

## 2.2. Thuật toán sinh (Generation)

Thuật toán sinh được dùng để giải lớp các bài toán thỏa mãn hai điều kiện:

- *Xác định được một thứ tự trên tập các cấu hình cần liệt kê của bài toán. Biết được cấu hình đầu tiên, biết được cấu hình cuối cùng.*
- *Từ một cấu hình cuối cùng, ta xây dựng được thuật toán sinh ra cấu hình đứng ngay sau nó theo thứ tự.*

**Thuật toán:**

**Thuật toán Generation:**

**Bước1 (Khởi tạo):**

*<Thiết lập cấu hình đầu tiên>;*

**Bước 2 (Bước lặp):**

**while** (*<Lặp khi cấu hình chưa phải cuối cùng>*) **do**

*<Đưa ra cấu hình hiện tại>;*

*<Sinh ra cấu hình kế tiếp>;*

**endwhile;**

**End.**

**Ví dụ 1.** Duyệt các xâu nhị phân có độ dài  $n$ .

**Lời giải.** Xâu  $X = (x_1, x_2, \dots, x_n) : x_i = 0, 1; i=1, 2, \dots, n$  được gọi là xâu nhị phân có độ dài  $n$ . Ví dụ với  $n=4$ , ta có 16 xâu nhị phân dưới đây:

$STT$	$X = (x_1, \dots, x_n)$	$F(X)$	$STT$	$X = (x_1, \dots, x_n)$	$F(X)$
1	0000	0	9	1000	8
2	0001	1	10	1001	9
3	0010	2	11	1010	10
4	0011	3	12	1011	11
5	0100	4	13	1100	12
6	0101	5	14	1101	13
7	0110	6	15	1110	14
8	0111	7	16	1111	15

## Thuật toán sinh xâu nhị phân kế tiếp;

Input:

- +  $X = (x_1, x_2, \dots, x_n)$  là biến toàn cục.
- +  $OK = 1$  là biến toàn cục

Output:  $2^n$  xâu nhị phân có độ dài  $n$ .

```
Void Next_Bit_String(void) {  
    int i=n;  
    while (i>0 && X[i]!=0) { X[i] = 0; i--; }  
    if (i >0) X[i]=1;  
    else OK = 0;  
}
```

Actions:

```
X =(0,0,...,0); // Xâu nhị phân ban đầu.  
while (OK) { //Lặp khi xâu chưa phải cuối cùng  
    Result(); //Đưa ra xâu hiện tại>;  
    Next_Bit_String(); //Sinh ra xâu kế tiếp  
}
```

Endactions

## Bài tập. Hãy cho biết kết quả thực hiện chương trình dưới đây?

```
#include <iostream.h>
#include <stdlib.h>
#define MAX 100
#define TRUE 1
#define FALSE 0
int n, X[MAX], OK=TRUE, dem=0;
void Init (void ){
    cout<<"\n Nhap n=";<<cin>>n;
    for (int i=1; i<=n; i++)
        X[i] =0;
}
void Result(void){
    cout<<"\n Ket qua "<<++dem<<":";
    for (int i=1; i<=n; i++)
        cout<<X[i]<<"  ";
}
```

```
void Next_Bit_String(void) {
    int i= n;
    while (i>0 && X[i]!=0){
        X[i] = 0; i --;
    }
    if (i > 0 ) X[i] = 1;
    else OK = FALSE;
}
int main() {
    Init(); //Nhap n = 4
    while (OK ){
        Result();
        Next_Bit_String();
    }
    system("PAUSE");
    return 0;
}
```

**Ví dụ 2.** *Duyệt các tổ hợp chập K của 1, 2,..., N.*

**Lời giải.** Mỗi tổ hợp chập K của 1, 2, ..., N là một tập con K phần tử khác nhau của 1, 2,..., N. Ví dụ với N=5, K= 3 ta sẽ có C(N,K) tập con dưới đây

STT	Tập con $X = (x_1, \dots, x_k)$
1	1 2 3
2	1 2 4
3	1 2 5
4	1 3 4
5	1 3 5
6	1 4 5
7	2 3 4
8	2 3 5
9	2 4 5
10	3 4 5

**Thứ tự tự nhiên.** Duyệt các tổ hợp chập  $K$  của  $1, 2, \dots, N$ .

Có thể xác định được nhiều trật tự khác nhau trên các tổ hợp. Tuy nhiên, thứ tự đơn giản nhất có thể được xác định như sau:

Ta gọi tập con  $X = (x_1, \dots, x_K)$  là đứng trước tập con  $Y = (y_1, y_2, \dots, y_K)$  nếu tìm được chỉ số  $t$  sao cho  $x_1 = y_1, x_2 = y_2, \dots, x_{t-1} = y_{t-1}, x_t < y_t$ . Ví dụ tập con  $X = (1, 2, 3)$  đứng trước tập con  $Y = (1, 2, 4)$  vì với  $t=3$  thì  $x_1 = y_1, x_2 = y_2, x_3 < y_3$ .

Tập con đầu tiên là  $X = (1, 2, \dots, K)$ , tập con cuối cùng là  $(N-K+1, \dots, N)$ . Như vậy điều kiện 1 của thuật toán sinh được thỏa mãn.

Thuật toán sinh tổ hợp:

```
Void Next_Combination(void) {  
    int i = k; // Xuất phát từ phần tử cuối cùng của tổ hợp  
    while ( i>0 && X[i] == N - K + i) i --; //Tìm phần tử  $X[i] \neq N-K+i$   
    if (i>0) { //Nếu i chưa vượt quá phần tử cuối cùng  
        X[i] = X[i] + 1; //Thay  $X[i] = X[i] + 1$   
        for (int j = i+1; j<=k; j++) //Từ phần tử thứ j +1 đến k  
            X[j] = X[i] + j - i; // Thay thế  $X[j] = X[i] + j - i$   
    }  
    else OK = 0; //OK =0 nếu đã đến tập con cuối cùng  
}
```

## Bài tập. Hãy cho biết kết quả thực hiện chương trình dưới đây?

```
#include <iostream.h>
#include <stdlib.h>
#define MAX 100
#define TRUE 1
#define FALSE 0
int n,k,X[MAX],OK=TRUE,dem=0;
void Init (void ){
    cout<<"\n Nhap n=";cin>>n;
    cout<<"\n Nhap k=";cin>>k;
    for (int i=1; i<=k; i++)
        X[i] =i;
}
void Result(void){
    cout<<"\n Ket qua buoc "<<++dem<<":";
    for (int i=1; i<=k; i++)
        cout<<X[i]<<"  ";
}
```

```
void Next_Combination(void) {
    int i= k;
    while (i>0 && X[i]==n-k+i)i--;
    if (i > 0 ) {
        X[i] = X[i] +1;
        for (int j = i+1; j<=k; j++)
            X[j] = X[i] + j - i;
    }
    else OK = FALSE;
}
int main()
{
    Init(); //Nhap n = 5, k = 3
    while (OK ){
        Result();
        Next_Combination();
    }
    system("PAUSE");
    return 0;
}
```



**Ví dụ 3.** Duyệt các hoán vị của 1, 2,..., N.

**Lời giải.** Mỗi hoán vị của 1, 2, ..., N là một cách xếp có tính đến thứ tự của 1, 2,...,N. Số các hoán vị là  $N!$ . Ví dụ với  $N = 3$  ta có 6 hoán vị dưới đây.

**Thứ tự tự nhiên.** Có thể xác định được nhiều trật tự khác nhau trên các hoán vị. Tuy nhiên, thứ tự đơn giản nhất có thể được xác định như sau. Hoán vị  $X = (x_1, x_2, \dots, x_n)$  được gọi là đứng sau hoán vị  $Y = (y_1, y_2, \dots, y_n)$  nếu tồn tại chỉ số  $k$  sao cho

$x_1 = y_1, x_2 = y_2, \dots, x_{k-1} = y_{k-1}, x_k < y_k$ . Ví dụ hoán vị  $X = (1, 2, 3)$  được gọi là đứng sau hoán vị  $Y = (1, 3, 2)$  vì tồn tại  $k = 2$  để  $x_1 = y_1$ , và  $x_2 < y_2$ .

STT	Hoán vị $X = (x_1, \dots, x_N)$
1	1 2 3
2	1 3 2
3	2 1 3
4	2 3 1
5	3 1 2
6	3 2 1



```

Void Next_Permutation(void) {
    int j = N-1; // Xuất phát từ phần tử N-1
    while ( j>0 && X[j]> X[j+1]) j --; //Tìm j sao cho X[j]>X[j+1]
    if (j>0) { //Nếu i chưa vượt quá phần tử cuối cùng
        int k =N; // Xuất phát từ k = N
        while ( X[j] > X[k] ) k --; // Tìm k sao cho X[j] <X[k]
        int t = X[j]; X[j] = X[k]; X[k] = t; //Đổi chỗ X[j] cho X[k]
        int r = j +1, s = N;
        while ( r <=s ) { //Lật ngược đoạn từ j +1 đến N
            t = r; r = s; s=t;
            r ++; s --;
        }
    }
    else OK =0; //Nếu đến hoán vị cuối cùng
}

```

## Bài tập. Hãy cho biết kết quả thực hiện chương trình dưới đây?

```
#include <iostream.h>
#include <stdlib.h>
#define MAX 100
#define TRUE 1
#define FALSE 0
int n,X[MAX],OK=TRUE,dem=0;
void Init (void ){
    cout<<"\n Nhap n=";<<cin>>n;
    for (int i=1; i<=n; i++)
        X[i] =i;
}
void Result(void){
    cout<<"\n Ket qua buoc " << ++dem << ":";
    for (int i=1; i<=n; i++)
        cout<<X[i]<<"  ";
}
```

```
void Next_Permutation(void) {
    int j= n-1;
    while (j>0 && X[j]>X[j+1]) j--;
    if (j > 0 ) {
        int k =n;
        while(X[j]>X[k]) k--;
        int t = X[j]; X[j]=X[k]; X[k]=t;
        int r = j +1 , s =n;
        while (r<=s ) {
            t = X[r]; X[r]=X[s]; X[s] =t;
            r ++; s--;
        }
    }
    else OK = FALSE;
}

void main() {
    Init(); //Nhap n = 4
    while (OK ){ Result();
        Next_Permutation();
    }
}
```

**Ví dụ 4.** Cho số tự nhiên  $N$  ( $N \leq 100$ ). Hãy liệt kê tất cả các cách chia số tự nhiên  $N$  thành tổng của các số tự nhiên nhỏ hơn  $N$ . Các cách chia là hoán vị của nhau chỉ được tính là một cách. Ví dụ với  $N = 5$  ta có 7 cách chia như sau:

5					
4	1				
3	2				
3	1	1			
2	2	1			
2	1	1	1		
1	1	1	1	1	

## //Hãy cho biết kết quả thực hiện ?

```
#include <iostream.h>
#include <stdlib.h>
#define MAX 100
#define TRUE 1
#define FALSE 0
int n, k, X[MAX], dem =0, OK =TRUE;
void Init(void ){
    cout<<"\n Nhap n=";  cin>>n;
    k = 1; X[k] = n;
}
void Result(void) {
    cout<<"\n Cach chia "<<++dem<<":";
    for (int i=1; i<=k; i++)
        cout<<X[i]<<" ";
}
```

```
void Next_Division(void ){
    int i = k, j, R, S,D;
    while (i > 0 && X[i]==1 ) i--;
    if (i>0 ) {
        X[i] = X[i] - 1;  D = k - i + 1;
        R = D / X[i];    S = D % X[i];
        k= i;
        if (R>0) {
            for ( j = i +1; j<=i + R; j++)
                X[j] = X[i];
            k = k + R;
        }
        if (S>0 ){
            k = k +1; X[k] = S;
        }
    }
    else OK =0;
}

int main() {  Init(); //Nhap n = 5.
    while (OK ) {
        Result(); Next_Division();
    }
    return 0;
}
```

## BÀI TẬP

### BÀI 1. XÂU NHỊ PHÂN KẾ TIẾP

Cho xâu nhị phân  $X[]$ , nhiệm vụ của bạn là hãy đưa ra xâu nhị phân tiếp theo của  $X[]$ .

Ví dụ  $X[] = "010101"$  thì xâu nhị phân tiếp theo của  $X[]$  là "010110".

Input:

- Dòng đầu tiên đưa vào số lượng test  $T$ .
- Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test là một xâu nhị phân  $X$ .
- $T, X[]$  thỏa mãn ràng buộc:  $1 \leq T \leq 100$ ;  $1 \leq \text{length}(X) \leq 10^3$ .

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input	Output
2	010110
010101	000000
111111	

## BÀI 2. TẬP CON KẾ TIẾP

Cho hai số  $N, K$  và một tập con  $K$  phần tử  $X[] = (X_1, X_2, \dots, X_K)$  của  $1, 2, \dots, N$ . Nhiệm vụ của bạn là hãy đưa ra tập con  $K$  phần tử tiếp theo của  $X[]$ . Ví dụ  $N=5, K=3, X[] = \{2, 3, 4\}$  thì tập con tiếp theo của  $X[]$  là  $\{2, 3, 5\}$ .

Input:

- Dòng đầu tiên đưa vào số lượng test  $T$ .
- Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test gồm hai dòng: dòng thứ nhất là hai số  $N$  và  $K$ ; dòng tiếp theo đưa vào  $K$  phần tử của  $X[]$  là một tập con  $K$  phần tử của  $1, 2, \dots, N$ .
- $T, K, N, X[]$  thỏa mãn ràng buộc:  $1 \leq T \leq 100; 1 \leq K \leq N \leq 10^3$ .

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input	Output
2	2 3 4
5 3	1 2 3
1 4 5	
5 3	
3 4 5	

### BÀI 3. HOÁN VỊ KẾ TIẾP

Cho số tự nhiên  $N$  và một hoán vị  $X[]$  của  $1, 2, \dots, N$ . Nhiệm vụ của bạn là đưa ra hoán vị tiếp theo của  $X[]$ . Ví dụ  $N=5$ ,  $X[] = \{1, 2, 3, 4, 5\}$  thì hoán vị tiếp theo của  $X[]$  là  $\{1, 2, 3, 5, 4\}$ .

Input:

- Dòng đầu tiên đưa vào số lượng test  $T$ .
- Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test gồm hai dòng: dòng thứ nhất là số  $N$ ; dòng tiếp theo đưa vào hoán vị  $X[]$  của  $1, 2, \dots, N$ .
- $T, N, X[]$  thỏa mãn ràng buộc:  $1 \leq T \leq 100$ ;  $1 \leq N \leq 10^3$ .

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input	Output
2	1 2 3 5 4
5	1 2 3 4 5
1 2 3 4 5	
5	
5 4 3 2 1	



#### BÀI 4. XÂU AB CÓ ĐỘ DÀI N

Xâu ký tự str được gọi là xâu AB nếu mỗi ký tự trong xâu hoặc là ký tự 'A' hoặc là ký tự 'B'. Ví dụ xâu str="ABBABB" là xâu AB độ dài 6. Nhiệm vụ của bạn là hãy liệt kê tất cả các xâu AB có độ dài n.

Input:

- Dòng đầu tiên đưa vào số lượng test T.
- Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test là một số tự nhiên n.
- T, n thỏa mãn ràng buộc:  $1 \leq T \leq 10$ ;  $1 \leq n \leq 10$ .

Output:

- Đưa ra kết quả mỗi test theo từng dòng. Mỗi xâu cách nhau 1 khoảng trống.

Input	Output
2	AA AB BA BB
2	AAA AAB ABA ABB BAA BAB BBA BBB
3	

## BÀI 5. SINH TỔ HỢP

Cho hai số nguyên dương  $N$  và  $K$ . Nhiệm vụ của bạn là hãy liệt kê tất cả các tập con  $K$  phần tử của  $1, 2, \dots, N$ . Ví dụ với  $N=5, K=3$  ta có 10 tập con của  $1, 2, 3, 4, 5$  như sau:  $\{1, 2, 3\}, \{1, 2, 4\}, \{1, 2, 5\}, \{1, 3, 4\}, \{1, 3, 5\}, \{1, 4, 5\}, \{2, 3, 4\}, \{2, 3, 5\}, \{2, 4, 5\}, \{3, 4, 5\}$ .

Input:

- Dòng đầu tiên đưa vào số lượng test  $T$ .
- Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test là một cặp số tự nhiên  $N, K$  được viết trên một dòng.
- $T, n$  thỏa mãn ràng buộc:  $1 \leq T \leq 100; 1 \leq k \leq n \leq 15$ .

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input	Output
2	123 124 134 234
4 3	123 124 125 134 135 145 234 235 245 345
5 3	

## BÀI 6. SINH HOÁN VỊ

Cho số nguyên dương  $N$ . Nhiệm vụ của bạn là hãy liệt kê tất cả các hoán vị của  $1, 2, \dots, N$ . Ví dụ với  $N = 3$  ta có kết quả: 123, 132, 213, 231, 312, 321.

Input:

- Dòng đầu tiên đưa vào số lượng test  $T$ .
- Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test là một số tự nhiên  $N$  được viết trên một dòng.
- $T, n$  thỏa mãn ràng buộc:  $1 \leq T, N \leq 10$ .

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input	Output
2	12 21
2	123 132 213 231 312 321
3	

## BÀI 8. HOÁN VỊ NGƯỢC

Cho số nguyên dương  $N$ . Nhiệm vụ của bạn là hãy liệt kê tất cả các hoán vị của  $1, 2, \dots, N$  theo thứ tự ngược. Ví dụ với  $N = 3$  ta có kết quả: 321, 312, 231, 213, 132, 123.

Input:

- Dòng đầu tiên đưa vào số lượng test  $T$ .
- Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test là một số tự nhiên  $N$  được viết trên một dòng.
- $T, n$  thỏa mãn ràng buộc:  $1 \leq T, N \leq 10$ .

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input	Output
2	21 12
2	321 312 231 213 132 123
3	

## BÀI 9. MÃ GRAY 1

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 bit. 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  $n$ .

Input:

- Dòng đầu tiên là số lượng test  $T$ .
- $T$  dòng kế tiếp ghi lại mỗi dòng một test. Mỗi test là một số tự nhiên  $n$ .
- $T, n$  thỏa mãn ràng buộc:  $1 \leq T, n \leq 10$ .

Output:

- Đưa ra kết quả mỗi test theo từng dòng.



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

