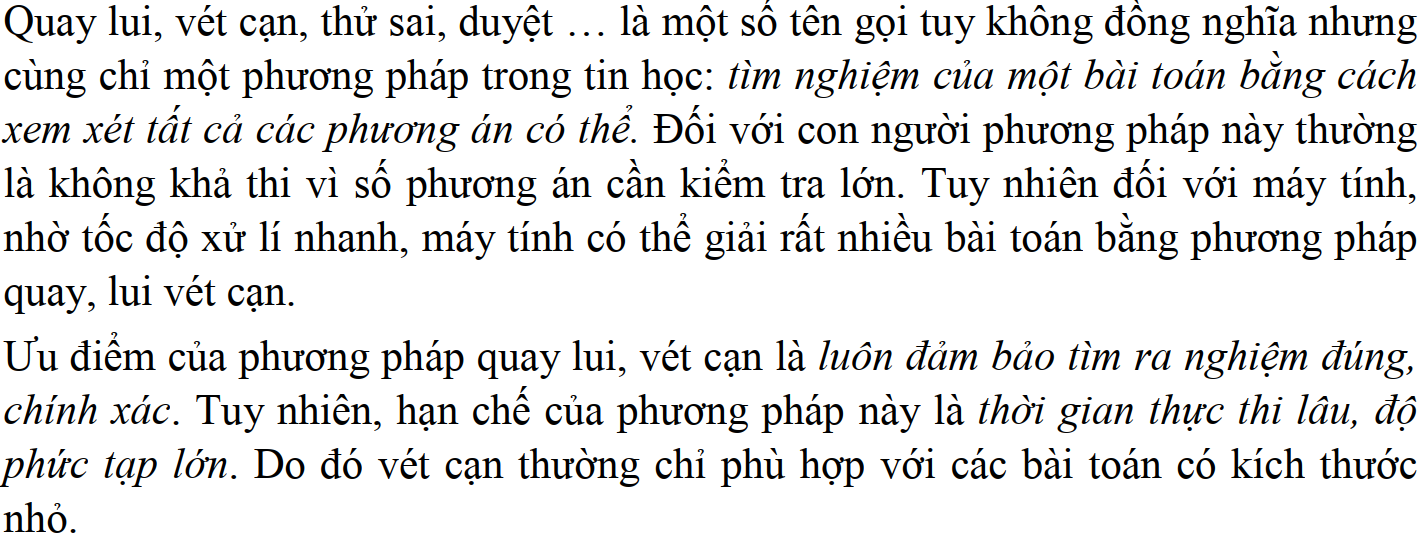
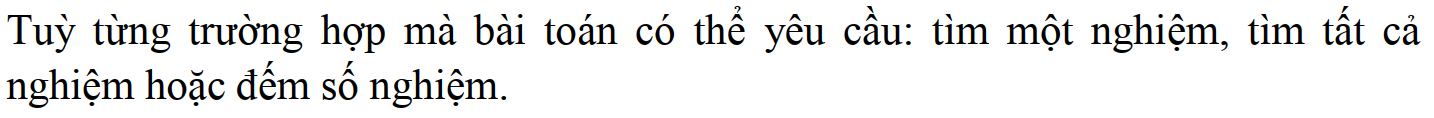
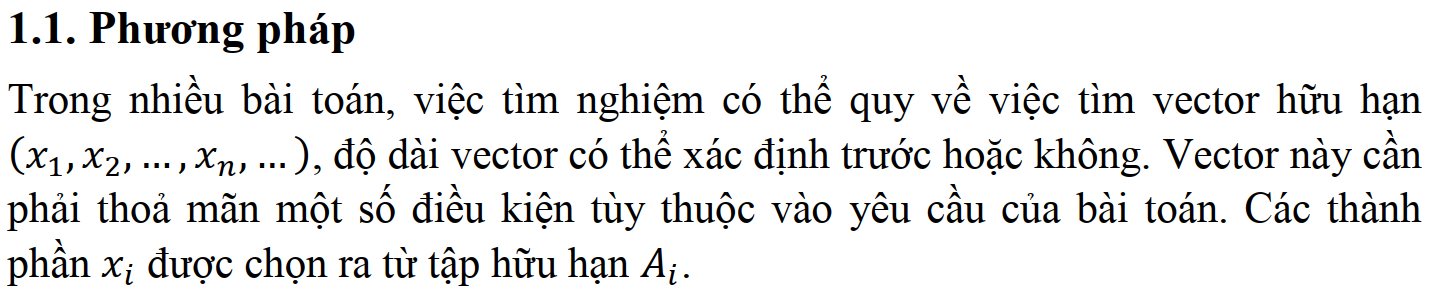
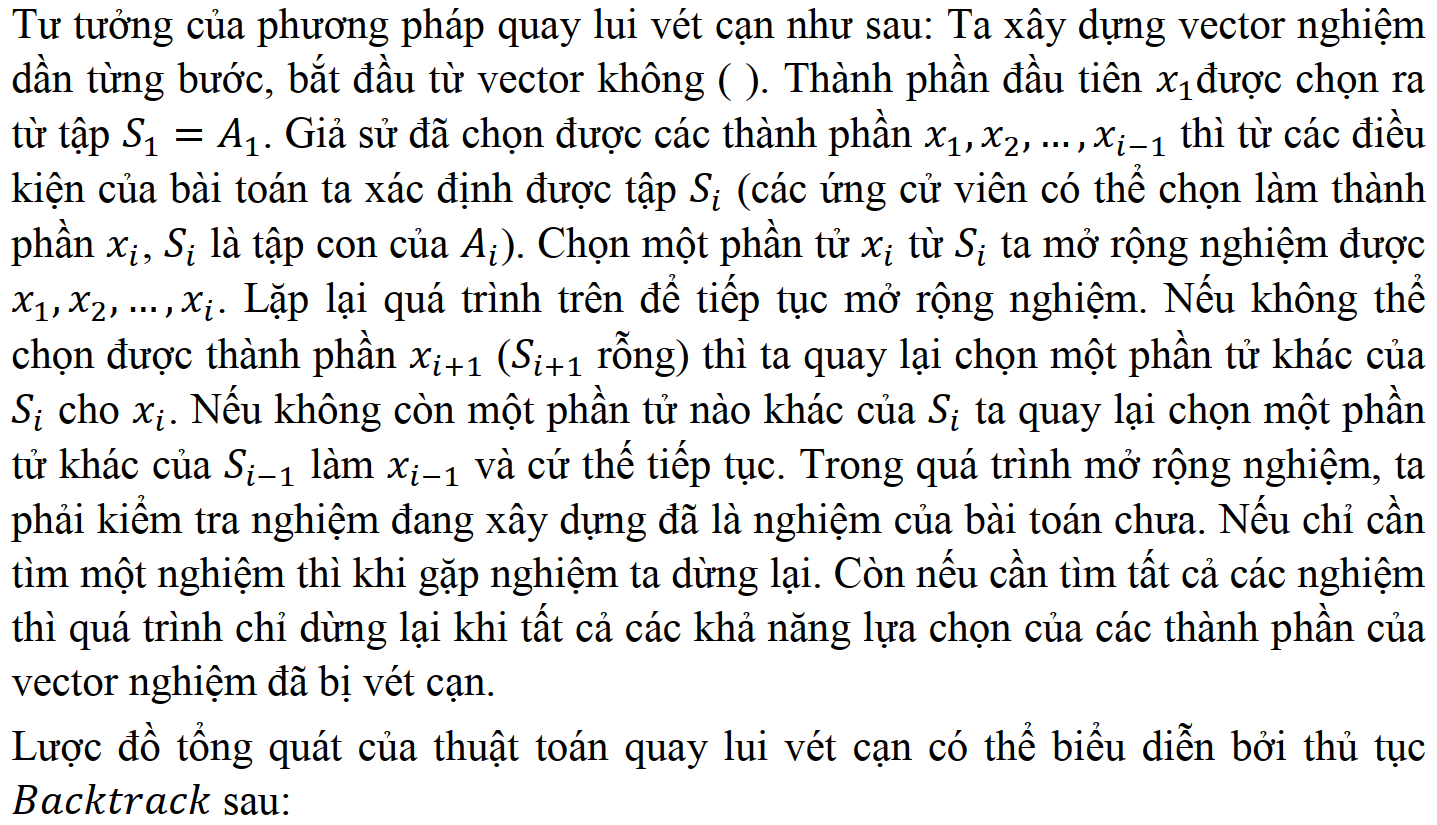
**Quay lui (Backtracking)**







void Backtrack;

{

S1=A1;

k=1;

while (k>0)

{

while ( Sk != rỗng)

{

<chọn xk  thuộc Si>;

Sk = Sk – { xk};

if (x1, x2, …,xk) là nghiệm thì đưa ra nghiệm;

k+=1;

<xác định Sk>;

}

k=k-1; // Quay lui

}

}

Trên thực tế thuật toán quay lui vét cạn thường được dùng bằng mô hình đệ quy như sau:

void Backtrack(i);

{

<Xác định Si>;

for( xi thuộc Si)

{

<ghi nhận thành phần thứ i>;

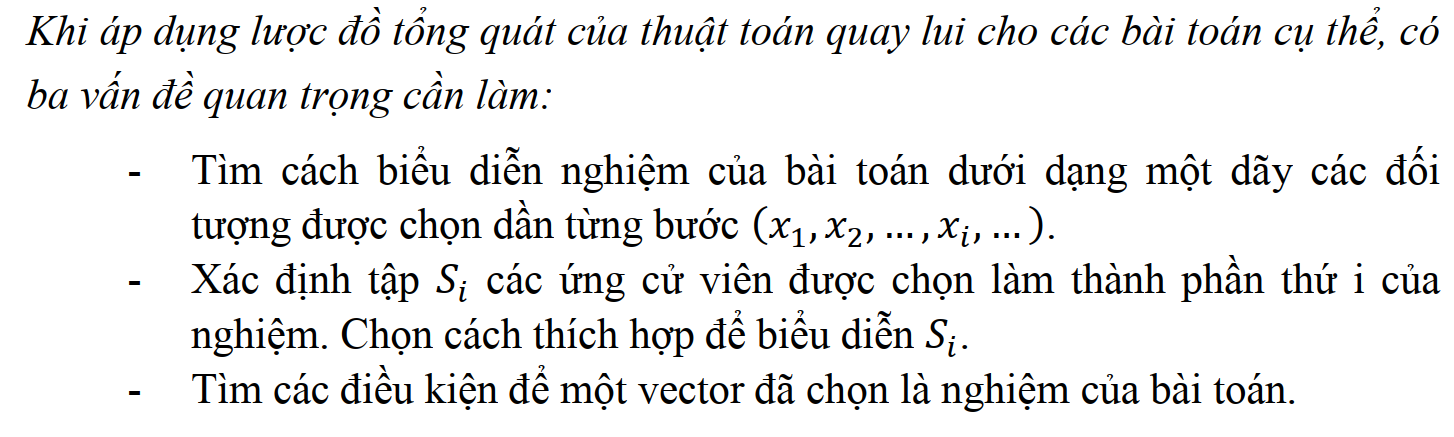
if (tìm thấy nghiệm) thì (đưa ra nghiệm);

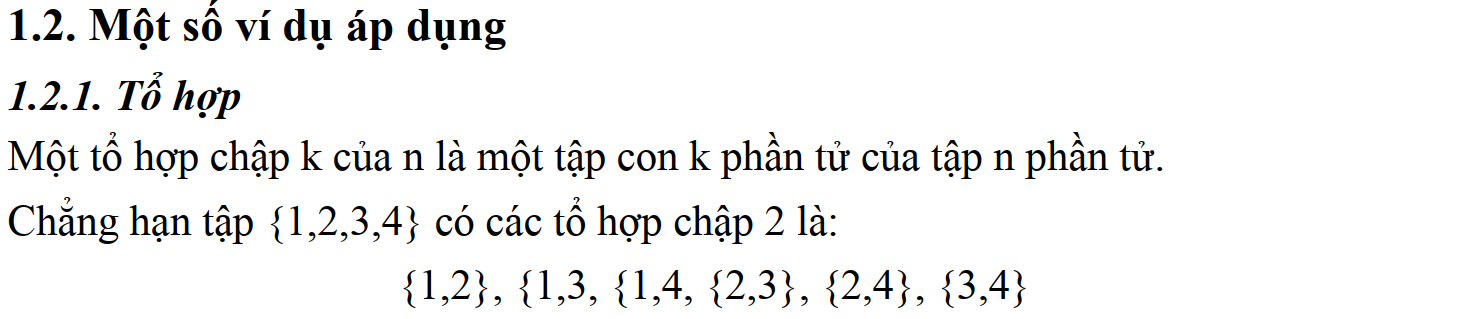
else Backtrack(i+1);

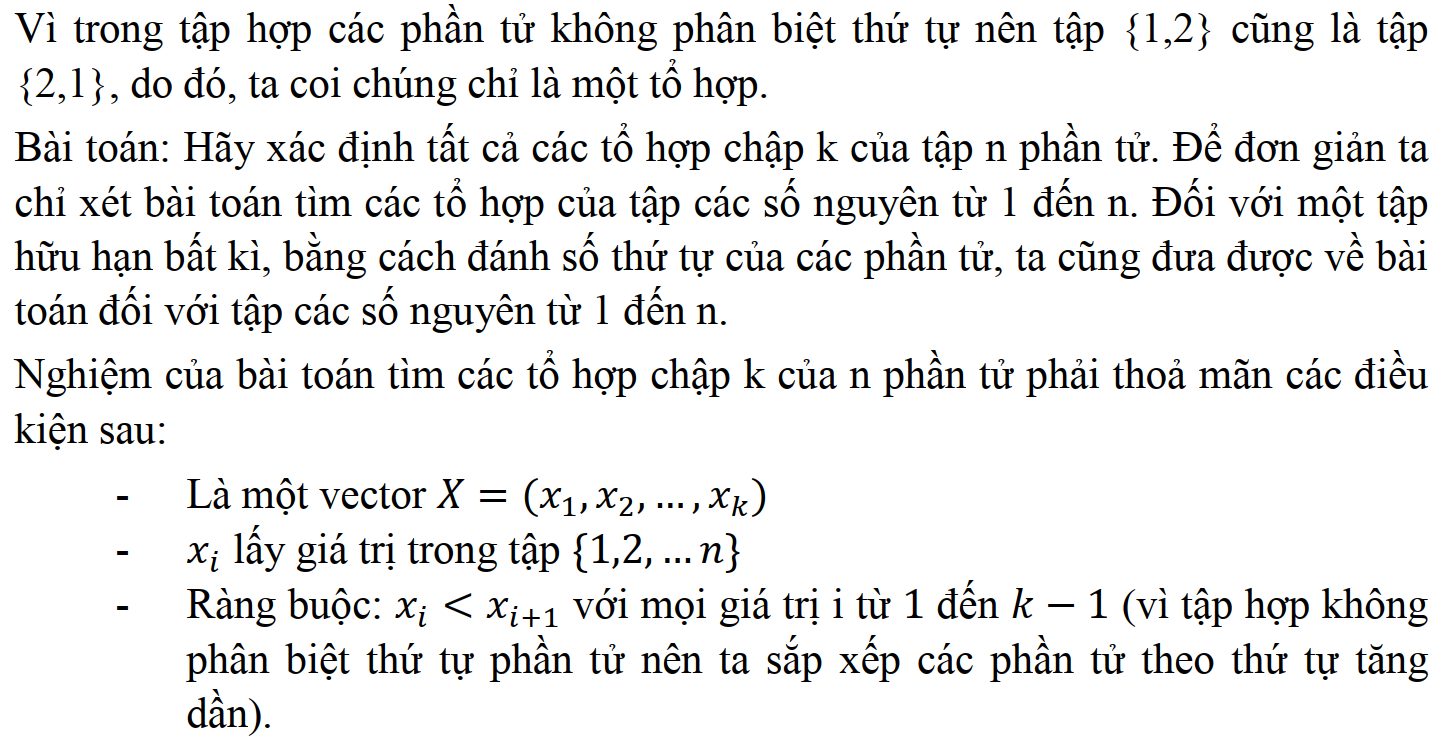
<loại thành phần i>;

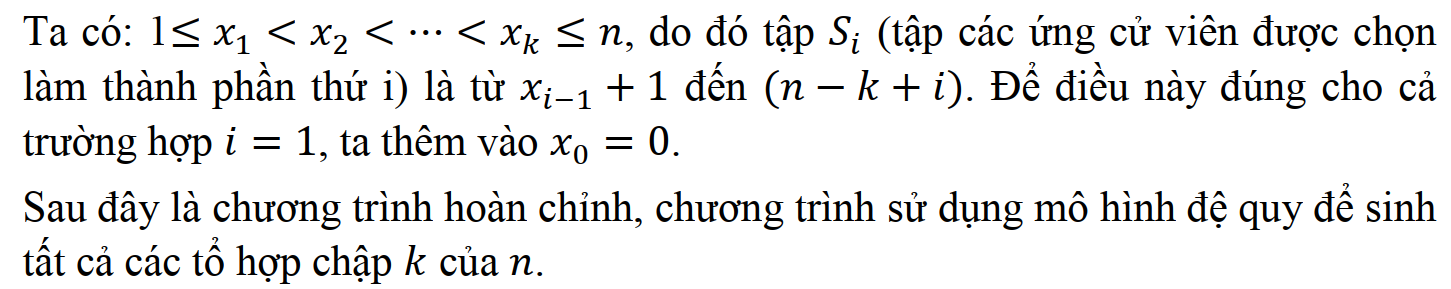
}

}









#include <bits/stdc++.h>

using namespace std;

long long x[100001],n,k;

void ghinghiem(long long x[])

{

for(long long i=1; i<=k;i++)

cout<< x[i]<<" ";

cout<<endl;

}

void tohop(long long i)

{

for(long long j=x[i-1]+1 ; j<=n-k+I ; j++)

{

x[i]=j;

if (i==k) ghinghiem(x);

else tohop(i+1);

}

}

int main()

{

cout << "Nhap n,k: ";

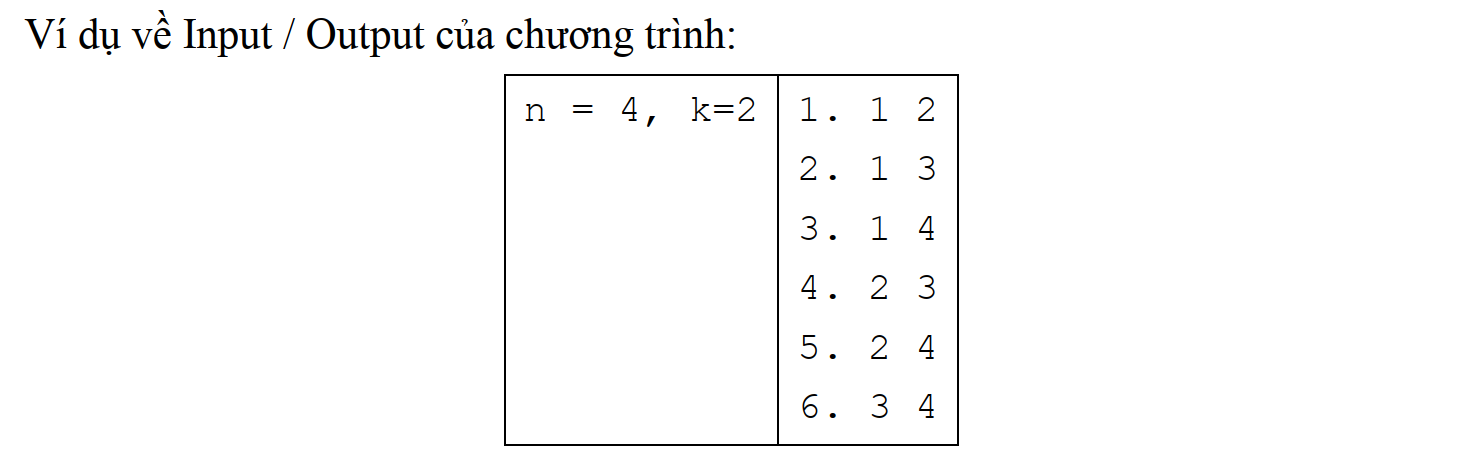
cin>>n>>k;

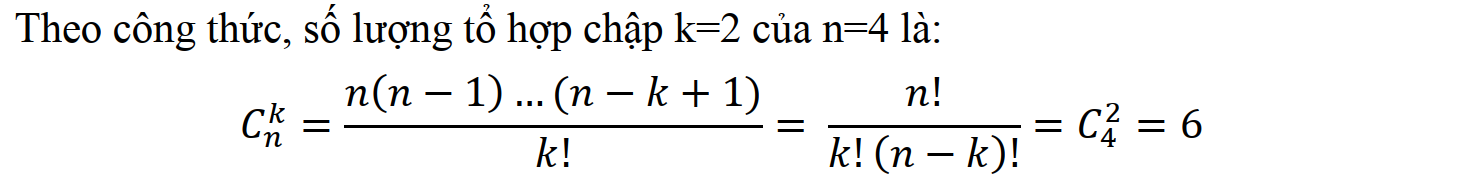
x[0]=0;

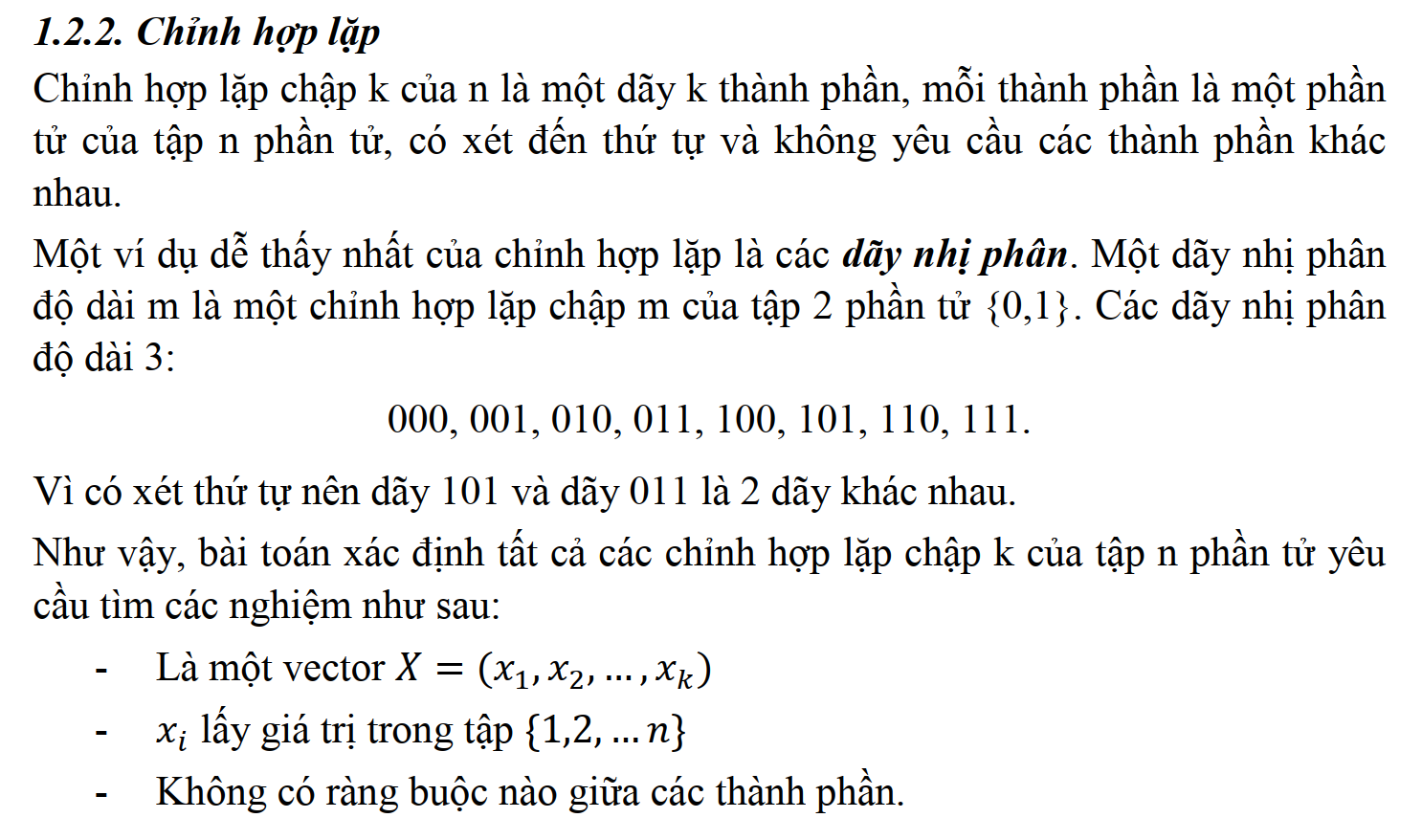
tohop(1);

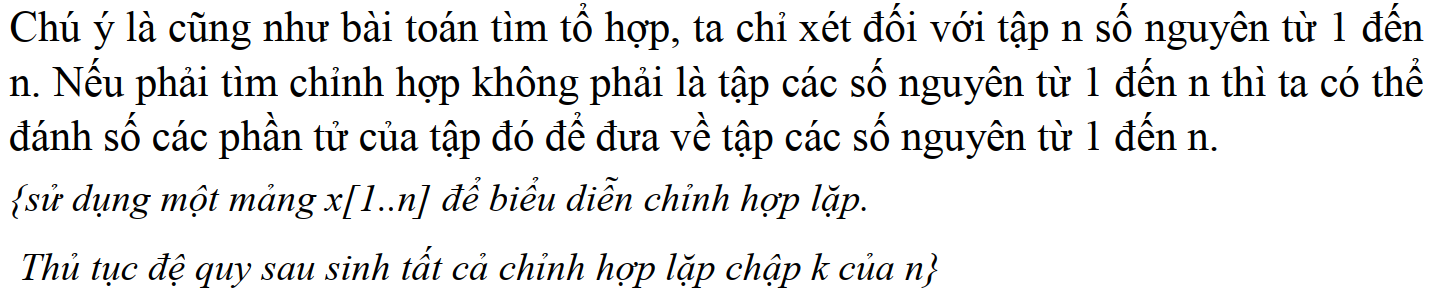
return 0;

}









void CHINHHOPLAP( long long i);

{

for ( long long j=1 ; j<=n; j++)

{

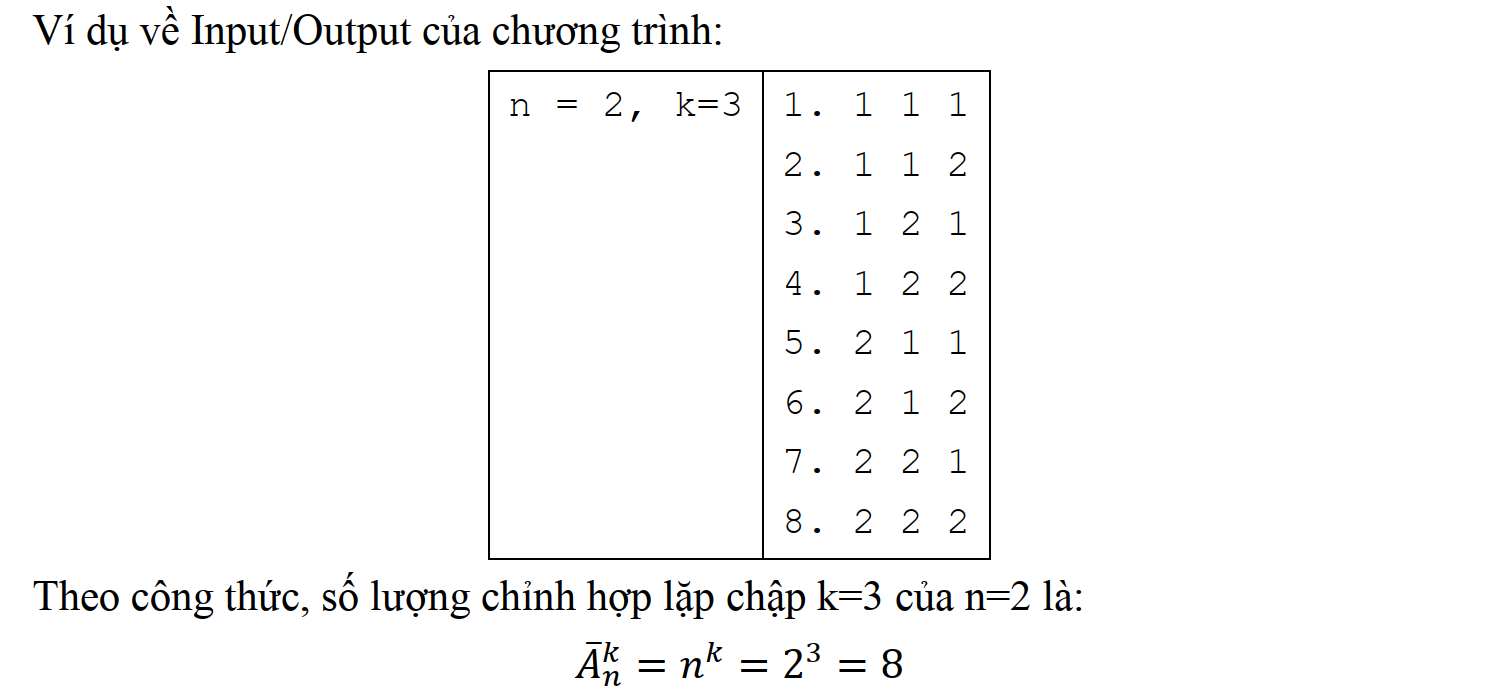
X[i] = j;

if ( i==k ) ghinghiem(x);

else CHINHHOPLAP(i+1);

}

}



Chương trình hoàn chỉnh sử dụng đệ quy sinh tất cả các chỉnh hợp lặp chập k của n:

#include <bits/stdc++.h>

using namespace std;

long long x[100001],n,k;

void ghinghiem(long long x[])

{

for ( long long i=1; i<=k;i++)

cout<< x[i]<<" ";

cout<<endl;

}

void chinhhoplap(long long i)

{

for(long long j=1;j<=n;j++)

{

x[i]=j;

if (i==k) ghinghiem(x);

else chinhhoplap(i+1);

}

}

int main()

{

cout << "Nhập n,k: ";

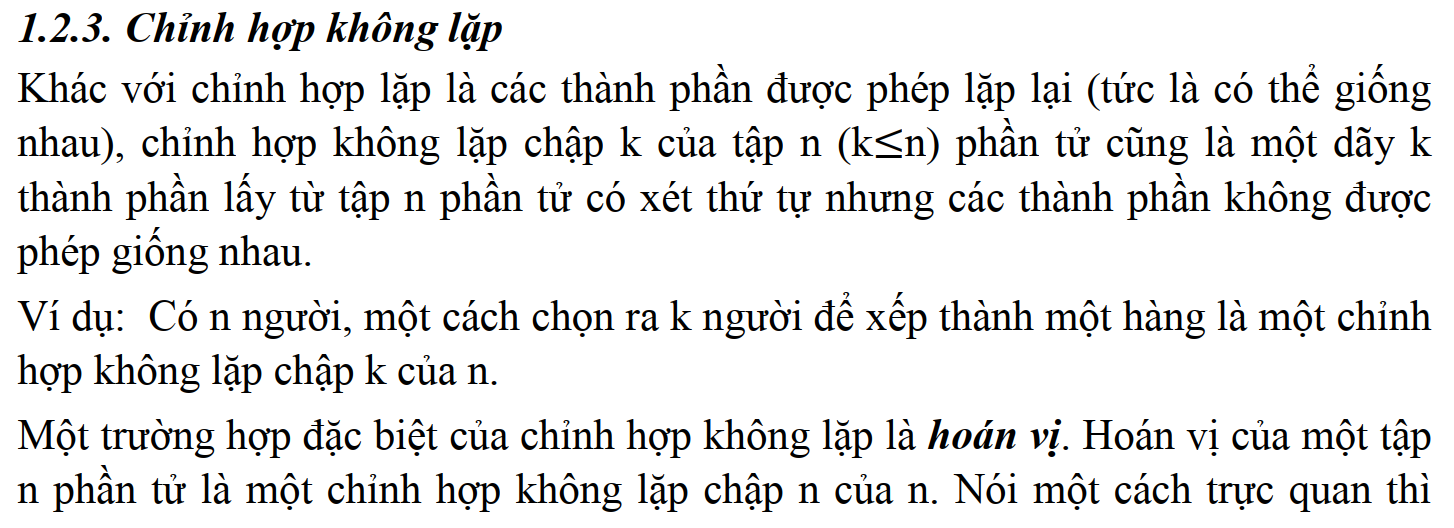
cin>>n; cin>>k;

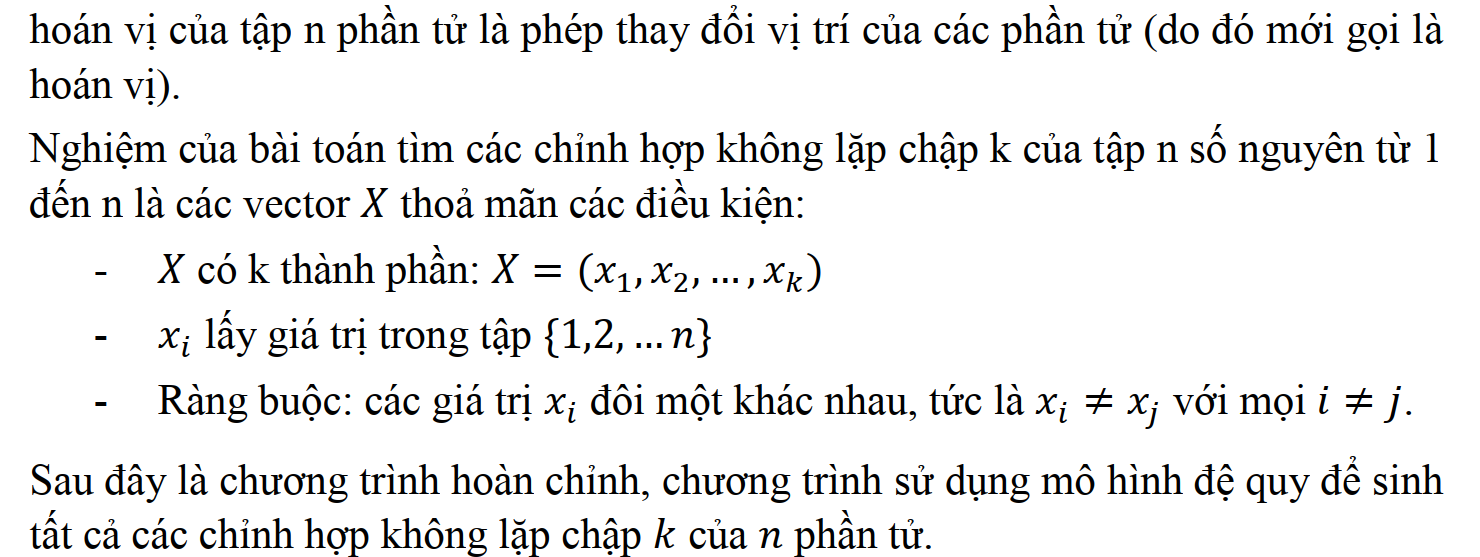
x[0]=0;

chinhhoplap(1);

cout<<d; return 0;

}





#include <bits/stdc++.h>

using namespace std;

long long x[100001],n,k,d[100001]={0};

void ghinghiem(long long x[])

{

for(long long i=1; i<=k;i++)

cout<< x[i]<<" ";

cout<<endl;

}

void chinhhopkhonglap(long long i)

{

for(long long j=1;j<=n;j++)

{

if (d[j]==0)

{

x[i]=j;

d[j]=1;

if (i==k) ghinghiem(x);

else chinhhopkhonglap(i+1);

d[j]=0;

}

}

}

int main()

{

cout << "Nhập n,k: ";

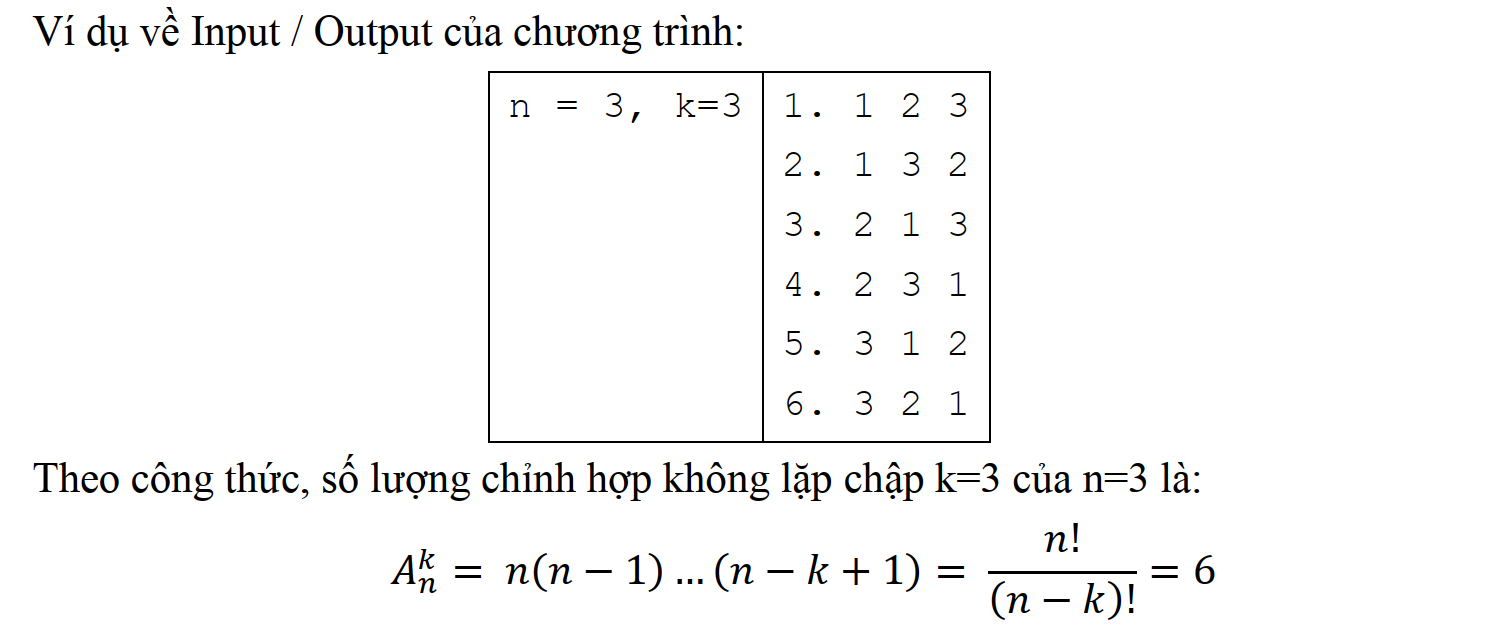
cin>>n;

cin>>k;

chinhhopkhonglap(1);

return 0;

}

BÀI TẬP

Bài 1. Cho danh sách tên của n (n<=10) học sinh (các tên đôi một khác nhau) và một số nguyên dương k (k<=n). Hãy liệt kê tất cả các cách chọn k học sinh trong n học sinh.

Dữ liệu vào:

* Dòng thứ 1 chứa hai số nguyên dương n, k
* N dòng tiếp theo mỗi dòng là tên một học sinh

Kết quả: Đưa ra các cách chọn k học sinh trong n học sinh.

Ví dụ:

|  |  |
| --- | --- |
| **hocsinh.inp** | **hocsinh.out** |
| 4 2  An  Binh  Hong  Minh | An Binh  An Hong  An Minh  Binh Hong  Binh Minh  Hong Minh |

Bài 2. Một dãy nhị phân độ dài n (n<=10) là một dãy x=x1x2…xn trong đó xi = {0;1}, i=1, 2, …, n. Hãy liệt kê tất cả các dãy nhị phân dộ dài n.

Dữ liệu vào:

* Chứa số nguyên dương n

Kết quả:

* Các dãy nhị phân độ dài n

Ví dụ:

|  |  |
| --- | --- |
| **nhiphan.inp** | **nhiphan.out** |
| 3 | 000  001  010  011  100  101  110  111 |

Bài 3. Cho xâu s (độ dài không vượt quá 10) chỉ gồm các kí tự từ ‘A’ đến ‘Z’ (các kí tự trong xâu s đôi một khác nhau). Hãy liệt kê tất cả các hoán vị khác nhau của xâu s.

Dữ liệu vào:

* Xâu s

Kết quả:

* Các hoán vị khác nhau của s

Ví dụ:

|  |  |
| --- | --- |
| **hoanvi.inp** | **hoanvi.out** |
| XYZ | XYZ  XZY  YXZ  YZX  ZXY  ZYX |

Bài 4. Cho số nguyên dương n (n<=20) , hãy liệt kê tất cả các xâu độ dài n chỉ gồm 2 kí tự ‘A’ hoặc ‘ B’ mà không có hai kí tự ‘B’ nào đứng cạnh nhau.

Dữ liệu vào:

* Chứa số nguyên dương n

Kết quả:

* Các xâu kí tự độ dài n

Ví dụ:

|  |  |
| --- | --- |
| **XAUAB.inp** | **XAUAB.out** |
| 4 | AAAA  AAAB  AABA  ABAA  ABAB  BAAA  BAAB  BABA |

Bài 5. Cho dãy số A gồm N (N<=10) số nguyên a1, a2, …, an và một số nguyên dương k (1<k<n). Hãy đưa ra một cách chia dãy số thành k nhóm mà các nhóm có tổng bằng nhau. Biết rằng luôn tồn tại cách chia.

Dữ liệu vào:

* Dòng thứ 1 chứa hai số nguyên dương n, k.
* Dòng tiếp theo chứa dãy số nguyên

Kết quả: Các nhóm có tổng bằng nhau.

Ví dụ:

|  |  |
| --- | --- |
| **nhom.inp** | **nhom.out** |
| 5 3  1 4 6 9 10 | 4 6  1 9  10 |

Bài 6. Cho số nguyên dương n (n<=10), liệt kê tất cả các cách khác nhau đặt n dấu ngoặc mở và n dấu ngoặc đóng đúng đắn.

Dữ liệu vào:

* Chứa số nguyên dương n

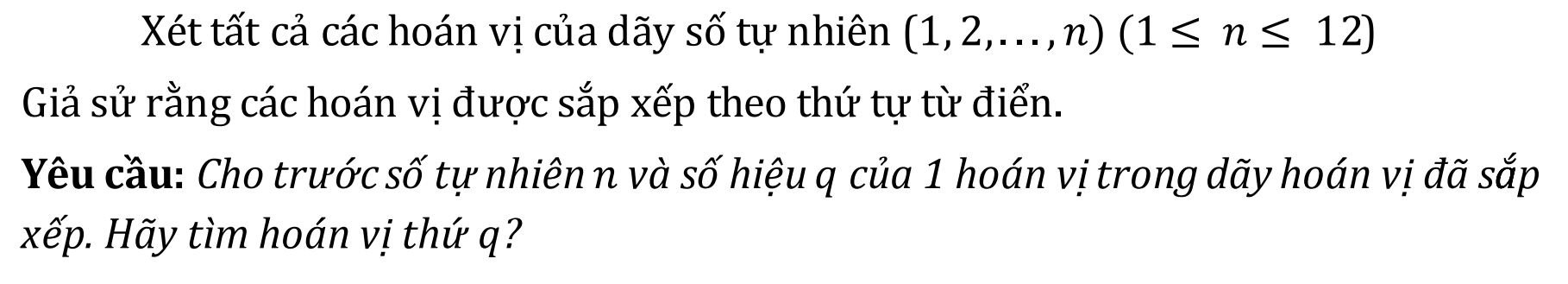
Kết quả:

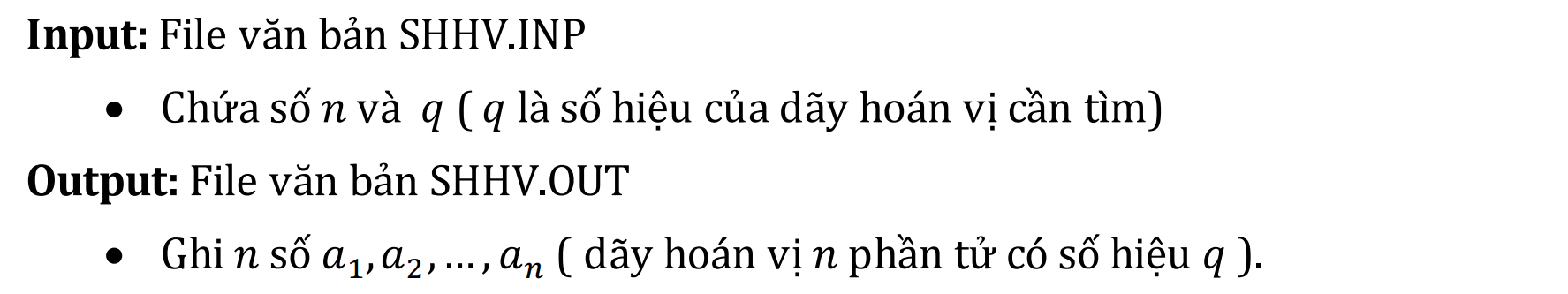
* Các cách đặt ngoặc đúng

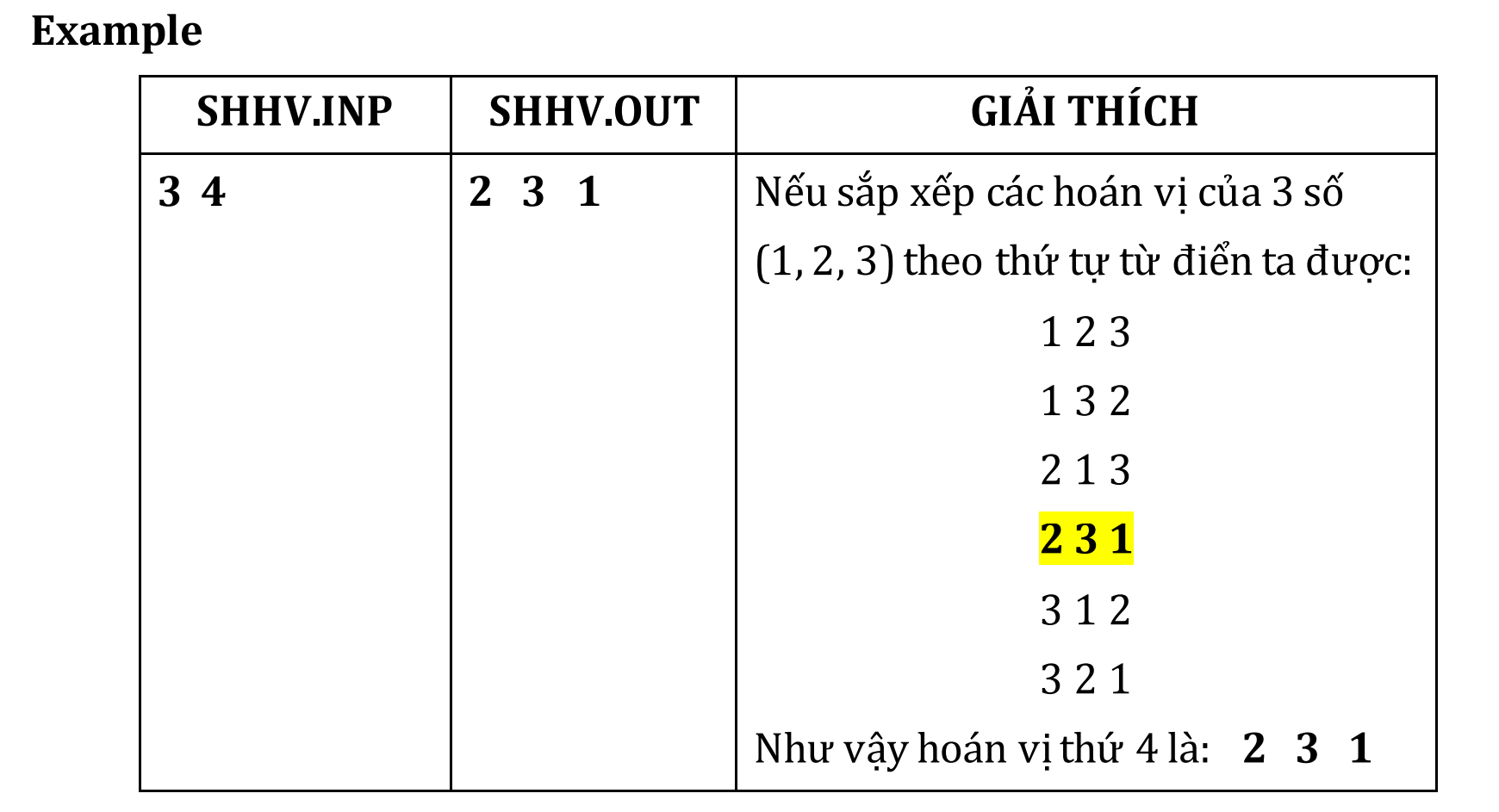
Ví dụ:

|  |  |
| --- | --- |
| **ngoac.inp** | **ngoac.out** |
| 3 | ((()))  (()())  (())()  ()(())  ()()() |

Bài 7.







Bài 8. Một hoán vị của {a1, a2, …, an } được gọi là hoán vị đầy đủ nếu ai ≠ I, với mọi i = 1, 2, …, n.

Yêu cầu: Cho một số nguyên n, tìm tất cả các hoán vị đầy đủ của tập {a1, a2, …, an }.

Dữ liệu vào: Gồm 1 số nguyên dương n.

Kết quả:

* Mỗi dòng là một hoán vị tìm được
* Dòng cuối cùng là số lượng các hoán vị.

Ví dụ:

|  |  |
| --- | --- |
| **hvdd.inp** | **hvdd.out** |
| 3 | 2 3 1  3 1 2 |