



[DSA WEEKLY CONTEST T6 2025]. TEST 6. SINH & QUAY LUI, NHÁNH CẶN

[Info](#)[Statistics](#)[Rankings](#)[Participation](#)[Submissions](#)[Virtual join](#)[DSA](#)

[Thuật Toán Sinh]. Bài 31. Tổng các tập con

[Submit](#)**Time limit:** 1.0 / **Memory limit:** 256M**Point:** 100

Cho mảng **A[]** gồm **N** phần tử, bạn hãy liệt kê tất cả các tổng khác nhau của các tập con khác rỗng của mảng **A[]**.

Ví dụ A[] = {1, 2, 3} có thể tạo thành các tổng 1, 2, 3, 4, 5, 6

Đầu vào

Dòng đầu tiên là số **N**

Dòng 2 gồm **N** số trong mảng **A[]**

Giới hạn

 $1 \leq N \leq 20$ $1 \leq A[i] \leq 10^9$

Đầu ra

In ra các tổng khác nhau theo thứ tự tăng dần

Ví dụ :



3

1 3 5

Copy

Output 01

1 3 4 5 6 8 9

Copy

[Thuật Toán Sinh]. Bài 32. Cụm từ bí mật passphrase

Submit

Time limit: 1.0 / **Memory limit:** 256M**Point:** 100

Passphrase là cụm từ bí mật được sử dụng trong các ví tiền điện tử hiện nay, biết được cụm từ bí mật của 1 ví tiền điện tử thì bạn có thể lấy tất cả tiền điện tử trong ví này. Hiện nay có các bạn trẻ Việt Nam bán tool dò **Passphrase**, họ sẽ sinh ra 12 từ ngẫu nhiên từ 2048 từ trong bộ từ **BIP39**.

Bây giờ **28tech** muốn bạn triển khai 1 tool dò **Passphrase**, bạn hãy sinh ra tất cả các bộ cụm từ gồm 6 từ ngẫu nhiên từ **N** từ cho trước. Bạn cần liệt kê các cụm **Passphrase** theo thứ tự từ điển tăng dần

Đầu vào

Dòng đầu tiên là số **N**

Dòng 2 gồm **N** từ trong bộ từ **BIP39**

Giới hạn

 $1 \leq N \leq 10$

Đầu ra

In ra các cụm **Passphrase** tạo được



Ví dụ :

Input 01

2

badge cart

Copy

Output 01

Copy



badge badge badge badge badge cart
badge badge badge badge cart badge
badge badge badge badge cart cart
badge badge badge cart badge badge
badge badge badge cart badge cart
badge badge badge cart cart badge
badge badge badge cart cart cart
badge badge cart badge badge badge
badge badge cart badge badge cart
badge badge cart badge cart badge
badge badge cart badge cart cart
badge badge cart cart badge badge
badge badge cart cart badge cart
badge badge cart cart cart badge
badge badge cart cart cart cart
badge cart badge badge badge badge
badge cart badge badge badge cart
badge cart badge badge cart badge
badge cart badge badge cart cart
badge cart badge cart badge badge
badge cart badge cart badge cart
badge cart badge cart cart badge
badge cart badge cart cart cart
badge cart cart badge badge badge
badge cart cart badge badge cart
badge cart cart badge cart badge
badge cart cart badge cart cart
badge cart cart cart badge badge
badge cart cart cart badge cart
badge cart cart cart cart badge
badge cart cart cart cart cart
cart badge badge badge badge badge
cart badge badge badge badge cart
cart badge badge badge cart badge
cart badge badge badge cart cart
cart badge badge cart badge badge
cart badge badge cart badge cart
cart badge badge cart cart badge
cart badge badge cart cart cart
cart badge cart badge badge badge
cart badge cart badge badge cart



```

cart badge cart cart badge badge
cart badge cart cart badge cart
cart badge cart cart cart badge
cart badge cart cart cart cart
cart cart badge badge badge badge
cart cart badge badge badge cart
cart cart badge badge cart badge
cart cart badge badge cart cart
cart cart badge cart badge badge
cart cart badge cart badge cart
cart cart badge cart cart badge
cart cart badge cart cart cart
cart cart cart badge badge badge
cart cart cart badge badge cart
cart cart cart badge cart badge
cart cart cart badge cart cart
cart cart cart cart badge badge
cart cart cart cart badge cart
cart cart cart cart cart badge
cart cart cart cart cart cart

```

[Thuật Toán Sinh]. Bài 33. Địa chỉ ví điện tử

[Submit](#)

Time limit: 1.0 / **Memory limit:** 256M

Point: 100

28tech đang muốn phát triển đồng tiền điện tử là **28coin**, anh ta cần tạo ra các địa chỉ ví để gửi và nhận đồng tiền này. Mỗi địa chỉ ví là sự kết hợp của hoán vị các chữ cái từ 'a' tới **X** với **X** là chữ cái in thường cho trước với các tổ hợp chập **K** của **N** phần tử các số nguyên từ 1 tới **N**.

Nhiệm vụ của bạn là hãy liệt kê tất cả các địa chỉ có thể có.

Đầu vào

Dòng duy nhất chứa 2 số **N**, **K** và kí tự **X**



$1 \leq K \leq N \leq 15$

Đầu ra

In ra các địa chỉ ví

Ví dụ :

Input 01

5 3 c

Copy

Output 01

Copy



Hello, Nguyễn Năng Anh.

abc124
abc125
abc134
abc135
abc145
abc234
abc235
abc245
abc345
acb123
acb124
acb125
acb134
acb135
acb145
acb234
acb235
acb245
acb345
bac123
bac124
bac125
bac134
bac135
bac145
bac234
bac235
bac245
bac345
bca123
bca124
bca125
bca134
bca135
bca145
bca234
bca235
bca245
bca345
cab123
cab124



cab135
cab145
cab234
cab235
cab245
cab345
cba123
cba124
cba125
cba134
cba135
cba145
cba234
cba235
cba245
cba345

[Quay Lui - Nhánh CẶN]. Bài 30. Phân hoạch 2

[Submit](#)

Time limit: 2.0 / **Memory limit:** 256M

Point: 100

Cho một số nguyên dương **N** và hãy liệt kê tất cả các phân hoạch biểu diễn **N** dưới dạng tổng các số tự nhiên nhỏ hơn hoặc bằng nó.

Ví dụ với **4** bạn có 8 cách biểu diễn :

1 1 1 1, 1 1 2, 1 2 1, 1 3, 2 1 1, 2 2, 3 1, 4

Đầu vào

- Dòng duy nhất chứa **N**

Giới hạn

- $1 \leq N \leq 15$



- Dòng 1 in ra số cách biểu diễn
- Các dòng tiếp theo in ra các cách biểu diễn theo thứ tự tăng dần về từ điển

Ví dụ :

Input 01

4

Copy

Output 01

8
1 1 1 1
1 1 2
1 2 1
1 3
2 1 1
2 2
3 1
4

Copy

[Quay Lui - Nhánh CẶN]. Bài 31. Đường đi ngắn nhất trên bảng số

Submit

Time limit: 1.0 / **Memory limit:** 256M

Point: 100

Cho một bảng số kích thước **N x M**. Chi phí khi đi qua ô **(i,j)** bằng **A[i][j]**.

Nhiệm vụ của bạn là hãy tìm một đường đi từ ô (1, 1) tới ô **(N, M)** sao cho chi phí là nhỏ nhất.



Đầu vào

- Dòng 1 bắt đầu bởi hai số nguyên **N** và **M**.
- **N** dòng tiếp theo, mỗi dòng gồm **M** số nguyên **A[i][j]**

Giới hạn

- $1 \leq N, M \leq 10$
- $0 \leq A[i][j] \leq 9$

Đầu ra

- In ra một số nguyên là chi phí nhỏ nhất cho đường đi tìm được.

Ví dụ :

Input 01

```
6 7
0 3 6 0 5 9 1
6 5 4 4 0 7 6
4 0 2 1 5 6 1
2 7 7 3 3 1 6
4 4 9 6 9 7 2
3 6 4 4 1 9 2
```

[Copy](#)

Output 01

```
28
```

[Copy](#)



Hello, Nguyễn Năng Anh.

Được phát triển bởi **28TECH.COM.VN**