**BÀI THỰC HÀNH SỐ 1**

1. Special Triangle [A]. Cho dãy số A[] gồm n số nguyên dương. Tam giác đặc biệt của dãy số A[] là tam giác được tạo ra bởi n hàng, trong đó hàng thứ n là dãy số A[], hàng i là tổng hai phần tử liên tiếp của hàng i+1 (1≤i≤n-1). Ví dụ A[] = {1, 2, 3, 4, 5}, khi đó tam giác được tạo nên như dưới đây:

[48]

[20, 28]

[8, 12, 16]

[3, 5, 7, 9 ]

[1, 2, 3, 4, 5 ]

Hãy viết chương trình in ra tam giác đặc biệt của dãy số A[].

1. **Tổng lũy thừa power(x, n) [C]**. Cho hai số nguyên x, n. Hãy tìm số các cách biểu diễn số x thành tổng lũy thừa bậc n của các số tự nhiên khác nhau.

Input: Output:

x = 10, n = 2 1 (10 = 12 + 32)

x = 100, n=2 3 ( 100 =102; 100 = 62 + 82; 100=12 + 32 +42+ 52 + 72)

1. **Số nghiệm nguyên không âm (Microsoft) [C]**. Cho hai số nguyên dương N và K. Hãy đếm số nghiệm nguyên không âm của phương trình x1 + x2 + x3 +…+xn = k. Ví dụ với n=4, k=1 thì số các nghiệm nguyên không âm của phương trình x1 + x2 + x3 + x4 = 1 là 4. Các nghiệm của phương trình bao gồm :(1, 0, 0, 0), (0, 1, 0, 0), (0, 0, 1, 0), (0, 0, 0, 1).

Input: Output:

2 4

4 1 70

5 4

1. **Gray Code Generation (Microsoft, Amazon) [A]**. Số nhị phân được xem là cách mặc định biểu diễn các số. Tuy nhiên, trong nhiều ứng dụng của điện tử và truyền thông lại dùng một biến thể của mã nhị phân đó là mã Gray. Mã Gray độ dài n có mã đầu tiên là n số 0, mã kế tiếp của nó là một xâu nhị phân độ dài n khác biệt với xâu trước đó một bít. Ví dụ với n=3 ta có 23 mã Gray như sau: 000, 001, 011, 010, 110, 111, 101, 100. Hãy viết chương trình liệt kê các mã Gray có độ dài n.

Dữ liệu vào:

* Dòng đầu tiên là số lượng test T (T≤10).
* T dòng kế tiếp ghi lại mỗi dòng một test. Mỗi test là một số tự nhiên n (n≤10).

Kết quả ra của mỗi test được đưa ra trên một dòng tương ứng với 2n mã Gray của test tương ứng. Hai mã gray khác nhau được ghi cách nhau một vài khoảng trống.

|  |  |
| --- | --- |
| Input: | Output: |
| 2  3  4 | 000 001 011 010 110 111 101 100  0000 0001 0011 0010 0110 0111 0101 0100 1100 1101 1111 1110 1010 1011 1001 1000 |

1. **Gray Code to Binay & Binary to Gray** **(Microsoft, Amazon) [A]**. Số nhị phân được xem là cách mặc định biểu diễn các số. Tuy nhiên, trong nhiều ứng dụng của điện tử và truyền thông lại dùng một biến thể của mã nhị phân đó là mã Gray. Mã Gray độ dài n có mã đầu tiên là n số 0, mã kế tiếp của nó là một xâu nhị phân độ dài n khác biệt với xâu trước đó một bít. Ví dụ với n=3 ta có 23 mã Gray như sau: 000, 001, 011, 010, 110, 111, 101, 100. Hãy viết chương trình chuyển đổi một xâu mã nhị phân X có độ dài n thành một xâu mã Gray.

Dữ liệu vào:

* Dòng đầu tiên là số lượng test T (T≤100).
* T dòng kế tiếp ghi lại mỗi dòng một test. Mỗi test là một xâu nhị phân độ dài n (n≤100).

Kết quả ra của mỗi test được đưa ra trên một dòng tương ứng với mã Gray của test tương ứng.

|  |  |
| --- | --- |
| Input: | Output: |
| 2  01001  01101 | 01101  01011 |

1. **Gray Code to Binay & Binary to Gray** **(Microsoft, Amazon) [A]**. Số nhị phân được xem là cách mặc định biểu diễn các số. Tuy nhiên, trong nhiều ứng dụng của điện tử và truyền thông lại dùng một biến thể của mã nhị phân đó là mã Gray. Mã Gray độ dài n có mã đầu tiên là n số 0, mã kế tiếp của nó là một xâu nhị phân độ dài n khác biệt với xâu trước đó một bít. Ví dụ với n=3 ta có 23 mã Gray như sau: 000, 001, 011, 010, 110, 111, 101, 100. Hãy viết chương trình chuyển đổi một xâu mã Gray X có độ dài n thành một xâu mã nhị phân.

Dữ liệu vào:

* Dòng đầu tiên là số lượng test T (T≤100).
* T dòng kế tiếp ghi lại mỗi dòng một test. Mỗi test là một xâu mã Gray độ dài n (n≤100).

Kết quả ra của mỗi test được đưa ra trên một dòng tương ứng với mã nhị phân của test tương ứng.

|  |  |
| --- | --- |
| Input: | Output: |
| 2  01101  01011 | 01001  01101 |

1. **Chia số trong xâu (Adopt)[C]**. Cho một xâu ký tự bao gồm các chữ số. Hãy tìm tất cả các số có thể tạo ra bằng cách kết hợp các số trong xâu nhưng giữ nguyên vị trí. Ví dụ với xâu S[] =”123” ta sẽ có các cách tạo các số như sau:

1 2 3 23 12 123

1. **Dãy sắp xếp tạo ra từ hai dãy sắp xếp (Microsoft)**. Cho hai dãy số đã được sắp xếp A[], B[]. Hãy in ra rất cả các dãy số đã được sắp xếp theo nguyên tắc: phần tử đầu tiên thuộc A[], phần tử tiếp theo thuộc B[]….

Ví dụ với dãy A[] = {10, 15, 25}, B[] = {1, 5, 20, 30 } ta sẽ có các dãy số được tạo ra theo nguyên tắc trên như sau:

10 20

10 20 25 30

10 30

15 20

15 20 25 30

15 30

1. 30

**11.** Cho số tự nhiên n và k. Hãy viết chương trình liệt kê tất cả các tổ hợp chập k của 1, 2, .., n. Ghi chú, mỗi tổ hợp chập k của 1, 2, .., n tương ứng với một tập con k phần tử của 1, 2, .., n. Ví dụ với n = 5, k =3 ta có 10 tập con ba phần tử của 1, 2, 3, 4, 5 như dưới đây:

|  |  |
| --- | --- |
| N=5, k = 3 |  |
| 1 2 3  1 2 4  1 2 5  1 3 4  1 3 5  1 4 5  2 3 4  2 3 5   1. 4 5 |  |

12. Ch hình chữ nhật gồm n×m hình vuông đơn vị. Hãy đếm số đường đi từ đỉnh của ô vuông cuối cùng góc trái lên đến đỉnh của ô vuông trên cùng góc phải. Biết mỗi bước đi ta chỉ được phép dịch sang phải hoặc lên trên theo các cạnh của hình vuông đơn vị.

Dữ liệu vào cho bởi file data.in theo khuôn dạng:

* Dòng đầu tiên ghi lại số T tương ứng với số lượng test.
* T dòng kế tiếp, mỗi dòng ghi lại một test. Mỗi test là một bộ đôi n, m tương ứng với hình chữ nhật gồm n×m hình vuông đơn vị.

Kết quả ra ghi lại trong file ketqua.out theo từng dòng ứng với mỗi test. Ví dụ dưới đây sẽ minh họa cho file data.in và ketqua.out của bài toán.

|  |  |
| --- | --- |
| Data.in | Ketqua.out |
| 3  5 3  6 3  7 4 | 10  20  35 |

1. Cho dãy số A =(a1, a2, .., an). Hãy liệt kê tất cả các dãy con k phần tử của dãy số A[] sao cho tổng các phẩn từ của dãy con đó đúng bằng số B cho trước. Dữ liệu vào cho bởi file data.in theo khuôn dạng:

* Dòng đầu tiên ghi lại số n, k, B.
* Dòng kế tiếp ghi lại n phần tử của dãy số A[].

Các dãy con k phần tử tìm được ghi lại trong file ketqua.out theo từng dòng. Ví dụ dưới đây sẽ minh họa cho file data.in và ketqua.out của bài toán.

|  |  |
| --- | --- |
| Data.in | Ketqua.out |
| 5 3 50  5 10 15 20 25 | 5 20 25   1. 5 25 |

1. Hãy viết chương trình liệt kê tất cả các hoán vị của 1, 2, .., n. Ví dụ với n = 4 sẽ cho ta 4! Hoán vị như dưới đây:

|  |  |
| --- | --- |
| N=4 |  |
| 1 2 3 4  1 2 4 3  1 3 2 4  1 3 4 2  1 4 2 3  1 4 3 2  2 1 3 4  2 1 4 3  2 3 1 4  2 3 4 1  2 4 1 3  2 4 3 1 | 3 1 2 4  3 1 4 2  3 2 1 4  3 2 4 1  3 4 1 2  3 4 2 1  4 1 2 3  4 1 3 2  4 2 1 3  4 2 3 1  4 3 1 2   1. 3 2 1 |

1. Hãy viết chương trình liệt kê tất cả các hoán vị của từ computer. Ví dụ từ computre là một hoán vị của từ compter.
2. Cho dãy số nguyên dương A[] = {a1, a2,.., an}. Hãy liệt kê tất cả các dãy số có thể tạo ra bằng cách tráo đổi các phần tử khác nhau của dãy số A[] sao cho tổng hai phần tử liên tiếp bất kỳ đều là một số nguyên tố. Dữ liệu vào cho bởi file data.in theo khuôn dạng:

* Dòng đầu tiên ghi lại số tự nhiên n là số phần tử của dãy số A[].
* Dòng kế tiếp ghi lại các phần tử của dãy số A[].

Các dãy số thỏa mã yêu cầu của bài toán ghi lại trong file ketqua.out, mỗi dãy số được ghi trên một dòng. Ví dụ dưới đây sẽ minh họa cho file data.in và ketqua.out của bài toán.

|  |  |
| --- | --- |
| Data.in | Kequa.out |
| 6  9 7 12 8 6 5 | 6 7 12 5 8 9  9 8 5 6 7 12  9 8 5 12 7 6  12 7 6 5 8 9 |

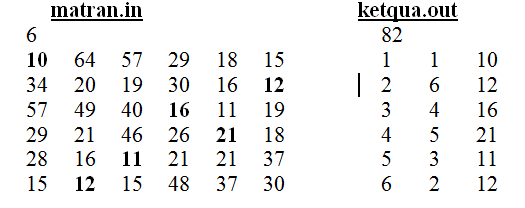
1. Cho ma trận vuông C = (*c*ij) cấp N (*1≤ i, j ≤ N≤100*) gồm *N*2 số tự nhiên (*các số không nhất thiết phải khác nhau*) ghi lại trong file matran.in theo khuôn dạng sau :

* Dòng đầu tiên ghi lại số tự nhiên N là cấp của ma trận vuông C;
* N dòng kế tiếp ghi lại ma trận vuông C = (*c*ij). Hai phần tử khác nhau của ma trận được ghi cách nhau bởi một vài khoảng trống.

Hãy viết chương trình lấy trên mỗi hàng, mỗi cột duy nhất một phần tử sao cho tổng các phần tử này là nhỏ nhất. Kết quả tìm được ghi lại trong file ketqua.out theo khuôn dạng:

* Dòng đầu tiên ghi lại tổng giá trị nhỏ nhất của N phần tử tìm được;
* N dòng kế tiếp, mỗi dòng ghi lại ba số i, j, *c*ij tương ứng với chỉ số hàng, chỉ số cột và giá trị phần tử tương ứng của ma trận. Ba số được viết cách nhau một vài khoảng trống.

Ví dụ về file **matran.in** và **ketqua.out**:



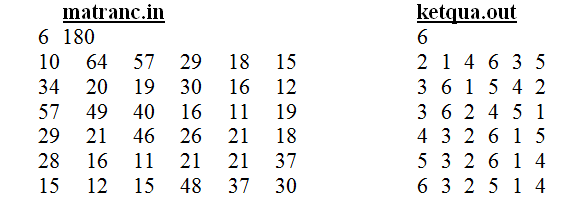
1. Cho ma trận vuông Ci,j cấp N (*1≤ i, j ≤ N≤100*) gồm N2 số tự nhiên và số tự nhiên K(*Các số không nhất thiết phải khác nhau*) ghi lại trong file matran.in theo khuôn dạng sau:

* Dòng đầu tiên ghi lại số tự nhiên N và K. Hai số được viết cách nhau một vài khoảng trống;
* N dòng kế tiếp ghi lại ma trận vuông Ci,j; Hai phần tử khác nhau của ma trận được ghi cách nhau bởi một vài khoảng trống.

Hãy viết chương trình lấy mỗi hàng, mỗi cột duy nhất một phần tử sao cho tổng các phần tử này đúng bằng K. Kết quả tìm được ghi lại trong file ketqua.out theo khuôn dạng:

* Dòng đầu tiên ghi lại số các nghiệm tìm được của bài toán.
* Những dòng kế tiếp, mỗi dòng ghi lại N số là một phương án của bài toán, số thứ i ghi lại giá trị j tương ứng với chỉ số cột của phần tử được lựa chọn. Các số được viết cách nhau một vài khoảng trống.

Ví dụ về file viec.in và ketqua.out:



1. Một người du lịch cần đi qua N thành phố (N≤100). Xuất phát tại thành phố số 1, người du lịch muốn qua tất cả các thành phố còn lại, mỗi thành phố đúng một lần rồi trở lại thành phố ban đầu. Biết chi phí đi lại từ thành phố thứ i đến thành phố thứ j là cij. Hãy viết chương trình tìm hành trình có chi phí tối đa cho người du lịch.

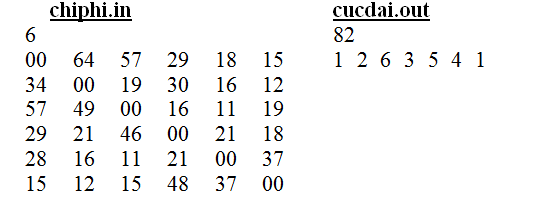
Dữ liệu vào cho bởi file **chiphi.in** theo khuôn dạng sau:

* Dòng đầu tiên ghi lại số tự nhiên N là cấp của ma trận chi phí vuông C = (cij);
* N dòng kế tiếp ghi lại ma trận vuông C = (*c*ij). Hai phần tử khác nhau của ma trận C = (cij) được ghi cách nhau bởi một vài khoảng trống.

Hành trình có chi phí lớn nhất tìm được ghi lại trong file **cucdai.out** theo khuôn dạng:

* Dòng đầu tiên ghi lại giá trị chi phí lớn nhất của hành trình tìm được;
* Dòng kế tiếp, ghi lại hành trình của người du lịch. Hai thành phố khác nhau của hành trình được ghi cách nhau một vài khoảng trống.

Ví dụ về dưới đây sẽ minh họa cho file **chiphi.in** và **cucdai.out** của bài toán.



1. **Xâu nhị phân tiến**. Cho xâu nhị phân X=(x1, x2,…,xn) có độ dài N (N<=100). Hãy tìm xâu nhị phân kế tiếp của xâu X ?

Input:

* Dòng đầu tiên ghi lại số T là số bộ test của bài toán (T<=100).
* Mỗi bộ test được viết trên 1 dòng ghi lại xâu nhị phân X.

Output: Đưa ra theo từng dòng ứng với mỗi bộ test là xâu nhị phân kế tiếp của X.

|  |  |
| --- | --- |
| Input: | Output |
| 2  1 0 1 0 1  1 0 1 1 1 1 | 1 0 1 1 0  1 1 0 0 0 0 |

1. **Tập con tiến**. Cho dãy số số tự nhiên X=(x1, x2,…,xk) là tập con K phần tử của 1, 2, .., N. Hãy cho tìm tập con kế tiếp K phần tử của dãy X ?

**Input**:

* Dòng đầu tiên ghi lại số T là số bộ test của bài toán (T<=100).
* Mỗi bộ test gồm hai dòng. Dòng đầu tiên ghi lại số tự nhiên N và K hai số được viết cách nhau một vài khoảng trống. Dòng tiếp theo ghi lại tập con K phần tử X. Các phần tử khác nhau của X được viết cách nhau một vài khoảng trống.

**Output**: Đưa ra theo từng dòng tập con K phần tử kế tiếp của X.

|  |  |
| --- | --- |
| Input: | Output |
| 2  5 3  1 2 3  7 4  1 3 4 6 | 1 2 4  1 3 4 7 |

1. **Hoán vị tiến**. Cho dãy số số tự nhiên X=(x1, x2,…,xN) là hoán vị của 1, 2, .., N (1< N≤ 100). Hãy tìm hoán vị kế tiếp của X?

**Input**:

* Dòng đầu tiên ghi lại số T là số bộ test của bài toán (T<=100).
* Mỗi bộ test gồm hai dòng. Dòng đầu tiên ghi lại số tự nhiên N. Hai dòng tiếp theo ghi lại hoán vị X. Các phần tử khác nhau của xâu X được viết cách nhau một vài khoảng trống.

**Output**: đưa ra theo từng dòng là hoán vị kế tiếp X.

|  |  |
| --- | --- |
| Input: | Output |
| 2  4  1 2 3 4  10  2 10 9 8 7 6 5 4 1 3 | 1 2 4 3  2 10 9 8 7 6 5 4 3 1 |

1. **Xâu nhị phân lùi**. Cho xâu nhị phân X=(x1, x2,…,xn) có độ dài N (N<=100). Hãy tìm xâu nhị phân đứng trước xâu X ?

**Input**:

* Dòng đầu tiên ghi lại số T là số bộ test của bài toán (T<=100).
* Mỗi bộ được viết trên một dòng là xâu nhị phân X.

Output: đưa ra theo từng dòng kết quả mỗi bộ test là xâu nhị phân đứng trước X. Đưa ra -1 nếu trước đó không còn bất kỳ xâu nhị phân nào.

|  |  |
| --- | --- |
| Input: | Output |
| 2  10101  1010111 | 10100  1010110 |

1. **Tập con lùi**. Cho dãy số số tự nhiên X=(x1, x2,…,xk) là tập con K phần tử của 1, 2, .., N. Hãy tìm tập con K phần tử trước đó X ?

**Input**:

* Dòng đầu tiên ghi lại số T là số bộ test của bài toán (T<=100).
* Mỗi bộ test gồm hai dòng. Dòng đầu tiên ghi lại số tự nhiên N và K hai số được viết cách nhau một vài khoảng trống. Dòng tiếp theo ghi lại tập con K phần tử X. Các phần tử khác nhau của X được viết cách nhau một vài khoảng trống.

**Output**: đưa ra theo từng dòng tập con K phần tử kế tiếp trước đó của X. Đưa ra -1 nếu trước đó không còn bất kỳ tập con k phần tử nào.

|  |  |
| --- | --- |
| Input | Output |
| 2  5 3  1 2 4  7 4  1 3 4 6 | 1 2 3  1 3 4 5 |

1. **Hoán vị lùi**. Cho dãy số số tự nhiên X=(x1, x2,…,xN) là hoán vị của 1, 2, .., N (1< N≤ 100). Hãy tìm hoán vị trước đó của X?

**Input:**

* Dòng đầu tiên ghi lại số T là số bộ test của bài toán (T<=100).
* Mỗi bộ test gồm hai dòng. Dòng đầu tiên ghi lại số tự nhiên N. Hai dòng tiếp theo ghi lại hoán vị X. Các phần tử khác nhau của xâu X được viết cách nhau một vài khoảng trống.

**Output**: đưa ra theo từng dòng là hoán vị trước X. Đưa ra -1 nếu trước đó không còn bất kỳ hoán vị nào.

|  |  |
| --- | --- |
| Input | Output |
| 2  4  1 2 4 3  5  2 1 3 5 4 | 2  1 2 3 4  2 2 3 4 5 |

1. **Bài toán n quân hậu** 1. Cho bàn cờ kích cỡ N×N. Hãy đếm số các cách đặt N quân hậu trên bàn cờ sao cho các quân hậu không ăn được lẫn nhau.

**Input**:

* Dòng đầu tiên ghi lại số T (T≤10) là số lượng test.
* T dòng kế tiếp ghi lại T test. Mỗi test là một số tự nhiên N ghi lại kích thước của bàn cờ.

Output:

* Đưa ra kết quả của mỗi test tương ứng trên một dòng.

|  |  |
| --- | --- |
| Input | Output |
| 3  4  7  8 | 2  40  92 |

1. **Bài toán n quân hậu** 2. Cho bàn cờ kích cỡ N×N. Cố định một quân hậu tại hàng s tại cột t. Hãy đếm số các cách đặt N-1 quân hậu còn lại trên các hàng khác nhau sao cho các quân hậu không ăn được lẫn nhau.

**Input**:

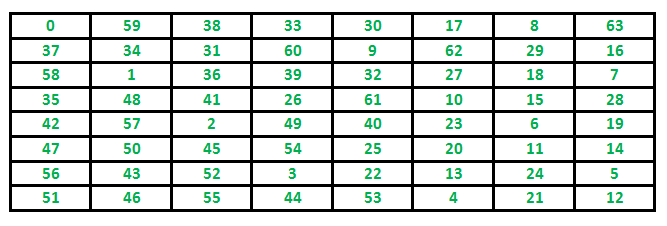
* Dòng đầu tiên ghi lại số T (T≤100) là số lượng test.
* T dòng kế tiếp ghi lại T test. Mỗi test gồm hai dòng: dòng đầu tiên ghi lại kích thước bàn cờ N, dòng kế tiếp ghi lại bộ đôi s, t (1≤N, s, t≤12).

**Output**:

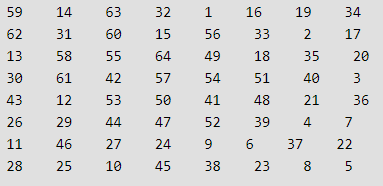
* Đưa ra kết quả của mỗi test tương ứng trên một dòng.

|  |  |
| --- | --- |
| Input | Output |
| 3  7  1 3  8  8 8 | 6  4 |

1. **Bài toán Mã đi tuần**. Cho bàn cờ kích thước N×N. Đặt quân mã ở ô đầu tiên (0,0), hãy tìm một hành trình của quân mã đi qua tất cả các ô trên bàn cờ. Dưới đây là một lời giải cho bàn cờ kích thước 8×8.



1. **Bài toán Mã đi tuần**. Cho bàn cờ kích thước N×N. Đặt quân mã ở ô đầu tiên (i,j), hãy tìm một hành trình của quân mã đi qua tất cả các ô trên bàn cờ. Dưới đây là một lời giải cho bàn cờ kích thước 8×8.



1. **0-1 Knapsack Problem (Bài toán cái túi 0-1)**. Cho dãy số nguyên dương A[] = (a1, a2, an), C=(c1, c2, .., cn) và W. Hãy tìm giá trị tối ưu của hàm

Input:

* Dòng đầu tiên ghi lại số lượng bộ test T (T≤10).
* Mỗi bộ test được tổ chức thành hai phần: dòng thứ nhất ghi lại N, W là số lượng phần tử của X[], A[], C[] và W. N dòng kế tiếp ghi lại n bộ đôi (ai, ci).

Output: Đưa ra kết quả mỗi test theo từng dòng.

|  |  |
| --- | --- |
| Input | Output |
| 1  3 50  10 60  20 100  30 120 | 220 |

1. **Fraction Knapsack Problem**. Cho dãy số nguyên dương A[] = (a1, a2, an), C=(c1, c2, .., cn) và W. Hãy tìm giá trị tối ưu của hàm

Input:

* Dòng đầu tiên ghi lại số lượng bộ test T (T≤10).
* Mỗi bộ test được tổ chức thành hai phần: dòng thứ nhất ghi lại N, W là số lượng phần tử của X[], A[], C[] và W. N dòng kế tiếp ghi lại n bộ đôi (ai, ci).

Output: Đưa ra kết quả mỗi test theo từng dòng.

|  |  |
| --- | --- |
| Input | Output |
| 1  3 50  10 60  20 100  30 120 | 240 |

1. **Activity Selection Problem**. Cho một hệ gồm N hành động. Mỗi hành động được biểu diễn bằng một bộ đôi (Si, Fi) tương ứng với thời gian bắt đầu và thời gian kết thúc mỗi hành động (i=0, 1, 2, .., N-1). Hãy lựa chọn tối đa các hành động có thể thực hiện được bằng một máy hoặc một người với giả thiết máy hoặc người chỉ thực hiện được ở chế độ đơn nhiệm.

Input:

* Dòng đầu tiên ghi lại số lượng test T (T≤100).
* Mỗi bộ test được tổ chức thành 3 dòng. Dòng đầu tiên ghi lại số N (N≤100) là số lượng hành động. Dòng kế tiếp ghi lại S =(S0, S1,.., SN-1). Các số được viết cách nhau một vài khoảng trống. Dòng cuối cùng ghi lại F =(F0, F1,.., FN-1). Các số được viết cách nhau một vài khoảng trống (1≤ Si, Fi ≤100000).

Output: Đưa ra kết quả mỗi test trên một dòng là danh sách nhiều nhất các hành động được thực thi.

|  |  |
| --- | --- |
| Input | Output |
| 2  6  1 3 0 5 8 5  2 4 6 7 9 9  8  75250 50074 43659 8931 11273 27545 50879 77924  112960 114515 81825 93424 54316 35533 73383 160252 | 1 2 4 5  6 7 1 |

1. **Bài toán n-ropes**. Cho hệ gồm n sợi dây có độ dài khác nhau. Nhiệm vụ của ta là cần nối n sợi dây lại với nhau thành một sợi. Chi phí nối giữa hai sợi dây được tính bằng tổng độ dài hai sợi. Hãy tìm phương án nối n sợi dây thành một sợi sao cho tổng chi phí nối dây là nhỏ nhất.

**Input**:

* Dòng đầu tiên ghi lại số lượng test T (T≤100).
* Mỗi bộ test được tổ chức thành hai dòng. Dòng đầu tiên ghi lại số N là số sợi dây (1≤N≤100). Dòng kế tiếp ghi lại N số Li tương ứng với độ dài sợi dây thứ i (1≤Li≤500). Các số được viết cách nhau một vài khoảng trống.

Output: Đưa ra kết quả mỗi test trên một dòng là phương án nối dây có chi phí nhỏ nhất.

|  |  |
| --- | --- |
| Input | Output |
| 2  4  4 2 3 6  8  9 7 12 8 6 5 14 4 | 29  190 |

1. **d-string**. Xâu S được gọi là một d-string nếu ta có thể sắp đặt lại các ký tự trong S sao cho các ký tự giống nhau đều cách nhau một khoảng là d. Ví dụ xâu S=”AAB” là một 2-string vì ta có cách sắp đặt lại các ký tự trong S thành xâu “ABA”. Xâu S=”GEEKSFORGEEKS” là một 3-string vì ta có cách sắp đặt lại các ký tự trong S thành xâu “GKEGKESFESFOR”. Cho xâu ký tự S và số tự nhiên d. Hãy kiểm tra xem xâu S có phải là một d-string hay không?

**Input**:

* Dòng đầu tiêng ghi lại số lượng test T (T≤100).
* Mỗi test được tổ chức thành hai dòng. Dòng đầu tiên ghi lại số d. Dòng kế tiếp ghi lại xâu ký tự S.

Output:

* Đưa ra kết quả “YES” hoặc “NO” của mỗi test trên một dòng tương ứng với S là một d-string hay S không phải là một d-string.

|  |  |
| --- | --- |
| Input | Output |
| 3  2  AAA  2  AAB  3  GEEKSFORGEEKS | NO  YES  YES |

1. Cho số tự nhiên N. Hãy tìm các số nguyên dương sao giai thừa của số đó kết thúc bởi đúng N số 0. Ví dụ N=1 ta có các số 5, 6, 7, 8, 9 vì 5!=120, 6!=720, 7!=5040, 8!=40320, 9!=362880.

**Input**:

* Dòng đầu tiên ghi lại số lượng test T (T≤30).
* T dòng kế tiếp ghi lại mỗi dòng một test. Mỗi test là một số tự nhiên N≤100.

Output:

* Đưa ra các số nguyên dương thỏa mãn yêu cầu bài toán theo từng dòng.

|  |  |
| --- | --- |
| Input | Output |
| 3  1  2  5 | 5 6 7 8 9  10 11 12 13 14  0 |

1. Cho mảng A[] gồm n phần tử và số k. Các số trong A[] có thể là số âm, dương, có giá trị khác nhau hoặc trùng nhau. Hãy tìm số gần nhất với số k.

Input:

* Dòng đầu tiên ghi lại số lượng test T (T≤100);
* Những dòng kế tiếp ghi lại các test. Mỗi test được tổ chức thành 2 dòng. Dòng đầu tiên đưa vào hai số n và k. Hai số được viết cách nhau một vài khoảng trống. Dòng kế tiếp đưa vào n số của mảng A[]. Các số viết cách nhau một vài khoảng trống.

Output:

* Đưa ra kết quả của mỗi test là số nguyên gần nhất với k.

|  |  |
| --- | --- |
| Input | Output |
| 2  8 11  1 2 4 5 6 6 8 9  7 4  2 5 6 7 8 8 9 | 9  5 |

1. Cho hai dãy số A[] gồm N phần tử, B[] gồm M phần tử và số tự nhiên K. Hãy tìm phần tử thứ K sau khi sắp xếp cả hai dãy số.

Input:

* Dòng đầu tiên ghi lại số lượng test T (T≤100).
* Những dòng kế tiếp ghi lại các test. Mỗi test được tổ chức thành 3 dòng. Dòng đầu tiên ghi lại bộ ba số N, M, K. Dòng tiếp theo ghi lại N số của mảng A[]. Dòng cuối cùng ghi lại M số của mảng B[]. Các số thỏa mãn dàng buộc 1≤N, M≤1000; 1≤K≤M+N; 1≤A[i], B[i]≤1000.

Output:

* Kết quả của mỗi test ghi lại trên 1 dòng là giá trị của phần tử thứ K sau khi sắp xếp hai mảng số.

|  |  |
| --- | --- |
| Input | Output |
| 1  5 4 5  2 3 6 7 9  1 4 8 10 | 6 |

1. Cho số tự nhiên N. Hãy đếm số các cách phân tích số tự nhiên N thành tổng các số tự nhiên nhỏ hơn N. Các cách phân tích là hoán vị của nhau chỉ được tính là một cách. Ví dụ với n = 5 ta có 7 cách chia như sau: 5+0, 4 + 1, 3 + 2, 3 + 1 + 1, 2 + 2 + 1 , 2 + 1 + 1 + 1, 1+1+1+1+ 1.

Input:

* Dòng đầu tiên đưa vào số lượng test T (T≤50).
* Những dòng kế tiếp mỗi dòng ghi lại một test. Mỗi test là một số tự nhiên N.

Output:

* Đưa ra theo từng dòng số các cách phân tích N thành tổng các số nhở hơn N.

1. Hãy đếm số các xâu nhị phân độ dài n không có dãy k số 0 liên tiếp. Ví dụ với n=4, k=2 ta có 8 xâu nhị phân độ dài 4 không có dãy 2 số 0 liên tiếp như sau: “0101” , “0110”, , “0111”, “1010”, “1011” , “1101” , “1110” , “1111”.

Input:

* Dòng đầu tiên đưa vào số lượng test T (T≤100).
* Những dòng kế tiếp đưa vào các test. Mỗi test được ghi trên một dòng là bộ đôi n, k. n và k thỏa mãn dàng buộc sau: 1≤k ≤n≤20).

Output:

* Đưa ra theo từng dòng số các xâu nhị phân độ dài n thỏa mãn yêu cầu bài toán.

|  |  |
| --- | --- |
| Input | Output |
| 1  4 2 | 8 |

1. Hãy đếm số các xâu nhị phân độ dài n có ít nhất môt dãy k số 0 liên tiếp. Ví dụ với n=4, k=2 ta có 8 xâu nhị phân độ dài 4 có ít nhất dãy 2 số 0 liên tiếp như sau: “0000” , “0001”, , “0010”, “0011”, “0100” , “1000” , “1001” , “1100”.

Input:

* Dòng đầu tiên đưa vào số lượng test T (T≤100).
* Những dòng kế tiếp đưa vào các test. Mỗi test được ghi trên một dòng là bộ đôi n, k. n và k thỏa mãn dàng buộc sau: 1≤k ≤n≤20).

Output:

* Đưa ra theo từng dòng số các xâu nhị phân độ dài n thỏa mãn yêu cầu bài toán.

|  |  |
| --- | --- |
| Input | Output |
| 1  4 2 | 8 |