau (hai kí tự luôn kề trái, phải, trên hoặc dưới nhau). Luôn tồn tại ít nhất **một** hàng chỉ có kí tự ‘**.**’ .

|  |  |
| --- | --- |
| **INPUT**  5 6  .XXXX.  ...X..  ......  #..###  ######  **OUTPUT**  ......  .XXXX.  ...X..  #..###  ###### | **INPUT**  9 7  XXX.XXX  X.XXX.X  X..X..X  X.....X  .......  .#...#.  .##.##.  .#####.  #######    **OUTPUT**  .......  .......  .......  .......  XXX.XXX  X#XXX#X  X##X##X  X#####X  ####### |

**OUTPUT:** Đưa ra file **OUT.TXT**

Bức ảnh mà nhiếp ảnh mong muốn.

**Sample test:**

**BÀI 95. TETA**

Teta được rất nhiều học sinh trong trường yêu quý. Cô ấy làm ở quầy tính tiền trong canteen. Một trong những lý do Teta rất được yêu quý là cách tính tiền của cô ấy luôn tiết kiệm túi tiền cho học sinh khi mua đồ ăn.

Cách tính tiền của cô ấy như thế nào? Rất đơn giản. Tất cả các món ăn trong canteen đều được mua với giá đã biết. Mỗi ngày có một menu với **4** món (thường là súp, món chính, món ăn kèm và món tráng miệng). Một điều đặc biệt là giá của một menu luôn **rẻ hơn** hoặc **tương đương** tổng giá khi mua riêng các món trong menu. Cô ấy giúp bạn chi trả ít tiền hơn khi mua nhưng món có trong menu. Như vậy bạn sẽ tiết kiệm được túi tiền cho bản thân mình.

Bạn hãy thử tính xem cô ấy sẽ yêu cầu bạn trả bao nhiêu tiền với những món ăn mà bạn đã chọn.

**Chú ý:** Cô ấy sẽ tính nhiều món có trong menu theo giá của menu nếu nó rẻ hơn.

**INPUT:** Nhập vào từ file **INP.TXT**

* Dòng thứ nhất là số nguyên **K** (1 ≤ **K** ≤ 20), số món có trong canteen.
* Dòng tiếp theo là **K** số nguyên **ai** (1 ≤ **a­i**≤ 250), giá tiền của món ăn thứ **i**.
* Dòng tiếp theo là số nguyên X (1 ≤ **X** ≤ 1000), giá tiền của menu.
* Dòng tiếp theo là **4** số nguyên khác nhau (≤ **K**), 4 món ăn trên menu.
* Dòng tiếp theo là số nguyên **T** (1 ≤ **T** ≤ 20), số món ăn bạn đã chọn.
* Dòng cuối cùng là **T** số nguyên (≤ **K**), **T** món ăn được chọn.

**OUTPUT:** Đưa ra file **OUT.TXT** một số nguyên duy nhất là giá tiền mà bạn phải thanh toán.

**SAMPLE** **TEST:**

|  |  |
| --- | --- |
| **INPUT**  7  10 6 8 9 4 5 3  14  1 2 3 4  5  1 3 4 6 7  **OUTPUT**  22 | **INPUT**  6  12 5 7 8 9 3  14  4 3 1 2  5  1 2 1 6 6  **OUTPUT**  32 |

**Giải thích Sample Test 2:** như đã thấy, có hai món **1** được chọn nhưng Teta chỉ tính tiền một món theo giá gốc, món **1** còn lại được tính cùng món **2** với giá tiền menu. Hai món **6** không có trong menu nên tính tiền theo giá gốc => số tiền phải trả là 32.

**BÀI 96. KRIZA**

Sisyphus là một thợ khóa mới vào nghề. Anh được ông chủ cho luyện tập cách mở khóa. Ông chủ đưa cho Sisyphus một bộ chìa khóa gồm có **N** chìa móc vào một vòng tròn, bịt mắt và đẩy anh vào một căn phòng có **N** cánh cửa. Nhiệm vụ của anh là đi theo các bức tường tới các cánh cửa và mở - khóa tất cả các cánh cửa đó. Ban đầu, anh thử chiếc chìa khóa ở phía trái nhất, nếu thành công, anh sẽ đi tiếp sang cánh cửa tiếp theo và thử chiếc chìa vừa mở trước đó. Ngược lại nếu thất bại khi mở khóa, anh chuyển sang chìa khóa liền kề bên phải và cứ tiếp tục như vậy cho tới khi thành công.

Sau nhiều giờ luyện tập, anh chợt nhận ra mình đã đi thành một vòng tròn và mở - khóa **K** lần thành công. Sisyphus tự hỏi: “Mình đã dùng sai chìa khóa bao nhiêu lần rồi nhỉ? ”

Bạn hãy giúp Sisyphus trả lời câu hỏi của anh ấy nhé.

**INPUT:** Nhập vào từ file **KRIZA.INP**

* Dòng thứ nhất là hai số nguyên **N** và **K** (1 ≤ **N** ≤ 105,1 ≤ **K** ≤ 109) tương ứng với số chìa khóa (số cánh cửa tương ứng) và số lần mở thành công.
* **N** dòng tiếp theo, mỗi dòng chứa một số nguyên **ai** (1 ≤ **ai** ≤ **N**) là chiếc chìa khóa để mở cửa **ai**.

**OUTPUT:** Đưa ra file **KRIZA.OUT** một số nguyên duy nhất là đáp án cho câu hỏi của anh thợ khóa.

SAMPLE TEST:

|  |  |  |
| --- | --- | --- |
| **INPUT**  3 5  1  2  3  **OUTPUT**  4 | **INPUT**  4 6  4  2  1  3  **OUTPUT**  13 | **INPUT**  10 7  1  3  2  4  5  7  6  8  9  10  **OUTPUT**  25 |