



PHƯƠNG PHÁP SINH (GENERATION)

```
Project
main.py x bai72.out x bai72.inp x
1  fi = open("bai72.inp", "r")
2  fo = open("bai72.out", "w")
3  t = tuple()
4  # Doc du lieu tu file luu vao tuple, moi nguoi tren mot dong
5  for i in fi:
6      st = i.split()
7      t = t + ([st[0], float(st[1])],)
8  print(t, file = fo)
9  # sap xep tuple theo Ten tang dan
10 tt = sorted(t, reverse= False)
11 print(tt, file = fo)
12 # sap xep tuple theo Diem giam dan
13 tt = sorted(t, key = lambda t : t[1], reverse= True)
14 print(tt, file = fo)
15 # sap xep tuple theo len(tu) giam dan
16 tt = sorted(t, key= lambda t : len(t[0]), reverse= True)
17 print(tt, file = fo)
```



LỜI MỞ ĐẦU

Nhằm cung cấp tài liệu, trang bị kỹ năng tự học, tự nghiên cứu cho học sinh và giáo viên, hình thành và phát triển kỹ năng cho người học; Xây dựng hệ thống bài tập củng cố kiến thức bộ môn theo chương trình môn Tin học; Bổ sung kiến thức nâng cao, bồi dưỡng học sinh giỏi môn Tin học bậc THCS và bậc THPT, tác giả sưu tầm và biên soạn cuốn tài liệu: “**Phương pháp sinh (generation)**”.

Giải mã và code chương trình trong “**Phương pháp sinh (Generation)**” được viết bởi ngôn ngữ lập trình **Pascal**, **C++** và **Python**. Nội dung gồm **51** bài tập của các chuyên đề được biên soạn từ dễ đến khó, tạo điều kiện thuận lợi cho việc tự học, tự nghiên cứu. Với mỗi đơn vị kiến thức, có các bài tập, hướng dẫn thuật toán và code chương trình giúp người học nhanh chóng làm quen định dạng, quy cách làm bài thi; Người học được hướng dẫn thuật toán có mức độ hoàn thiện bài khác nhau, phát triển kỹ năng “làm mịn chương trình”.

Trong thời gian tới, tác giả viết tiếp tài liệu mới “**Quy hoạch động**” với các chuyên đề nâng cao liên quan đến kỳ thi học sinh giỏi các cấp. Rất mong các bạn đón đọc.

Tài liệu này được biên soạn rất tỉ mỉ nhưng không tránh được những thiếu sót, tác giả rất mong được sự đóng góp nội dung, phản hồi của bạn đọc vào mail: dtthanh.c3ntt@yenbai.edu.vn hoặc tại diễn đàn <https://antit.forumvi.com/>.

Tác giả

BÀI TOÁN LIỆT KÊ

CHƯƠNG 1. LIỆT KÊ DÃY NHỊ PHÂN ĐỘ DÀI N VÀ BÀI TOÁN ÁP DỤNG	6
1. DQBN0.....	6
2. DQBN01.....	7
3. DQBN02.....	8
4. DQBN03.....	10
5. DQBN04.....	11
6. DQBN05.....	13
7. DQBN06.....	14
8. DQBN7.*	16
9. DQBN8.*	19
10. DQBN9.*	21
11. DQBN10.*	22
12. DQBN11.*	25
13. DQBN12.*	30
14. DQBN13.*	35
15. DQBN14.*	39
16. DQBN15.*	44
17. DQBN16.*	49
18. NGỌC LỤC BẢO	53
19. MÔ ĐUN M.....	57
20. ĐẦU TƯ CHỨNG KHOÁN	58
21. CHUỖI NHỊ PHÂN	59
LIỆT KÊ CÁC TẬP CON K PHẦN TỬ CỦA TẬP N PHẦN TỬ.....	61
22. DQCKN1.* Tập con K phần tử.....	61
23. DQCKN2.* Tập con K có tổng bằng S	64
24. DQCKN3.* Tập con K có tổng nguyên tố	68
25. DQCKN4.* Tập con K có tổng là số Palindrom.....	73
26. DQCKN51.* Tập con K chưa min và max.....	76
27. DQCKN52.* Tập con K có độ chênh lệch nhỏ nhất.....	83
28. DQCKN6.* Bài toán cái túi 2	88
29. DQCKN7.* Tập con K có tổng lớn nhất	94
30. DQCKN8.* Bài toán cái túi 3	101
31. DQCKN9.* Tập con K có tổng bằng P	108

LIỆT KÊ HOÁN VỊ.....	115
32. DQHV1.* Liệt kê hoán vị.....	115
33. DQHV2.* Số thứ tự của hoán vị.....	121
34. DQHV3.* Hoán vị thứ K.....	125
35. DQHV4.* Hoán vị kế tiếp	129
36. DQHV5.*	130
37. DQHV6.*	132
THUẬT TOÁN QUAY LUI	133
38. QLBIN1.*	133
39. QLBIN2.*	134
40. QLCKN1.*	135
41. QLHV1.*	137
42. QLCHKN.*	139
43. QLPTS.*	141
44. QLPTSPR.*	143
45. QLQHAU.*	144
46. QLBIN3.*	144
47. QLHVHT.*	144
48. QLCHLAP.*	145
49. QLMADT.*	145
50. QLDDT.*	145
51. QLTURI.*	145

CHƯƠNG 1. LIỆT KÊ DÃY NHỊ PHÂN ĐỘ DÀI N VÀ BÀI TOÁN ÁP DỤNG

Low:

1. DQBN0.

Liệt kê dãy nhị phân có độ dài N.

Ví dụ: N = 3, các dãy nhị phân thu được là:

0 0 0

0 0 1

0 1 0

0 1 1

1 0 0

1 0 1

1 1 0

1 1 1

Thuật toán:

- Chừng nào $N > 0$ làm:
 - Đi từ phải sang tìm phần tử bằng 0 đầu tiên và dừng lại.
 - Thay phần tử đó bằng 1.
 - Giữ nguyên các phần tử đứng trước nó.
 - Thay các phần tử đứng sau nó bằng 0.
- Độ phức tạp thuật toán: $O(2^N)$

Code C++	Code Python
<pre>#include<bits/stdc++.h> using namespace std; int a[34]; int n; void sinh(){ int i,j; do{ for(i=1; i<=n; i++) cout<<a[i]; cout<<endl; i=n; while(i>0 && a[i]==1) i--; if(i>0){ a[i]=1; for(j=i+1; j<=n; j++) a[j]=0; } }while(i!=0);</pre>	<pre>n = int(input("Nhập n: ")) x = n * [0] def _print(x): tmp = "" for i in x: tmp += str(i) return tmp def thu(i): for j in range(0, 2): x[i] = j if i == n - 1: print(_print(x)) else: thu(i + 1) thu(0)</pre>

Code C++	Code Python
<pre> } int main(){ ios_base::sync_with_stdio(0); cin.tie(0); cout.tie(0); cin>>n; for(int i=1; i<=n; i++) a[i]=0; sinh(); return 0; } </pre>	

2. DQBN01.

Liệt kê dãy nhị phân không có 3 số 1 đứng cạnh nhau.

Thuật toán:

Lần lượt sinh ra các dãy nhị phân.

Với mỗi cấu hình thu được, ta kiểm tra có “3 số 1 đứng cạnh nhau” hay không?

Điều kiện: $a[i]+a[i+1]+a[i+2] = 3$ là có 3 số 1 đứng cạnh nhau.

Nếu cấu hình mà không có 3 phần tử nào thỏa mãn điều kiện trên thì ta đưa cấu hình đó ra.

Code C++	Code Python
<pre> #include<bits/stdc++.h> using namespace std; int a[34]; int n; void xuat(){ for(int i=1; i<=n-2; i++) if(a[i]==1 && a[i+1]==1 && a[i+2]==1) return; for(int i=1; i<=n; i++) cout<<a[i]; cout<<"\n"; } void sinh(){ int i,j; for(i=1; i<=n; i++) a[i]=0; do{ xuat(); i=n; while(i>0 && a[i]==1) i--; </pre>	<pre> n = int(input("Nhập n: ")) x = n * [0] def _xuli(): for p in range(0, n-2): if (x[p]==1) and (x[p+1] == 1) and (x[p+2]) == 1 : return print(x) def thu(i): for j in range(0, 2): x[i] = j if i == n - 1: _xuli() else: thu(i + 1) thu(0) </pre>

Code C++	Code Python
<pre> if(i>0){ a[i]=1; for(j=i+1; j<=n; j++) a[j]=0; } }while(i!=0); } int main(){ ios_base::sync_with_stdio(0); cin.tie(0); cout.tie(0); cin>>n; sinh(); return 0; } </pre>	

3. DQBN02.

Liệt kê dãy nhị phân có số lượng số 0 bằng số lượng số 1.

Thuật toán:

Lần lượt sinh các dãy nhị phân.

Với mỗi cấu hình ta đi đếm số lượng số 0 lưu vào res.

Khi đó số lượng số 1 là: $N - res$.

Vậy điều kiện thoả mãn là $res = N - res$ hay $2*res = N$.

Ta có thể thêm nhánh cận: Nếu N là số lẻ thì vô nghiệm.

Code C++	Code Python
<pre> #include <bits/stdc++.h> #define nmax 100007 using namespace std; int a[nmax]; int n,i; void xuli() { int d1=0,d2=0; </pre>	<pre> n = int(input("Nhập n: ")) x = n * [0] def _xuli(x): res=0 for p in range(0, n): if (x[p]==1): res=res+1 if res==n-res: print(x) def thu(i): for j in range(0, 2): x[i] = j if i == n - 1: _xuli(x) else: thu(i + 1) thu(0) </pre>

Code C++	Code Python
<pre> for (int i=1 ; i<=n ; i++) { if (a[i]==0) d1++; else d2++; } if (d1==d2) { for (int i=1;i<=n;i++) cout << a[i] << " "; cout << endl; } } void thu () { for (int i=1;i<=n;i++) a[i]=0; int i; do { xuli(); i = n; while ((i>0) && (a[i]==1)) i--; if (i>0) { a[i] = 1; for (int j=i+1;j<=n;j++) a[j]=0; } } while (i > 0); } int main() { //freopen("sinhlietke.inp","r",stdin); </pre>	

Code C++	Code Python
<pre>//freopen("sinhlietke.out","w",stdout); cin >> n ; if (n%2==0) thu(); else cout << -1; return 0; }</pre>	

4. DQBN03.

Liệt kê dãy nhị phân có đúng K số 1.

Thuật toán:

Lần lượt sinh các dãy nhị phân.

Với mỗi cấu hình ta đi đếm số lượng số 1 lưu vào res.

Nếu res = K thì đưa cấu hình đó ra.

Code C++	Code Python
<pre>#include <bits/stdc++.h> #define nmax 100007 using namespace std; int a[nmax]; int n,i,k; void xuli() { int d1=0,d2=0; for (int i=1 ; i<=n ; i++) { if (a[i]==1) d1++; if (d1 > k) return; } if (d1==k) { for (int i=1;i<=n;i++) cout << a[i]; } }</pre>	<pre>n = int(input("Nhập n: ")) k=int(input("nhap so luong chu so 1:")) x = n * [0] def _xuli(x): res=0 for p in range(0, n): if (x[p]==1): res=res+1 if res==k: print(x) def thu(i): for j in range(0, 2): x[i] = j if i == n - 1: _xuli(x) else: thu(i + 1) thu(0)</pre>

Code C++	Code Python
<pre> cout << endl; } } void thu () { for (int i=1;i<=n;i++) a[i]=0; int i; do { xuli(); i = n; while ((i>0) && (a[i]==1)) i--; if (i>0) { a[i] = 1; for (int j=i+1;j<=n;j++) a[j]=0; } } while (i > 0); } int main() { //freopen("sinhlietke.inp","r",stdin); //freopen("sinhlietke.out","w",stdout); cin >> n >> k; thu(); return 0; } </pre>	

5. DQBN04.

Liệt kê dãy nhị phân có không quá K số 1.

Thuật toán:

Lần lượt sinh các dãy nhị phân.

Với mỗi cấu hình ta đi đếm số lượng số 1 lưu vào res.

Nếu $res \leq K$ thì đưa cấu hình đó ra.

Code C++	Code Python
<pre> #include <bits/stdc++.h> using namespace std; int n, k, a[25]; bool check () { int d = 0; for (int i = 1; i <= n; ++i) { if (a[i] == 1) ++d; if (d > k) return false; // nhánh cận } if (d <= k) return true; else return false; } void show () { for (int i = 1; i <= n; ++i) cout << a[i]; cout << "\n"; } void gen () { for (int i = 1; i <= n; ++i) a[i] = 0; int i; do { if (check()) show (); i = n; while ((i > 0) && (a[i] == 1)) --i; if (i > 0) { a[i] = 1; for (int j = i + 1; j <= n; ++j) a[j] = 0; } } </pre>	<pre> n = int(input("Nhập n: ")) k=int(input("nhap so luong chu so 1 nhieu nhat:")) x = n * [0] def _xuli(x): res=0 for p in range(0, n): if (x[p]==1): res=res+1 if res<=k: print(x) def thu(i): for j in range(0, 2): x[i] = j if i == n - 1: _xuli(x) else: thu(i + 1) thu(0) </pre>

Code C++	Code Python
<pre> } while (i > 0); } int main () { ios_base::sync_with_stdio(NULL); cin.tie(NULL); cout.tie(NULL); cin >> n >> k; gen(); return 0; } </pre>	

Mid: Nhập N và dãy b_1, b_2, \dots, b_N .

6. DQBN05.

Hãy đưa ra các cách chọn một số phần tử trong **dãy b** sao cho có tổng bằng K.

Thuật toán:

Sử dụng mảng $a[i] = 0/1$ tương ứng cho việc chọn/không chọn phần tử $b[i]$.

Áp dụng phương pháp sinh dãy nhị phân có độ dài N ta thu được các cấu hình trong mảng a.

Với mỗi cấu hình, ta for (1, N) nếu $a[i] = 1$ thì $b[i]$ được chọn và đẩy $b[i]$ vào tổng S.

Nếu $S = K$ thì ta duyệt For(1, N) nếu $a[i] = 1$ thì đưa $b[i]$ ra.

Ta có thể thêm nhánh cận: nếu tổng $S > K$ thì return ra ngoài.

Code C++	Code Python
<pre> #include<bits/stdc++.h> using namespace std; int n,k,sum; int a[34],b[34]; void xuat(){ sum=0; for(int i=1; i<=n; i++) { sum += a[i]*b[i]; if (sum > k) return; } if(sum==k){ for(int i=1; i<=n; i++) </pre>	<pre> n = int(input("Nhập n: ")) k=int(input("nhap so k:")) a=n*[0] for i in range(0,n): h=int(input(f"nhap ptu thu {i+1}:")) a[i]=h x = n * [0] def _xuli(x): res=0 for p in range(0, n): if (x[p]==1): res=res+a[p] if res==k: print(x) def thu(i): for j in range(0, 2): x[i] = j </pre>

Code C++	Code Python
<pre> if(a[i]==1) cout<<b[i]<<" "; cout<<"\n"; } } void sinh(){ int i,j; for(i=1; i<=n; i++) a[i]=0; do{ xuat(); i=n; while(i>0 && a[i]==1) i--; ; if(i>0){ a[i]=1; for(j=i+1; j<=n; j++) a[j]=0; } }while(i!=0); } int main(){ ios_base::sync_with_stdio(0); cin.tie(0); cout.tie(0); cin>>n>>k; for(int i=1; i<=n; i++) cin>>b[i]; sinh(); return 0; } </pre>	<pre> if i == n - 1: _xuli(x) else: thu(i + 1) thu(0) </pre>

7. DQBN06.

Hãy đưa ra các cách chọn **hiều phần tử nhất** trong dãy b sao cho có tổng $\leq K$. Đưa ra số lượng nhiều nhất các phần tử được chọn.

Thuật toán:

Sử dụng mảng $a[i] = 0/1$ tương ứng cho việc chọn/không chọn phần tử $b[i]$.

Áp dụng phương pháp sinh dãy nhị phân có độ dài N ta thu được các cấu hình trong mảng a.

Với mỗi cấu hình, ta for (I, 1, N) nếu $a[i] = 1$ thì $b[i]$ được chọn và đẩy $b[i]$ vào tổng S, tăng d – số lượng phần tử được chọn.

Nếu $S \leq K$ thì ta:

- Nếu $res < d$ thì
 - $Res = d$;
 - Lưu cấu hình $a[i]$ vào mảng $c[i]$.

Ở chương trình chính: ta đưa Res ra.

Nếu bài yêu cầu đưa ra một cách chọn thì ta for(I,1,N) nếu $c[i] = 1$ thì đưa $b[i]$ ra.

Code C++

```
#include<bits/stdc++.h>
using namespace std;
int n,k,maxx=0,c,sum,m=0; int a[34],b[34],d[34],f[34][34];
void xuat(){
    int i,j;
    for(i=1; i<=m; i++)
        if(d[i]==maxx){
            for(j=1; j<=n; j++)
                if(f[i][j]==1) cout<<b[j]<<" ";
            cout<<"\n";
        }
}
void tim(){
    sum=0;
    for(int i=1; i<=n; i++)
        if(a[i]==1){
            sum+=b[i];
            if(sum>k) return;
        }
    c=0;
    for(i=1; i<=n; i++)
        if(a[i]==1) c++;
    if(c>=maxx){
        maxx=c; m++; d[m]=c;
        for(i=1; i<=n; i++) f[m][i]=a[i];
    }
}
```

Code C++

```

void sinh(){
    int i,j;
    for(i=1; i<=n; i++) a[i]=0;
    do{
        tim(); i=n;
        while(i>0 && a[i]==1) i--;
        if(i>0){
            a[i]=1;
            for(j=i+1; j<=n; j++) a[j]=0;
        }
    }while(i!=0);
    xuat();
}

int main(){
    ios_base::sync_with_stdio(0); cin.tie(0); cout.tie(0);
    cin>>n>>k;
    for(int i=1; i<=n; i++) cin>>b[i];
    sinh();
    return 0;
}

```

8. DQBN7.*

Hãy liệt kê ra các cách chia dãy trên thành hai nhóm sao cho tổng mỗi nhóm bằng nhau. Đưa ra hai dãy là chỉ số của các phần tử được chọn của mỗi nhóm.

Thuật toán:

Sử dụng mảng $a[i] = 0/1$ tương ứng cho việc chọn/không chọn phần tử $b[i]$.

Áp dụng phương pháp sinh dãy nhị phân có độ dài N ta thu được các cấu hình trong mảng a .

Với mỗi cấu hình, ta for $(I, 1, N)$ nếu $a[i] = 1$ thì $b[i]$ được chọn và đẩy $b[i]$ vào tổng $S1$, ngược lại đẩy vào $S2$.

Nếu $S1 = S2$ thì ta:

- for $(I, 1, N)$ nếu $a[i] = 1$ thì đưa $b[i]$ ra là cách chọn cho người thứ nhất.
- for $(I, 1, N)$ nếu $a[i] = 0$ thì đưa $b[i]$ ra là cách chọn cho người thứ hai.

Ta thêm nhánh cận:

Nếu tổng S là số lẻ thì không có cách chia.

Nếu $(S1 > S/2)$ hoặc $(S2 > S/2)$ thì return. Trong đó, S là tổng tất cả các phần tử $b[i]$.

Code C++

```

#include <bits/stdc++.h>
//minhttk332
using namespace std;

int n, k, d = 0, sum = 0, b[25];
bool a[25];

bool check () {
    int sum1 = 0;
    for (int i = 1; i <= n; ++i) {
        if (a[i]) sum1 += b[i];
    }
    if (sum1 == (sum / 2)) return true;
    else return false;
}

void show () {
    ++d;
    for (int i = 1; i <= n; ++i) if (a[i]) cout << b[i] << ' ';
    cout << '\n';
}

void gen () {
    for (int i = 1; i < n; ++i) a[i] = 0;
    a[n] = 1;
    int i;
    do {
        if (check()) show ();
        i = n;
        while ((i > 0) && (a[i] == 1)) --i;
        if (i > 0) {
            a[i] = 1;
            for (int j = i + 1; j <= n; ++j) a[j] = 0;
        }
    } while (i > 0);
}

```

Code C++

```

}

int main () {
    ios_base::sync_with_stdio(NULL);
    cin.tie(NULL); cout.tie(NULL);
    cin >> n;
    for (int i = 1; i <= n; ++i) {
        cin >> b[i];
        sum += b[i];
    }
    if (sum % 2) cout << -1;
    else {
        gen();
        if (d == 0) cout << -1;
    }
    return 0;
}

```

```

#include<bits/stdc++.h>
using namespace std;
int a[34],b[34]; int n,sum=0,w,tmp;
void xuat(){
    tmp=0;
    for(int i=1; i<=n; i++){
        tmp+=a[i]*b[i];
        if(tmp>w) return;
    }
    if(tmp==sum/2){
        cout<<"nhom 1: ";
        for(int i=1; i<=n; i++)
            if(a[i]==1) cout<<i<<" ";
        cout<<"\nnhom 2: ";
        for(int i=1; i<=n; i++)
            if(a[i]==0) cout<<i<<" ";
        cout<<"\n";
    }
}
}

```

Code C++

```

void sinh(){
    int i;
    for(i=1; i<=n; i++) a[i]=0;
    do{
        xuat(); i=n;
        while(i>0 && a[i]==1) i--;
        if(i>0){
            a[i]=1;
            for(int j=i+1; j<=n; j++) a[j]=0;
        }
    }while(i!=0);
}

int main(){
    ios_base::sync_with_stdio(0); cin.tie(0); cout.tie(0);
    cin>>n;
    for(int i=1; i<=n; i++){
        cin>>b[i]; sum+=b[i];
    }
    w=sum/2;
    if(sum%2==0) sinh();
    return 0;
}

```

9. DQBN8.*

Hãy đưa ra cách chọn nhiều phần tử nhất có tổng chia hết cho K.

Thuật toán:

Sử dụng mảng $a[i] = 0/1$ tương ứng cho việc chọn/không chọn phần tử $b[i]$.

Áp dụng phương pháp sinh dãy nhị phân có độ dài N ta thu được các cấu hình trong mảng a.

Với mỗi cấu hình, ta for (1, N) nếu $a[i] = 1$ thì $b[i]$ được chọn và đẩy $b[i]$ vào tổng S, đếm số lượng phần tử được chọn lưu vào d.

Nếu S chia hết cho K thì ta:

Nếu $d > res$ thì

- Cập nhật $res = d$;
- Lưu cấu hình $a[i]$ vào $c[i]$.

Đưa res ra và các cấu hình thoả mãn ra.

Code C++

```

#include<bits/stdc++.h>
using namespace std;
int n,k,maxx=0,c; int a[34],b[34],d[34]; long long sum;
void xuat(){
    for(int i=1; i<=n; i++)
        if(d[i]==1) cout<<b[i]<<" ";
}
void tim(){
    sum=0;
    for(int i=1; i<=n; i++) sum+=1LL*a[i]*b[i];
    if(sum%k==0){
        c=0;
        for(int i=1; i<=n; i++)
            if(a[i]==1) c++;
        if(c>=maxx){
            maxx=c;
            for(int i=1; i<=n; i++) d[i]=a[i];
        }
    }
}
void sinh(){
    int i;
    for(i=1; i<=n; i++) a[i]=0;
    do{
        tim(); i=n;
        while(i>0 && a[i]==1) i--;
        if(i>0){
            a[i]=1;
            for(int j=i+1; j<=n; j++) a[j]=0;
        }
    }while(i!=0);
    xuat();
}
int main(){
    ios_base::sync_with_stdio(0); cin.tie(0); cout.tie(0);
    cin>>n>>k;

```

Code C++

```

    for(int i=1; i<=n; i++) cin>>b[i];
    sinh();
    return 0;
}

```

10. DQBN9.*

Hãy liệt kê ra các cách chia dãy trên thành hai nhóm sao cho tổng mỗi nhóm có độ chênh lệch nhỏ nhất. Đưa ra hai dãy là chỉ số của các phần tử được chọn của mỗi nhóm và độ chênh lệch nhỏ nhất.

Code C++

```

#include<bits/stdc++.h>
#define ll long long
#define h round (1e18)
using namespace std;
int a[34],b[34],d[34]; int n; ll sum=0,res=h,tmp;
void xuat(){
    cout<<"nhom1:";
    for(int i=1; i<=n; i++)
        if(d[i]==1) cout<<i<<" ";
    cout<<"\nnhom2:";
    for(int i=1; i<=n; i++)
        if(d[i]==0) cout<<i<<" ";
}
void tim(){
    tmp=0;
    for(int i=1; i<=n; i++) tmp+=1LL*a[i]*b[i];
    if(res>=abs(sum-2*tmp)){
        res=abs(sum-2*tmp);
        for(int i=1; i<=n; i++) d[i]=a[i];
    }
}
void sinh(){
    int i;
    for(i=1; i<=n; i++) a[i]=0;
    do{

```

Code C++

```

        tim(); i=n;
        while(i>0 && a[i]==1) i--;
        if(i>0){
            a[i]=1;
            for(int j=i+1; j<=n; j++) a[j]=0;
        }
    }while(i!=0);
    xuat();
}
int main(){
    ios_base::sync_with_stdio(0); cin.tie(0); cout.tie(0);
    cin>>n;
    for(int i=1; i<=n; i++){
        cin>>b[i]; sum+=b[i];
    }
    sinh();
    return 0;
}

```

High:

Nhập N và dãy b_1, b_2, \dots, b_n , dãy c_1, c_2, \dots, c_n .

11.DQBN10.*

Hãy đưa ra các cách chọn các phần tử $b[i]$ để **tổng chia hết cho K** và tổng $c[i]$ được chọn lớn nhất. Đưa ra giá trị tổng $c[i]$ lớn nhất tìm được.

Thuật toán:

Thử các cách chọn, với mỗi cấu hình ta thực hiện:

Duyệt từ 1 tới N:

Nếu $a[i] = 1$ thì:

{ $Sb += b[i];$

$Sc += c[i];$ }

Nếu $Sb \% K == 0$ thì $res = \max(res, Sc);$

Code C++

```
#include<bits/stdc++.h>
```

```
#define ll long long
```

Code C++

```

#define h round(1e18+1)
using namespace std;
int a[34],b[34],c[34],d[34]; int n,k; ll sumb,sumc,res=-h; bool ok;
void xuat(){
    for(int i=1; i<=n; i++)
        if(d[i]==1) cout<<b[i]<<" ";
    cout<<"\n"<<res;
}
void tim(){
    sumb=sumc=0;
    for(int i=1; i<=n; i++){
        sumb+=1LL*a[i]*b[i];
        sumc+=1LL*a[i]*c[i];
    }
    if(sumb%k==0)
        if(sumc>=res){
            res=sumc; ok=true;
            for(int i=1; i<=n; i++) d[i]=a[i];
        }
}
void sinh(){
    int i;
    for(i=1; i<=n; i++) a[i]=0;
    do{
        tim(); i=n;
        while(i>0 && a[i]==1) i--;
        if(i>0){
            a[i]=1;
            for(int j=i+1; j<=n; j++) a[j]=0;
        }
    }while(i!=0);
    if(ok) xuat();
}
int main(){
    ios_base::sync_with_stdio(0);
    cin.tie(0); cout.tie(0);
}

```

Code C++

```

        cin>>n>>k;
        for(int i=1; i<=n; i++) cin>>b[i];
        for(int i=1; i<=n; i++) cin>>c[i];
        sinh();
        return 0;
    }

#include<bits/stdc++.h>

using namespace std;

int n,k,sumb,sumc,m=0;
int a[34], b[34], c[34],d[34];

void xuat(){
    for(int i=1; i <= n; i++){
        if(d[i]==1)cout<<b[i]<<" ";
    }
}

void tim(){
    sumb=0;
    sumc=0;
    for(int i=1; i <= n; i++){
        sumb+=a[i]*b[i];
        sumc+=a[i]*c[i];
    }
    if(sumb%k==0){
        if(sumc>=m){
            m=sumc;
            for(int i=1; i <= n; i++) d[i]=a[i];
        }
    }
}

void sinh(){
    int i;
    for(int i=1; i <= n; i++) a[i]=0;

```


Code C++

```

do{
    tim(); i=n;
    while(i>0&& a[i]==1)i--;
    if(i>0){
        a[i]=1;
        for(int j=i+1; j <= n; j++)a[j]=0;
    }
}while(i!=0);
xuat();
}
int main(){
    ios_base::sync_with_stdio(NULL);
    cin.tie(0); cout.tie(0);
    cin>>n>>k;
    for(int i=1; i<=n; i++)cin>>b[i];
    for(int i=1; i<=n; i++)cin >>c[i];
    sinh();
    return 0;
}

```

12. DQBN11.*

Hãy đưa ***các cách chọn*** các phần tử $b[i]$ để tổng các phần tử được chọn $\leq K$ và tổng $c[i]$ được chọn là lớn nhất. Với K cho trước.

Thuật toán:

Thử các cách chọn, với mỗi cấu hình ta thực hiện:

Duyệt từ 1 tới N :

Nếu $a[i] = 1$ thì:

```

{ Sb+=b[i];
  Sc+=c[i]; }

```

Nếu $Sb \leq K$ thì

```

if(sc>=m)m=sc;

```

Khi đó thu được m là giá trị lớn nhất của sc .

Thực hiện sinh lại lần nữa, với mỗi cấu hình tương tự ta tính sb , sc :

Nếu $Sb \leq K$ thì

```

if(sc == m)

```

//đưa cấu hình ra.

Code C++

```
#include<bits/stdc++.h>
#define ll long long
#define h round(1e18+1)
using namespace std;
int a[34],b[34],c[34],d[34]; int n,k; ll sumb,sumc,res=-h; bool ok;
void xuat(){
    for(int i=1; i<=n; i++)
        if(d[i]==1) cout<<b[i]<<" ";
    cout<<"\n"<<res;
}
void tim(){
    sumb=sumc=0;
    for(int i=1; i<=n; i++){
        sumb+=1LL*a[i]*b[i];
        sumc+=1LL*a[i]*c[i];
    }
    if(sumb<=k)
        if(sumc>=res){
            res=sumc; ok=true;
            for(int i=1; i<=n; i++) d[i]=a[i];
        }
}
void sinh(){
    int i;
    for(i=1; i<=n; i++) a[i]=0;
    do{
        tim(); i=n;
        while(i>0 && a[i]==1) i--;
        if(i>0){
            a[i]=1;
            for(int j=i+1; j<=n; j++) a[j]=0;
        }
    }while(i!=0);
    if(ok) xuat();
}
```

Code C++

```
int main(){
    ios_base::sync_with_stdio(0);
    cin.tie(0); cout.tie(0);
    cin>>n>>k;
    for(int i=1; i<=n; i++) cin>>b[i];
    for(int i=1; i<=n; i++) cin>>c[i];
    sinh();
    return 0;
}
```

```
#include<bits/stdc++.h>
```

```
using namespace std;
```

```
int n,k,sumb,sumc,m=0;
```

```
int a[34],b[34],c[34],d[34];
```

```
void xuat(){
```

```
    for(int i=1; i<=n; i++){
        if(d[i]==1)cout << b[i];
    }
```

```
}
```

```
void tim(){
```

```
    sumb=0;
    sumc=0;
    for(int i=1; i<=n; i++){
        sumb+=a[i]*b[i];
        sumc+=a[i]*c[i];
        if(sumb>k)return;
    }
    if(sumc>=m){
        m=sumc;
        for(int i=1; i <= n; i++) d[i]=a[i];
    }
```

```
}
```

```
void sinh(){
```

Code C++

```

int i;
for(int i=1; i<=n; i++)a[i]=0;
do{
    tim();i=n;
    while(i>0&&a[i]==1)i--;
    if(i>0){
        a[i]=1;
        for(int j=i+1;j<=n;j++)a[j]=0;
    }
}while(i!=0);
xuat();
}
int main(){
    ios_base::sync_with_stdio(NULL);
    cin.tie(0); cout.tie(0);
    cin>>n>>k;
    for(int i=1; i<=n; i++) cin >> b[i];
    for(int i=1; i<=n; i++) cin >> c[i];
    sinh();
    return 0;
}

```

```

#include <bits/stdc++.h>
#define ll long long
using namespace std;

const int nmax = 1e5;
bool a[25], t[nmax][25];
int n, k, b[25], c[25], val[nmax], d = 0;
ll Max = INT_MIN;

void input () {
    cin >> n >> k;
    for (int i = 1; i <= n; ++i) cin >> b[i];
    for (int i = 1; i <= n; ++i) cin >> c[i];
}

```

Code C++

```

void handle () {
    ll sum = 0, x = 0;
    for (int i = 1; i <= n; ++i) {
        x += a[i] * b[i];
        sum += a[i] * c[i];
    }
    if (x <= k) {
        Max = max (Max, sum);
        val[d] = sum;
        for (int i = 1; i <= n; ++i) t[d][i] = a[i];
        ++d;
    }
}

void output () {
    for (int i = 0; i < d; ++i)
        if (val[i] == Max) {
            for (int j = 1; j <= n; ++j) if (t[i][j]) cout << b[j] << ' ';
            cout << "\n";
        }
}

void gen () {
    memset (a, 0, sizeof(a));
    int i;
    do {
        // handle
        handle ();
        i = n;
        while ((i > 0) && a[i]) --i;
        if (i > 0) {
            a[i] = 1;
            memset(a + i + 1, 0, sizeof(int) * (n - i + 1));
        }
    } while (i > 0);
}

```

Code C++

```

}

int main () {
    ios_base::sync_with_stdio(NULL);
    cin.tie(NULL); cout.tie(NULL);
    input ();
    gen ();
    output ();
    return 0;
}

```

13. DQBN12.*

Hãy đưa ra ***các cách chọn*** một số phân tử để tổng $b[i]$ được chọn đúng bằng K và $|C_{\max} - C_{\min}|$ của các phân tử được chọn là lớn nhất. $|C_{\max} - C_{\min}|$ lớn nhất. Với K cho trước.

Thuật toán:

Với mỗi cấu hình:

+ tính tổng $b[i]$ được chọn.

Nếu $sb = k$ thì

- + cập nhật $rcmax = \max(rcmax, c[i]);$
- + cập nhật $rcmin = \min(rcmin, c[i]);$
- + cập nhật $res = \max(res, \text{abs}(rmax - rmin));$

+ sinh2(): gọi xuli2() thực hiện

Nếu $sb = k$ thì

- + cập nhật $rcmax = \max(rcmax, c[i]);$
- + cập nhật $rcmin = \min(rcmin, c[i]);$
- Nếu $res = \text{abs}(rcmax - rcmin)$ thì đưa cấu hình ra.

Code C++

```

#include<bits/stdc++.h>
#define ll long long
#define h round(1e18+1)
using namespace std;
int a[34],b[34],c[34],d[34]; int n,k,m=0; ll sumb,sumc; ll res[3]; bool ok;

```

Code C++

```

void xuat(){
    for(int i=1; i<=n; i++)
        if(d[i]==1) cout<<b[i]<<" ";
    if(m==2) cout<<"\n"<<res[2]-res[1];
}

void tim(){
    sumb=sumc=0;
    for(int i=1; i<=n; i++){
        sumb+=1LL*a[i]*b[i];
        sumc+=1LL*a[i]*c[i];
    }
    if(sumb==k){
        ok=true;
        if(m==0) for(int i=1; i<=n; i++) d[i]=a[i];
        if(m<2) res[++m]=sumc;
        else{
            if(sumc<res[1]) res[1]=sumc;
            else if(sumc>res[2]) res[2]=sumc;
        }
    }
}

void sinh(){
    int i;
    for(i=1; i<=n; i++) a[i]=0;
    do{
        tim(); i=n;
        while(i>0 && a[i]==1) i--;
        if(i>0){
            a[i]=1;
            for(int j=i+1; j<=n; j++) a[j]=0;
        }
    }while(i!=0);
    if(ok) xuat();
}

int main(){
    ios_base::sync_with_stdio(0);

```

Code C++

```

        cin.tie(0); cout.tie(0);
        cin>>n>>k;
        for(int i=1; i<=n; i++) cin>>b[i];
        for(int i=1; i<=n; i++) cin>>c[i];
        res[1]=h; res[2]=-h;
        sinh();
        return 0;
    }

#include<bits/stdc++.h>

using namespace std;

int n,k,sumb,sumc,mi=0,ma=0,m;
int a[34], b[34], c[34], d[34];
void xuat(){
    for(int i=1; i<=n; i++){
        if(d[i]==1)cout << b[i] <<" ";
    }
}
void tim(){
    sumb=0;
    sumc=0;
    for(int i=1; i<=n; i++){
        sumb+=a[i]*b[i];
        sumc+=a[i]*c[i];
        if(sumb>k)return;
    }
    if(sumb==k){
        if(sumc>=ma){
            ma=sumc;
        }
        if(sumc<mi){
            mi=sumc;
        }
        if((ma-mi)>=m){

```


Code C++

```

                m=ma-mi;
                for(int i=1; i <= n; i++) d[i]=a[i];
            }
        }
    }
void sinh(){
    int i;
    for(int i=1; i<=n; i++)a[i]=0;
    do{
        tim();i=n;
        while(i>0&& a[i]==1)i--;
        if(i>0){
            a[i]=1;
            for(int j=i+1;j<=n;j++)a[j]=0;
        }
    }while(i!=0);
    xuat();
}
int main(){
    ios_base::sync_with_stdio(NULL);
    cin.tie(0);cout.tie(0);
    cin>>n>>k;
    for(int i=1; i<=n; i++)cin>>b[i];
    for(int i=1; i<=n; i++)cin>>c[i];
    sinh();
    return 0;
}

#include <bits/stdc++.h>
#define ll long long
using namespace std;

const int nmax = 1e5;
bool a[25], t[nmax][25];
int n, k, b[25], c[25], val[nmax], d = 0, res = INT_MIN;

```

Code C++

```

void input () {
    cin >> n >> k;
    for (int i = 1; i <= n; ++i) cin >> b[i];
    for (int i = 1; i <= n; ++i) cin >> c[i];
}

void handle () {
    ll sum = 0;
    int Max = INT_MIN, Min = INT_MAX;
    for (int i = 1; i <= n; ++i) {
        sum += a[i] * b[i];
        if (a[i]) Max = max(Max, c[i]);
        if (a[i]) Min = min(Min, c[i]);
        t[d][i] = a[i];
    }
    if (sum == k) {
        res = max(res, Max - Min);
        val[d] = Max - Min;
        ++d;
    }
}

void output () {
    for (int i = 0; i < d; ++i)
        if (val[i] == res) {
            for (int j = 1; j <= n; ++j) if (t[i][j]) cout << b[j] << ' ';
            cout << '\n';
        }
}

void gen () {
    memset (a, 0, sizeof(a));
    int i;
    do {
        // handle
        handle ();
    } while (1);
}

```

Code C++

```

    i = n;
    while ((i > 0) && a[i]) --i;
    if (i > 0) {
        a[i] = 1;
        memset(a + i + 1, 0, sizeof(int) * (n - i + 1));
    }
} while (i > 0);
}

int main () {
    ios_base::sync_with_stdio(NULL);
    cin.tie(NULL); cout.tie(NULL);
    input ();
    gen ();
    output ();
    return 0;
}

```

14. DQBN13.*

Hãy đưa ra **các cách chọn đúng K** phần tử để tổng $(b[i] + c[i])$ là lớn nhất. Đưa tổng lớn nhất của $b[i]$ và $c[i]$ ra.

Thuật toán:

Với mỗi cấu hình:

Đếm số lượng bit 1: for (1,1,N)

```

{
    dem += a[i];
    + tính tổng b[i] được chọn
    + tính tổng c[i] được chọn.
}

```

nếu $dem = K$ thì:

+ cập nhật $rmax = \max(rmax, sb+sc);$

Code C++

```
#include<bits/stdc++.h>
```

Code C++

```

#define ll long long
#define h round(1e18+1)
using namespace std;
int a[34],b[34],c[34],d[34]; int n,k,m=0; ll sumb,sumc,res=-h,ans=-h; bool ok;
void xuat(){
    for(int i=1; i<=n; i++)
        if(d[i]==1) cout<<b[i]<<" ";
    cout<<"\n"<<res<<" "<<ans;
}
void tim(){
    sumb=sumc=0; m=0;
    for(int i=1; i<=n; i++)
        if(a[i]==1){
            sumb+=1LL*b[i];
            sumc+=1LL*c[i];
            m++;
        }
    if(m==k){
        ok=true;
        if(sumb+sumc>=res+ans){
            res=sumb; ans=sumc;
            for(int i=1; i<=n; i++) d[i]=a[i];
        }
    }
}
void sinh(){
    int i;
    for(i=1; i<=n; i++) a[i]=0;
    do{
        tim(); i=n;
        while(i>0 && a[i]==1) i--;
        if(i>0){
            a[i]=1;
            for(int j=i+1; j<=n; j++) a[j]=0;
        }
    }while(i!=0);
}

```

Code C++

```

        if(ok) xuat();
    }
int main(){
    ios_base::sync_with_stdio(0);
    cin.tie(0); cout.tie(0);
    cin>>n>>k;
    for(int i=1; i<=n; i++) cin>>b[i];
    for(int i=1; i<=n; i++) cin>>c[i];
    sinh();
    return 0;
}

```

```

#include <bits/stdc++.h>
#define ll long long
using namespace std;

const int nmax = 1e5;
bool a[25], t[nmax][25];
int n, k, b[25], c[25], valb[nmax], valc[nmax], d = 0;
ll maxb = LLONG_MIN, maxc = LLONG_MIN;

void input () {
    cin >> n >> k;
    for (int i = 1; i <= n; ++i) cin >> b[i];
    for (int i = 1; i <= n; ++i) cin >> c[i];
}

void handle () {
    ll sumb = 0, sumc = 0;
    int dem = 0;
    ll Max = LLONG_MIN, Min = LLONG_MAX;
    for (int i = 1; i <= n; ++i) {
        dem += a[i];
        sumb += a[i] * b[i];
        sumc += a[i] * c[i];
        t[d][i] = a[i];
    }
}

```

Code C++

```

    }
    if (dem == k) {
        Max = max (Max,sumb + sumc);
        val [d] = sumb + sumc;
        ++d;
    }
}

void output () {
    for (int i = 0; i < d; ++i)
        if (val[i] == Max) {
            for (int j = 1; j <= n; ++j) if (t[i][j]) cout << j << ' ';
            cout << '\n';
        }
}

void gen () {
    memset (a, 0, sizeof(a));
    int i;
    do {
        // handle
        handle ();
        i = n;
        while ((i > 0) && a[i] --i;
        if (i > 0) {
            a[i] = 1;
            memset(a + i + 1, 0, sizeof(int) * (n - i + 1));
        }
    } while (i > 0);
}

int main () {
    ios_base::sync_with_stdio(NULL);
    cin.tie(NULL); cout.tie(NULL);
    input ();
    gen ();
}

```

Code C++

```

output ();
return 0;
}

```

Hard:

Nhập N và dãy b_1, b_2, \dots, b_n , dãy c_1, c_2, \dots, c_N , dãy d_1, d_2, \dots, d_N .

15. DQBN14.*

Hãy đưa ra **các cách chọn một** số phần tử để tổng $b[i]$ được chọn $\leq M$, tổng $d[i]$ được chọn $\leq V$ và tổng $c[i]$ được chọn là lớn nhất. Đưa ra tổng sc lớn nhất tìm được.

Thuật toán:

- Duyệt các cấu hình, với mỗi cấu hình tính sb, sc, sd . Thêm nhánh cận cho trường hợp $sb > M$ và $sd > V$.
- Nếu thoả mãn thì cập nhật $res = \max(res, sc)$;
- Tổ chức sinh1(1);
- Tổ chức sinh2(1); đưa cấu hình đó ra.

```

#include<bits/stdc++.h>
#define ll long long
#define h round(1e18+1)
using namespace std;
int a[34],b[34],c[34],d[34],f[34][34],w[34]; int n,cnt=0; ll m,v,sb,sc,sd,res=-h; bool
ok;
void xuat(){
    for(int i=1; i<=cnt; i++)
        if(w[i]==res){
            for(int j=1; j<=n; j++)
                if(f[i][j]) cout<<b[j]<<" "<<d[j]<<" "<<c[j]<<" ";
            cout<<"\n";
        }
    cout<<res;
}
void tim(){
    sb=sd=sc=0;
    for(int i=1; i<=n; i++){
        sb+=1LL*a[i]*b[i];
        sc+=1LL*a[i]*c[i];
    }
}

```

```

        sd+=1ll*a[i]*d[i];
        if((sb>m) || (sd>v)) return;
    }
    if(sc>=res){
        res=sc; ok=true; w[++cnt]=res;
        for(int i=1; i<=n; i++) f[cnt][i]=a[i];
    }
}
void sinh(){
    int i;
    for(i=1; i<=n; i++) a[i]=0;
    do{
        tim(); i=n;
        while(i>0 && a[i]==1) i--;
        if(i>0){
            a[i]=1;
            for(int j=i+1; j<=n; j++) a[j]=0;
        }
    }while(i!=0);
    if(ok) xuat();
    else cout<<-1;
}
int main(){
    ios_base::sync_with_stdio(0);
    cin.tie(0); cout.tie(0);
    cin>>n>>m>>v;
    for(int i=1; i<=n; i++) cin>>b[i];
    for(int i=1; i<=n; i++) cin>>c[i];
    for(int i=1; i<=n; i++) cin>>d[i];
    sinh();
    return 0;
}

```

```
#include <bits/stdc++.h>
```

```
#define ll long long
```

```
using namespace std;
```



```

const int nmax = 1e5;
int n, dem = 0;
ll b[nmax], c[nmax], d[nmax], t[nmax][20], valc[nmax], res = LLONG_MIN, m, v;
bool a[24];

void input () {
    cin >> n >> m >> v;
    for (int i = 1; i <= n; ++i) cin >> b[i];
    for (int i = 1; i <= n; ++i) cin >> c[i];
    for (int i = 1; i <= n; ++i) cin >> d[i];
}

void handle () {
    ll sumb, sumc, sumd;
    sumb = sumc = sumd = 0;
    for (int i = 1; i <= n; ++i) {
        sumb += a[i] * b[i];
        sumc += a[i] * c[i];
        sumd += a[i] * d[i];
        t[dem][i] = a[i];
    }
    if ((sumb <= m) && (sumd <= v) && (sumc >= res)) {
        res = sumc;
        valc[dem] = sumc;
        ++dem;
    }
}

void gen () {
    memset (a, 0, sizeof(a));
    int i;
    do {
        handle ();
        i = n;
        while ((i > 0) && (a[i])) --i;
        if (i > 0) {
            a[i] = 1;

```

```

        memset (a + i + 1, 0, sizeof(bool) * (n - i + 1));
    }
} while (i > 0);
}

void output () {
    for (int i = 0; i < dem; ++i) {
        if (valc[i] == res) {
            for (int j = 1; j <= n; ++j) cout << t[i][j] << ' ';
            cout << '\n';
        }
    }
}

int main () {
    ios_base::sync_with_stdio(NULL);
    cin.tie(NULL); cout.tie(NULL);
    freopen ("DQBN14.INP", "r", stdin);
    freopen ("DQBN14.OUT", "w", stdout);
    input ();
    gen ();
    cout << res << '\n';
    output();
    return 0;
}

```

```
#include<bits/stdc++.h>
```

```
using namespace std;
```

```
int n,m,v,res=INT_MIN;
```

```
int sb,sc,sd;
```

```
int a[34],b[34],c[34],d[34],e[34];
```

```
void out(){
```

```
    for(int i=1;i<=n;i++){
```

```
        if(e[i]==1)cout<<b[i]<<" ";
```

```
    }
```

```

    cout << '\n' << sc;
}
void in(){
    cin >> n >> m >> v;
    for(int i=1; i<=n; i++) cin >> b[i];
    for(int i=1; i<=n; i++) cin >> d[i];
    for(int i=1; i<=n; i++) cin >> c[i];
}
void go(){
    sb=0;
    sc=0;
    sd=0;
    for(int i=1; i<=n; i++){
        sb+=1LL*a[i]*b[i];
        sc+=1LL*a[i]*c[i];
        sd+=1LL*a[i]*d[i];
        if(sb>m)return;
        if(sd>v)return;
    }
    if(sb<=m){
        if(sd<=v){
            res = max(res,sc);
        }
    }
    if(sb<=m){
        if(sd<=v){
            if(sc==res){
                for(int i=1; i<=n; i++)e[i]=a[i];
            }
        }
    }
}
void sinh(){
    int i;
    for(int i=1; i <= n; i++) a[i]=0;
    do{
        go(); i=n;
    }
}

```

```

        while(i>0&& a[i]==1)i--;
        if(i>0){
            a[i]=1;
            for(int j=i+1; j <= n; j++)a[j]=0;
        }
    }while(i!=0);
    out();
}
int main(){
    ios_base::sync_with_stdio(NULL);
    cin.tie(0);cout.tie(0);
    in();
    sinh();
    return 0;
}

```

16. DQBN15.*

Hãy đưa ra các cách chọn K phần tử để tổng $b[i] \leq M$, tổng $d[i] \leq V$ và tổng $c[i]$ là lớn nhất. Đưa ra tổng lớn nhất thu được.

Thuật toán:

- Duyệt các cấu hình, với mỗi cấu hình tính sb, sc, sd. Thêm nhánh cận cho trường hợp $sb > M$ và $sd > V$. Tính đếm = đếm + a[i];
- Nếu **thoả mãn** thì cập nhật $res = \max(res, sc)$;
- Tổ chức sinh1(1);
- Tổ chức sinh2(1); đưa cấu hình đó ra.

```

#include<bits/stdc++.h>
#define ll long long
#define h round(1e18+1)
using namespace std;
int a[34],b[34],c[34],d[34],f[34][34],w[34]; int n,k,dem,cnt=0; ll m,v,sb,sc,sd,res=-h;
bool ok;
void xuất(){
    for(int i=1; i<=cnt; i++)
        if(w[i]==res){
            for(int j=1; j<=n; j++)
                if(f[i][j]) cout<<b[j]<<" "<<d[j]<<" "<<c[j]<<" ";
            cout<<"\n";
        }
}

```

```

    }
    cout<<res;
}
void tim(){
    sb=sd=sc=0; dem=0;
    for(int i=1; i<=n; i++)
        if(a[i]){
            sb+=1LL*b[i];
            sc+=1LL*c[i];
            sd+=1ll*d[i];
            dem++;
            if((dem>k) || (sb>m) || (sd>v)) return;
        }
    if(dem==k && sc>=res){
        res=sc; ok=true; w[++cnt]=res;
        for(int i=1; i<=n; i++) f[cnt][i]=a[i];
    }
}
void sinh(){
    int i;
    for(i=1; i<=n; i++) a[i]=0;
    do{
        tim(); i=n;
        while(i>0 && a[i]==1) i--;
        if(i>0){
            a[i]=1;
            for(int j=i+1; j<=n; j++) a[j]=0;
        }
    }while(i!=0);
    if(ok) xuat();
    else cout<<-1;
}
int main(){
    ios_base::sync_with_stdio(0);
    cin.tie(0); cout.tie(0);
    cin>>n>>k>>m>>v;
    for(int i=1; i<=n; i++) cin>>b[i];

```

```

        for(int i=1; i<=n; i++) cin>>c[i];
        for(int i=1; i<=n; i++) cin>>d[i];
        sinh();
        return 0;
    }

```

```
#include<bits/stdc++.h>
```

```
using namespace std;
```

```
int n,k,m,v,dem=0,res=INT_MIN;
```

```
int sb,sc,sd;
```

```
int a[34],b[34],c[34],d[34],e[34];
```

```
void out(){
```

```
    for(int i=1; i<=n; i++){
```

```
        if(e[i]==1)cout<<b[i]<<" ";
```

```
    }
```

```
    cout << '\n' << sc;
```

```
}
```

```
void in(){
```

```
    cin >> n >> k >> m >> v;
```

```
    for(int i=1; i<=n; i++) cin >> b[i];
```

```
    for(int i=1; i<=n; i++) cin >> d[i];
```

```
    for(int i=1; i<=n; i++) cin >> c[i];
```

```
}
```

```
void go(){
```

```
    sb=0;
```

```
    sc=0;
```

```
    sd=0;
```

```
    for(int i=1; i<=n; i++){
```

```
        sb+=1LL*a[i]*b[i];
```

```
        sc+=1LL*a[i]*c[i];
```

```
        sd+=1LL*a[i]*d[i];
```

```
        dem++;
```

```
        if(dem>k)return;
```

```
        if(sb>m)return;
```

```
        if(sd>v)return;
```

```

    }
    if(dem==k){
        if(sb<=m){
            if(sd<=v){
                res = max(res,sc);
            }
        }
        if(sb<=m){
            if(sd<=v){
                if(sc==res){
                    for(int i=1; i<=n; i++)e[i]=a[i];
                }
            }
        }
    }
}

void sinh(){
    int i;
    for(int i=1; i <= n; i++) a[i]=0;
    do{
        go(); i=n;
        while(i>0&& a[i]==1)i--;
        if(i>0){
            a[i]=1;
            for(int j=i+1; j <= n; j++)a[j]=0;
        }
    }while(i!=0);
    out();
}

int main(){
    ios_base::sync_with_stdio(NULL);
    cin.tie(0);cout.tie(0);
    in();
    sinh();
    return 0;
}

```

```
#include <bits/stdc++.h>
```

```

#define ll long long

using namespace std;

const int nmax = 1e5;
int n, k, dem = 0;
ll b[nmax], c[nmax], d[nmax], t[nmax][20], valc[nmax], res = LLONG_MIN, m, v;
bool a[24];

void input () {
    cin >> n >> m >> v >> k;
    for (int i = 1; i <= n; ++i) cin >> b[i];
    for (int i = 1; i <= n; ++i) cin >> c[i];
    for (int i = 1; i <= n; ++i) cin >> d[i];
}

void handle () {
    ll sumb, sumc, sumd;
    sumb = sumc = sumd = 0;
    int x = 0;
    for (int i = 1; i <= n; ++i) {
        x += a[i];
        sumb += a[i] * b[i];
        sumc += a[i] * c[i];
        sumd += a[i] * d[i];
        t[dem][i] = a[i];
    }
    if ((x == k) && (sumb <= m) && (sumd <= v) && (sumc >= res)) {
        res = sumc;
        valc[dem] = sumc;
        ++dem;
    }
}

void gen () {
    memset (a, 0, sizeof(a));
    int i;

```



```

do {
    handle ();
    i = n;
    while ((i > 0) && (a[i])) --i;
    if (i > 0) {
        a[i] = 1;
        memset (a + i + 1, 0, sizeof(bool) * (n - i + 1));
    }
} while (i > 0);
}

void output () {
    for (int i = 0; i < dem; ++i) {
        if (valc[i] == res) {
            for (int j = 1; j <= n; ++j) cout << t[i][j] << ' ';
            cout << '\n';
        }
    }
}

int main () {
    ios_base::sync_with_stdio(NULL);
    cin.tie(NULL); cout.tie(NULL);
    freopen ("DQBN15.INP", "r", stdin);
    freopen ("DQBN15.OUT", "w", stdout);
    input ();
    gen ();
    cout << res << '\n';
    output();
    return 0;
}

```

17.DQBN16.*

Hãy đưa ra cách chọn nhiều nhất các phần tử để tổng $b[i] \leq M$, tổng $d[i] \leq V$ và tổng $c[i]$ là bằng K . Đưa ra tổng lớn nhất thu được.

Thuật toán:

- Duyệt các cấu hình, với mỗi cấu hình tính sb, sc, sd. Thêm nhánh cận cho trường hợp sb > M và sd > V. Tính đếm = đếm + a[i];
- Nếu **thoả mãn** thì cập nhật res = max (res, **đếm**);
- Tổ chức sinh1(1);
- Tổ chức sinh2(1); đưa cấu hình đó ra.

```
#include<bits/stdc++.h>
#define ll long long
#define h round(1e9+1)
using namespace std;
int a[34],b[34],c[34],d[34],f[34][34],w[34]; int n,dem,res=-h,cnt=0; ll m,v,sb,sc,sd,k; bool
ok;
void xuat(){
    for(int i=1; i<=cnt; i++)
        if(w[i]==res){
            for(int j=1; j<=n; j++)
                if(f[i][j]) cout<<b[j]<<" "<<d[j]<<" "<<c[j]<<" ";
            cout<<"\n";
        }
    cout<<res;
}
void tim(){
    sb=sd=sc=0; dem=0;
    for(int i=1; i<=n; i++)
        if(a[i]){
            sb+=1LL*b[i];
            sc+=1LL*c[i];
            sd+=1ll*d[i];
            dem++;
            if((sc>k) || (sb>m) || (sd>v)) return;
        }
    if(sc==k && dem>=res){
        res=dem; ok=true; w[++cnt]=res;
        for(int i=1; i<=n; i++) f[cnt][i]=a[i];
    }
}
void sinh(){
```

```

int i;
for(i=1; i<=n; i++) a[i]=0;
do{
    tim(); i=n;
    while(i>0 && a[i]==1) i--;
    if(i>0){
        a[i]=1;
        for(int j=i+1; j<=n; j++) a[j]=0;
    }
}while(i!=0);
if(ok) xuat();
else cout<<-1;
}
int main(){
    ios_base::sync_with_stdio(0);
    cin.tie(0); cout.tie(0);
    cin>>n>>k>>m>>v;
    for(int i=1; i<=n; i++) cin>>b[i];
    for(int i=1; i<=n; i++) cin>>c[i];
    for(int i=1; i<=n; i++) cin>>d[i];
    sinh();
    return 0;
}

```

```
#include<bits/stdc++.h>
```

```
using namespace std;
```

```
int n,k,m,v,dem=0,res=INT_MIN;
```

```
int sb,sc,sd;
```

```
int a[34],b[34],c[34],d[34],e[34];
```

```
void out(){
```

```
    for(int i=1;i<=n;i++){
```

```
        if(e[i]==1)cout<<b[i]<<" ";
```

```
    }
```

```
}
```

```
void in(){
```

```

    cin >> n >> k >> m >> v;
    for(int i=1; i<=n; i++) cin >> b[i];
    for(int i=1; i<=n; i++) cin >> d[i];
    for(int i=1; i<=n; i++) cin >> c[i];
}
void go(){
    sb=0;
    sc=0;
    sd=0;
    for(int i=1; i<=n; i++){
        sb+=1LL*a[i]*b[i];
        sc+=1LL*a[i]*c[i];
        sd+=1LL*a[i]*d[i];
        if(sb>m)return;
        if(sd>v)return;
        if(sc>k)return;
    }
    if(sc==k){
        if(sb<=m){
            if(sd<=v){
                dem++;
                res = max(res,dem);
            }
        }
        if(sb<=m){
            if(sd<=v){
                if(dem==res){
                    for(int i=1; i<=n; i++)e[i]=a[i];
                }
            }
        }
    }
}
void sinh(){
    int i;
    for(int i=1; i <= n; i++) a[i]=0;
    do{

```

```

        go(); i=n;
        while(i>0&& a[i]==1)i--;
        if(i>0){
            a[i]=1;
            for(int j=i+1; j <= n; j++)a[j]=0;
        }
    }while(i!=0);
    out();
}
int main(){
    ios_base::sync_with_stdio(NULL);
    cin.tie(0);cout.tie(0);
    in();
    sinh();
    return 0;
}

```

18. NGỌC LỤC BẢO

Ngọc lục bảo là một loại đá quý màu xanh lá cây. Đó là một loại muối của Bery có công thức hóa học là $\text{Be}_3\text{Al}_2(\text{SiO}_3)_6$. Những viên ngọc lục bảo xanh đậm từ 5 cara trở lên còn đắt hơn cả kim cương (một cara bằng 200mg).



Sau vài năm làm việc cật lực một nhóm **3 người** gồm hai anh em và một người bạn thân quyết định chia tay nhau về nhà. Họ đã kiếm được một số ngọc đủ để về lập nghiệp với những công việc an toàn, ổn định hơn. Trước khi chia tay họ quyết định chia đều số ngọc theo khối lượng cho mỗi người: người bạn sẽ lấy một phần ba, hai anh em cùng giữ phần còn lại. Tổng cộng cả nhóm tìm được n viên ngọc, viên thứ i có khối lượng m_i cara ($1 \leq n \leq 500$, $1 \leq m_i \leq 1\,000$). Nếu có phương án chia ngọc thì họ sẽ chia ngay trước khi ra về. Trong trường hợp ngược lại – mọi người sẽ phải tìm cách bán và chia nhau bằng tiền, điều mà chưa ai muốn làm lúc này.

Yêu cầu: Cho n và m_i ($i = 1 \div n$). Hãy xác định xem có cách chia ngọc hay không. Nếu có – chỉ ra một phương án cho biết người bạn sẽ được nhận những viên nào. Trong trường hợp không chia được – đưa ra một số 0.

Dữ liệu: Vào từ file văn bản EMERALD.INP:

- Dòng đầu tiên chứa số nguyên n ,

- Dòng thứ 2 chứa n số nguyên m_1, m_2, \dots, m_n .

Kết quả: Đưa ra file văn bản EMERALD.OUT:

- Trong trường hợp chia được:
 - Dòng thứ nhất chứa số nguyên k – số ngọc người bạn được chia,
 - Dòng thứ 2 chứa k số nguyên – số thứ tự các viên ngọc người bạn sẽ lấy.
- Trong trường hợp không chia được: đưa ra một số 0.

Ví dụ:

EMERALD.INP
8
1 3 4 1 2 5 1 1

EMERALD.OUT
2
6 8



Thuật toán:

- Đọc dữ liệu và tính S.
- Duyệt các cấu hình, với mỗi cấu hình tính sb, tính dem. Thêm nhánh cận cho trường hợp $3*sb > S$.
- Nếu **thoả mãn** thì lưu cấu hình.
- Tổ chức sinh(1);
- Đưa đếm ra và đưa chỉ số các phần tử được chọn ra.

```
#include <bits/stdc++.h>

using namespace std;

const int nmax = 507, mmax = 5 * 1e5 + 4;
int n, m, l = 0, a[nmax], sum = 0, res[nmax];
bool f[mmax];

void input () {
    cin >> n;
    for (int i = 1; i <= n; ++i) {
        cin >> a[i];
        sum += a[i];
    }
}
```

```

    m = sum / 3;
}

void dp () {
    memset (f, 0, sizeof(f));
    f[0] = 1;
    for (int i = 1; i <= n; ++i) {
        for (int j = a[i]; j <= m; ++j) {
            f[j] = f[j - a[i]];
        }
    }
}

int main () {
    ios_base::sync_with_stdio(NULL);
    cin.tie(NULL); cout.tie(NULL);
    freopen ("EMERALD.INP", "r", stdin);
    freopen ("EMERALD.OUT", "w", stdout);
    input ();
    if (sum % 3 != 0) {
        cout << 0;
        return 0;
    }
    dp ();
    if (f[m] == false) {
        cout << 0;
        return 0;
    }
    while ((n > 0) && (m > 0)) {
        if (f[m - a[n]]) {
            ++l;
            res[l] = n;
            m -= a[n];
            --n;
        } else --n;
    }
    cout << l << '\n';
}

```

```

    for (int i = 1; i >= 1; --i) cout << res[i] << ' ';
    return 0;
}

```

```

#include<bits/stdc++.h>
#define N 501
using namespace std;
int a[N],b[N]; int n,s=0,tmp,k; bool ok=false;
void nhap(){
    cin>>n;
    for(int i=1; i<=n; i++){
        cin>>b[i];
        s+=b[i];
    }
}
void tim(){
    tmp=k=0;
    for(int i=1; i<=n; i++){
        if(a[i]){
            tmp+=b[i];
            k++;
        }
        if(tmp==s){
            ok=true;
            cout<<k<<"\n";
            for(int i=1; i<=n; i++){
                if(a[i]) cout<<i<<" ";
            }
        }
    }
}
void sinh(){
    int i; s/=3;
    for(i=1; i<=n; i++) a[i]=0;
    do{
        tim(); i=n;
        if(ok) return ;
        while(i>0 && a[i]==1) i--;
        if(i>0){
            a[i]=1;

```



```

        for(int j=i+1; j<=n; j++) a[j]=0;
    }
    }while(i!=0);
    if(!ok) cout<<0;
}
int main(){
    ios_base::sync_with_stdio(0);
    cin.tie(0); cout.tie(0);
    freopen("emerald.inp","r",stdin);
    freopen("emerald.out","w",stdout);
    nhap();
    if(s%3==0) sinh();
    else cout<<0;
    return 0;
}

```

19. MÔ ĐUN M

Cho n số nguyên a_1, a_2, \dots, a_n ($|a_i| < 10^9, 0 \leq n \leq 100\,000$). Hãy xác định dãy con nhiều phần tử nhất từ dãy đã cho, sao cho không có hai phần tử nào của dãy con có tổng chia hết cho m ($2 \leq m \leq 100\,000$).

Dữ liệu: Vào từ file văn bản MODM.INP:

- Dòng thứ nhất chứa 2 số nguyên n và m ,
- Dòng thứ 2 chứa n số nguyên a_1, a_2, \dots, a_n .

Kết quả: Đưa ra file văn bản MODM.OUT:

- Dòng thứ nhất chứa số nguyên k – số phần tử của dãy con tìm được,
- Dòng thứ 2 chứa k số nguyên – chỉ số trong dãy ban đầu của các phần tử thuộc dãy con.

Nếu có nhiều kết quả thì đưa ra một trong số chúng.

Ví dụ:

MODM.INP
3 2
1 100 10

MODM.OUT
2
1 2



p11 IND _7/2008A

Thuật toán:

Tổ chức thủ tục sinh(1);

Tổ chức thủ tục xuli(); thực hiện việc kiểm tra cấu hình có thoả mãn điều kiện.

Tổ chức hàm kiểmtra() mảng chứa các phần tử được chọn có thoả mãn.

Dùng mảng c[k] để lưu nghiệm, kiểm tra c[k] có thoả mãn.

```

Bool kiểmtra()
{
    For(I,1,k-1)
        For(j,i+1,k) if ( (c[i] + c[j]) % m == 0) return false;
    Return true;
}

Void xuli()
{
    k = 0;
    For(I,1,n)
        If (b[i] == 1)
            { k++; c[k] = a[i];
            }
    If (kiểmtra())
        for(I,1,n) d[i] = b[i];
}

```

20. ĐẦU TƯ CHỨNG KHOÁN

Harry làm việc ở một công ty tư vấn đầu tư chứng khoán. Nhiệm vụ của Harry là phân tích sự giao động chỉ số chứng khoán hàng ngày của sàn giao dịch, từ đó có thể các thông tin hữu ích để tư vấn cho các nhà đầu tư.

Nhiệm vụ của Harry trong đề án đang thực hiện là như sau: Cho dãy chỉ số chứng khoán của n ngày liên tiếp. Cần phải tìm ra dãy con dài nhất sao cho *chênh lệch hai chỉ số liên tiếp không ít hơn k* . Ví dụ, với dãy chỉ số chứng khoán 1014, 1024, 1034, 1045, 1030, 998 và $k = 15$ thì dãy con 1014, 1034, 998 là chấp nhận được, nhưng dãy 1014, 1045, 1030, 998 là dãy con dài nhất cần tìm.

Yêu cầu: Cho n, k và dãy các chỉ số chứng khoán ($1 \leq n \leq 100\,000, 1 \leq k \leq 10^9$). Các chỉ số chứng khoán là những số nguyên dương, có giá trị không vượt quá 10^9 . Hãy chỉ ra dãy con dài nhất thoả mãn các điều kiện đã nêu.

Dữ liệu: Vào từ file văn bản FINANCIAL.INP

- Dòng đầu tiên chứa hai số nguyên n và k ,
- Dòng thứ 2 chứa n số nguyên – các chỉ số chứng khoán.

Kết quả: Đưa ra file văn bản FINANCIAL.OUT:

- Dòng đầu tiên đưa ra độ dài của dãy con tìm được,

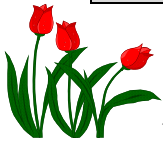
- Dòng thứ hai – các chỉ số thuộc dãy con theo trình tự xuất hiện.

Nếu có nhiều dãy con cùng độ dài thì đưa ra dãy tùy ý.

Ví dụ:

FINANCIAL.INP
6 15
1014 1024 1034 1045 1030 998

FINANCIAL.OUT
4
1014 1045 1030 998



p28 Int _10A/2008D

Thuật toán:

Tổ chức thủ tục sinh(1);

Tổ chức thủ tục xuli(); thực hiện việc kiểm tra cấu hình có thoả mãn điều kiện.

Tổ chức hàm kiểmtra() mảng chứa các phần tử được chọn có thoả mãn.

Dùng mảng c[p] để lưu nghiệm, kiểm tra c[p] có thoả mãn.

```

Bool kiểmtra()
{
    For(I,1,p-1)
        if ( abs(c[i] - c[i+1]) < K) return false;
    Return true;
}

```

```

Void xuli()
{
    p = 0;
    For(I,1,n)
        If (b[i] == 1)
            { p++; c[p] = a[i]; }
    If (kiểmtra())
        for(I,1,n) d[i] = b[i];
}

```

21. CHUỖI NHỊ PHÂN

Xét xâu nhị phân, tức là xâu chỉ chứa các ký tự trong tập $\{0, 1\}$. Gọi k là số lượng xâu nhị phân độ dài n ($1 \leq n \leq 10^4$) chứa xâu S (độ dài không quá 100) như một xâu con (các ký tự liên tiếp) đúng một lần.

Yêu cầu: Hãy tính phần dư của kết quả chia k cho 10^9+7 .

Dữ liệu: Vào từ file văn bản BINARY.INP:

- Dòng thứ nhất chứa số nguyên n ,
- Dòng thứ hai chứa xâu S .

Kết quả: Đưa ra file văn bản BINARY.OUT một số nguyên – kết quả tìm được.

Ví dụ: // p32 Ind9 _0103/2008A

BINARY.INP
4
01

BINARY.OUT
10

Thuật toán:

Tổ chức thủ tục sinh(1);

Tổ chức thủ tục xuli(); thực hiện việc kiểm tra cấu hình có thoả mãn điều kiện.

Tổ chức hàm kiểmtra() mảng chứa các phân tử được chọn có thoả mãn.

Bool kiểmtra() // kiểm tra xâu S xuất hiện đúng 1 lần trong xâu S0.

```
{
    Int d = 0;
    While pos(S,S0) > 0 do
        Begin
            Q = pos(S,S0);
            If q > 0 then
                Begin
                    delete(s0,1,q + length(s) - 1);
                    d++;
                end;
            if pos(s,s0) > 0 then exit(false);
        end;
        if d = 0 then exit(false);
        if d = 1 then exit(true);
    }
```

LIỆT KÊ CÁC TẬP CON K PHẦN TỬ CỦA TẬP N PHẦN TỬ

22. DQCKN1.* Tập con K phần tử

Cho N và K. Hãy liệt kê các tập con có K phần tử của N.

DQCKN1.INP	DQCKN1.OUT
5 3	0 1 2 0 1 3 0 1 4 0 1 5 0 2 3 0 2 4 0 2 5 0 3 4 0 3 5 0 4 5 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

Thuật toán:

- Phần tử thứ I có giá trị thuộc đoạn $a[i-1] \leq a[i] \leq N - K + i$;
- Khởi tạo cấu hình đầu tiên.
- Đi từ phải sang trái, tìm phần tử đầu tiên chưa đạt giá trị lớn nhất tại vị trí i.
- Tăng phần tử đó lên 1 đơn vị.
- Thay tất cả các phần tử sau nó bằng giới hạn dưới ($a[i-1] + 1$).

Chương trình:

Code C++

```
#include<bits/stdc++.h>
using namespace std;
int a[101]; int n,k;
void xuat(){
    for(int i=1; i<=k; i++) cout<<a[i]<<" ";
    cout<<"\n";
}
void xuly(){
```

```

int i;
for(i=1; i<=k; i++) a[i]=i;
do{
    xuat(); i=k;
    while(i>0 && a[i]==n-k+i) i--;
    if(i>0){
        a[i]++;
        for(int j=i+1; j<=k; j++) a[j]=a[j-1]+1;
    }
}while(i!=0);
}
int main(){
    ios_base::sync_with_stdio(0);
    cin.tie(0); cout.tie(0);
    cin>>n>>k;
    xuly();
    return 0;
}

```

Code Pascal

```

var
    n, k, i, ii, x: longint;
    a: array[-7..100007] of longint;
procedure xuli();
var p: longint;
begin
    for p:=1 to k do write(a[p],#32);
    writeln();
end;
begin
    assign(input,'kltt_22.inp');reset(input);
    assign(output,'kltt_22.out');rewrite(output);;
    readln(n, k);
    i := n;
    for i := 1 to k do a[i] := i;
    while i > 0 do
        begin
            xuli;

```

```

    i := k;
while (i > 0) and (a[i] = n - k + i) do dec(i);
if i > 0 then
    begin
        inc(a[i]);
        for ii:=i + 1 to k do a[ii] := a[ii - 1] + 1;
    end;
end;
close(input);close(output);
end.

```

Code Python

```

fi = open("DQCKN1.INP", "r")
fo = open("DQCKN1.OUT", "w")
sn = fi.readline().split()
n = int(sn[0])
k = int(sn[1])
x = n * [0]
def _xuli():
    for i in range(k):
        print(x[i], end = " ", file = fo)
    print("", file = fo)
def _lietke():
    for i in range(k):
        x[i] = i
    i = 0
    while (i >= 0):
        _xuli()
        i = k - 1
        while (i >= 0) and (n - k + i == x[i]):
            i = i - 1
        if (i >= 0):
            x[i] = x[i] + 1
            for j in range (i + 1, k ):
                x[j] = x[j-1] + 1
    _lietke()

```

Mid: Nhập N và dãy b_1, b_2, \dots, b_N .

23.DQCKN2.* Tập con K có tổng bằng S

Hãy đưa ra các cách chọn K phần tử trong dãy b sao cho có tổng bằng S.

DQCKN2.INP	DQCKN2.OUT
5 3 3	1 2 0
1 1 2 0 1	2 0 1

Thuật toán:

Chọn K phần tử trong dãy b bằng cách: Sinh các cấu hình chứa chỉ số các phần tử được chọn là a[i].

Với mỗi cấu hình ta tính tổng **tong** = **tong** + **b[a[i]]**;

Nếu **tong** = S thì lưu cấu hình.

Code C++

```
#include<bits/stdc++.h>
#define N 101
#define ll long long
using namespace std;
int a[N],b[N],f[N][N]; int n,m=0,k; ll s,sum;
void xuat(){
    for(int i=1; i<=m; i++){
        for(int j=1; j<=k; j++) cout<<f[i][j]<<" ";
        cout<<"\n";
    }
}
void tim(){
    sum=0;
    for(int i=1; i<=k; i++){
        sum+=1LL*b[a[i]];
        if(sum>s) return;
    }
    if(sum==s){
        m++;
        for(int i=1; i<=k; i++) f[m][i]=b[a[i]];
    }
}
```



```

void xuly(){
    int i;
    for(i=1; i<=k; i++) a[i]=i;
    do{
        tim(); i=k;
        while(i>0 && a[i]==n-k+i) i--;
        if(i>0){
            a[i]++;
            for(int j=i+1; j<=k; j++) a[j]=a[j-1]+1;
        }
    }while(i!=0);
}
int main(){
    ios_base::sync_with_stdio(0);
    cin.tie(0); cout.tie(0);
    cin>>n>>k>>s;
    xuly();
    xuat();
    return 0;
}

```

```

#include<bits/stdc++.h>

using namespace std;

int n,k,d,b[101],a[101],sb;

void in(){
    cin >> n >> k >> S;
    for(int i=1;i<=n;i++)cin>>b[i];
}

void out(){
    for(int i=1;i<=n;i++)cout<<a[i]<<' ';
    cout << "\n";
}

void go(){
    sb=0; d=0;
    for(int i=1; i<=n; i++){

```

```

        sb+=b[a[i]];
        d++;
        if((sb>S)||(d>k))return;
    }
    if((sb==S)&&(d==k)){
        for(int i=1;i<=n;i++)a[i]=b[i];
    }
}
void sinh(){
    int i;
    for(int i=1;i<=k;i++)a[i]=i;
    do{
        go();i=k;
        while((i>0)&&(a[i]==n-k+1)) i--;
        if(i>0){
            a[i]++;
            for(int j=i+1;j<=k;j++){
                a[j]=a[j-1]+1;
            }
        }
    }while(i!=0);
    out();
}
int main(){
    ios_base::sync_with_stdio(NULL);
    cin.tie(0);cout.tie(0);
    in();
    sinh();
    return 0;
}

```

Code Pascal

```

var
    k, n, s, i, ii: longint;
    b: array[-7..100007] of longint;
procedure xuli();
var p, tam, q: longint;
begin

```

```

    tam := 0;
    for p := 1 to k do
        tam := tam + b[p];
    if tam = s then
        for q := 1 to k do write(b[q],#32);
    writeln;
end;
begin
    assign(input,'btlk_22.inp');reset(input);
    assign(output,'btlk_22.out');rewrite(output);
    readln(n, k, s);
    i := n;
    for i := 1 to k do b[i] := i;
    while i > 0 do
        begin
            xuli;
            i := k;
            while (i > 0) and (b[i] = n - k + i) do dec(i);
            if i > 0 then
                begin
                    inc(b[i]);
                    for ii := i + 1 to k do b[ii] := b[ii - 1] + 1;
                end;
            end;
        close(input);close(output);
    end.

```

Code Python

```

fi = open("dqckn2.inp","r")
fo = open("dqckn2.out","w")
sn = fi.readline().split()
n = int(sn[0])
k = int(sn[1])
s = int(sn[2])
x = n * [0]
b = n * [0]
sa = fi.readline().split()
for i in range(n):

```

```

b[i] = int(sa[i])

def _xuli():
    tmp = 0
    for i in range(k):
        tmp = tmp + b[x[i]]
    if (tmp == s):
        for i in range(k):
            print(b[x[i]], end = " ", file = fo)
        print("", file = fo)

def _lietke():
    for i in range(k):
        x[i] = i
    i = 0
    while (i >= 0):
        _xuli()
        i = k - 1
        while (i >= 0) and (n - k + i == x[i]):
            i = i - 1
        if (i >= 0):
            x[i] = x[i] + 1
            for j in range (i + 1, k ):
                x[j] = x[j-1] + 1
    _lietke()

```

24. DQCKN3.* Tập con K có tổng nguyên tố

Hãy đưa ra các cách chọn K phần tử trong **dãy b** sao cho có tổng là số nguyên tố.

DQCKN3.INP	DQCKN3.OUT
5 3	1 2 4
1 2 1 4 5	1 1 5
	2 1 4
	2 4 5

Thuật toán:

Chọn K phần tử trong dãy b bằng cách: Sinh các cấu hình chứa chỉ số các phần tử được chọn là a[i].

Tổ chức hàm kiểm tra tính nguyên tố của số nguyên dương u.

Với mỗi cấu hình ta tính tổng **tong** = **tong** + **b[a[i]]**;

Nếu nếu **nguyên tố(tổng)** = **true** thì lưu cấu hình.

```
#include<bits/stdc++.h>

using namespace std;

int n,k,d,a[100000],b[100000],g[1000][1000],sb,m;

void in(){
    cin >> n >> k;
    for(int i=1;i<=k;i++)cin>>b[i];
}
void out(){
    for(int i=1;i<=m;i++){
        for(int j=1;j<=k;j++) cout << g[i][j] <<' ';
        cout << '\n';
    }
}
bool checksnt(int x){
    if(x<2)return false;
    for(int i=2;i*i <= x;i++){
        if(x%i==0)return false;
    }
    return true;
}
void go(){
    sb=0;
    for(int i=1; i<=n; i++){
        sb+=b[a[i]];
    }
    M = 0;
    if(checksnt(S))
        for(int i=1;i<=k;i++) g[m][i]=a[i];
```

```

}
void sinh(){
    int i;
    for(int i=1;i<=k;i++)a[i]=i;
    do{
        go();i=k;
        while((i>0)&&(a[i]==n-k+1)) i--;
        if(i>0){
            a[i]++;
            for(int j=i+1;j<=k;j++){
                a[j]=a[j-1]+1;
            }
        }
    }while(i!=0);
    out();
}
int main(){
    ios_base::sync_with_stdio(NULL);
    cin.tie(0);cout.tie(0);
    in();
    sinh();
    return 0;
}

```

```

#include<bits/stdc++.h>
#define N 101
#define ll long long
using namespace std;
int a[N],b[N],f[N][N]; int n,m=0,k; ll s,sum;
bool kt(ll x){
    if(x<2) return false;
    int y=sqrt(x);
    for(int i=2; i<=y; i++)
        if(!(x%i)) return false;
    return true;
}
void xuat(){

```

```

        for(int i=1; i<=m; i++){
            for(int j=1; j<=k; j++) cout<<f[i][j]<<" ";
            cout<<"\n";
        }
    }
void tim(){
    sum=0;
    for(int i=1; i<=k; i++) sum+=1LL*b[a[i]];
    if(kt(sum)){
        m++;
        for(int i=1; i<=k; i++) f[m][i]=b[a[i]];
    }
}
void xuly(){
    int i;
    for(i=1; i<=k; i++) a[i]=i;
    do{
        tim(); i=k;
        while(i>0 && a[i]==n-k+i) i--;
        if(i>0){
            a[i]++;
            for(int j=i+1; j<=k; j++) a[j]=a[j-1]+1;
        }
    }while(i!=0);
}
int main(){
    ios_base::sync_with_stdio(0);
    cin.tie(0); cout.tie(0);
    cin>>n>>k>>s;
    xuly();
    xuat();
    return 0;
}

```

Code Python

```

from math import sqrt
fi = open("dqckn3.inp", "r")
fo = open("dqckn3.out", "w")

```

```

sn = fi.readline().split()
n = int(sn[0])
k = int(sn[1])
x = n * [0]
b = n * [0]
sa = fi.readline().split()
for i in range(n):
    b[i] = int(sa[i])
def prime(u):    #Kiểm tra u có nguyên tố
    if (u <= 1):
        return False
    if (u == 2) or (u == 3):
        return True
    for i in range(2, int(sqrt(u))+1):
        if (u % i == 0):
            return False
    return True
def _xuli():
    tmp = 0
    for i in range(k):
        tmp = tmp + b[x[i]]
    if (prime(tmp)):
        for i in range(k):
            print(b[x[i]], end = " ", file = fo)
        print(end = "\n", file = fo)

def _lietke():
    for i in range(k):
        x[i] = i
    i = 0
    while (i >= 0):
        _xuli()
        i = k - 1
        while (i >= 0) and (n - k + i == x[i]):
            i = i - 1
        if (i >= 0):
            x[i] = x[i] + 1

```



```

for j in range (i + 1, k ):
    x[j] = x[j-1] + 1
_lietke()

```

25. DQCKN4.* Tập con K có tổng là số Palindrom

Hãy đưa ra các cách chọn K phần tử trong dãy b sao cho có tổng là số Palindrom lớn nhất. Số Palidrom là số đối xứng. Ví dụ: 1, 2, 3,...9, 11, 22, 101, 111, 202,...,1221...
Và đưa ra giá trị lớn nhất tìm được.

DQCKN4.INP	DQCKN4.OUT	Giải thích
5 3 1 13 8 1 2	22	1 13 8 = 22 1 8 2 = 11 1 1 2 = 4 13 8 1 = 22 8 1 2 = 11 Rmax = 22

Thuật toán:

Chọn K phần tử trong dãy b bằng cách: Sinh các cấu hình chứa chỉ số các phần tử được chọn là a[i].

Tổ chức hàm kiểm tra số nguyên dương u có là số Palidrom.

Với mỗi cấu hình ta tính tổng **tong** = **tong** + **b[a[i]]**;

Nếu nếu **doixung(tong)** = **true** thì cập nhật rmax và lưu cấu hình.

Code C++

```

#include<bits/stdc++.h>
#define N 101
#define ll long long
#define h round(1e18+1)
using namespace std;
int a[N],b[N],c[N],d[N],f[N][N]; int n,m=0,k; ll sum,res=-h;
bool kt(ll x){
    int w=0,y; ll tmp=x;
    while(tmp>0){
        c[++w]=tmp%10;
        tmp/=10;
    }

```

```

        y=w/2;
        for(int i=1; i<=y; i++)
            if(c[i]!=c[w-i+1]) return false;
        return true;
    }
    int timkiem(ll x){
        int l=1,r=m,mid;
        while(l<=r){
            mid=(l+r)/2;
            if(d[mid]==x) return mid;
            else if(d[mid]<x) l=mid+1;
            else r=mid-1;
        }
        return -1;
    }
    void nhap(){
        cin>>n>>k;
        for(int i=1; i<=n; i++) cin>>b[i];
    }
    void xuat(){
        int i=timkiem(res);
        if(i<1) return;
        for( ; i<=m; i++){
            for(int j=1; j<=k; j++) cout<<f[i][j]<<" ";
            cout<<"\n";
        }
        cout<<res;
    }
    void tim(){
        sum=0;
        for(int i=1; i<=k; i++) sum+=1LL*b[a[i]];
        if(sum>=res && kt(sum)){
            m++; res=sum; d[m]=sum;
            for(int i=1; i<=k; i++) f[m][i]=b[a[i]];
        }
    }
    void xuly(){

```

```

int i;
for(i=1; i<=k; i++) a[i]=i;
do{
    tim(); i=k;
    while(i>0 && a[i]==n-k+i) i--;
    if(i>0){
        a[i]++;
        for(int j=i+1; j<=k; j++) a[j]=a[j-1]+1;
    }
}while(i!=0);
}
int main(){
    ios_base::sync_with_stdio(0);
    cin.tie(0); cout.tie(0);
    nhap();
    xuly();
    xuat();
    return 0;
}

```

Code Python

```

from math import sqrt
fi = open("dqckn4.inp", "r")
fo = open("dqckn4.out", "w")
sn = fi.readline().split()
n = int(sn[0])
k = int(sn[1])
x = n * [0]
b = n * [0]
sa = fi.readline().split()
res = 0
for i in range(n):
    b[i] = int(sa[i])
def Palin(u):
    u = str(u)
    l = len(u)
    if (l == 1):
        return True

```

```

for i in range(l//2):
    if (u[i] != u[l - i - 1]):
        return False
    return True
def _xuli():
    global res
    tmp = 0
    for i in range(k):
        tmp = tmp + b[x[i]]
    if (palin(tmp)):
        res = max(res, tmp)
    # for i in range(k):
    #     print(b[x[i]], end = " ", file = fo)
    # print(end = "\n", file = fo)

def _lietke():
    for i in range(k):
        x[i] = i
    i = 0
    while (i >= 0):
        _xuli()
        i = k - 1
        while (i >= 0) and (n - k + i == x[i]):
            i = i - 1
        if (i >= 0):
            x[i] = x[i] + 1
            for j in range (i + 1, k ):
                x[j] = x[j-1] + 1
    _lietke()
print(res, end="", file = fo)

```

26. DQCKN51.* Tập con K chứa min và max

Hãy đưa ra các cách chọn K phần tử trong **dãy b** sao cho dãy được chọn **vừa chứa giá trị nhỏ nhất, vừa chứa giá trị lớn nhất.**

DQCKN51.INP	DQCKN51.OUT
5 3	1 8 2
1 8 2 1 3	1 8 1

	1 8 3
	8 2 1
	8 1 3

Thuật toán:

Tìm rmin, rmax khi đọc dữ liệu.

Chọn K phần tử trong dãy b bằng cách: Sinh các cấu hình chứa chỉ số các phần tử được chọn là a[i].

Tổ chức hàm kiểm tra các phần tử được chọn có thoả mãn điều kiện: vừa chứa rmin vừa chứa rmax.

Tổ chức hàm TKNP(U) có nằm trong dãy C có K phần tử hay không.

Nếu thoả mãn thì lưu cấu hình.

Sau khi sinh xong, đưa các cấu hình ra.

Code C++

```
#include<bits/stdc++.h>

using namespace std;

int n,k,d,a[100000],b[100000],g[1000][1000];

void in(){
    cin >> n >> k;
    for(int i = 1; i <= k; i++) cin >> b[i];
}

void out(){
    for(int i = 1; i <= k-1; i++){
        for(int j = i+1; j <= k; j++) cout << g[i][j] << ' ';
    }
    cout << "\n";
}

int minn(int b[], int k){
    int minn = b[0];
    for(int i = 1; i <= k; i++)
        if(minn > b[i])
            minn = b[i];
    return minn;
}
```

```

int maxx(int b[], int k){
    int maxx = b[0];
    for(int i = 1; i <= k; i++)
        if(maxx < b[i])
            maxx = b[i];
    return maxx;
}

bool check(int a[]){ //O(N^2)
    for(int i = 1; i < k; i++){
        if(a[i] == minn(b,k)){
            for(int j = i+1; j <= k; j++)
                if(a[j] == maxx(b,k)) return true;
        }
    }
    return false;
}

void go(){
    if(check(a)){
        d++;
        for(int i = 1; i <= d; i++)
            g[d][i] = a[i];
    }
}

void sinh(){
    int i;
    for(int i = 1; i <= k; i++) a[i] = i;
    do{
        go(); i = k;
        while((i > 0) && (a[i] == n-k+1)) i--;
        if(i > 0){
            a[i]++;
            for(int j = i+1; j <= k; j++){
                a[j] = a[j-1]+1;
            }
        }
    } while(i != 0);
    out();
}

```

```

    }

    int main(){
        ios_base::sync_with_stdio(NULL);
        cin.tie(0);cout.tie(0);
        in();
        sinh();
        return 0;
    }
}

#include <bits/stdc++.h>

using namespace std;

int n, k, a[24], b[24], Max = INT_MIN, Min = INT_MAX;

void show () {
    for (int i = 1; i <= k; ++i) cout << b[a[i]] << ' ';
    cout << '\n';
}

bool check () {
    bool check_max = false, check_min = false;
    for (int i = 1; i <= k; ++i) {
        if (b[a[i]] == Min) {
            check_min = true;
            if (check_max && check_min) return true;
        }
        if (b[a[i]] == Max) {
            check_max = true;
            if (check_max && check_min) return true;
        }
    }
    return false;
}

void gen () {
    for (int i = 1; i <= k; ++i) a[i] = i;
    int i;

```

```

do {
    if (check()) show ();
    i = k;
    while ((i > 0) && (a[i] == n - k + i)) --i;
    if (i > 0) {
        ++a[i];
        for (int j = i + 1; j <= k; ++j) a[j] = a[j - 1] + 1;
    }
} while (i > 0);
}

```

```

int main () {
    ios_base::sync_with_stdio(NULL);
    cin.tie(NULL); cout.tie(NULL);
    cin >> n >> k;
    for (int i = 1; i <= n; ++i) {
        cin >> b[i];
        Min = min(Min , b[i]);
        Max = max(Max , b[i]);
    }
    gen ();
    return 0;
}

```

```

#include<bits/stdc++.h>
#define N 101
#define ll long long
#define h round(1e18+1)
using namespace std;
int a[N],b[N],d[N],f[N][N]; int n,m=0,k; ll sum,res=-h,ans=h;
int timkiem(ll x){
    int l=1,r=m,mid;
    while(l<=r){
        mid=(l+r)/2;
        if(d[mid]==x){
            int i=mid-1;
            while(d[i]==d[mid]) i--;
            return i+1;
        }
    }
}

```



```

        }else if(d[mid]<x) l=mid+1;
        else r=mid-1;
    }
    return -1;
}
void xuat(){
    int i=timkiem(res);
    if(i<1) return;
    for( ; i<=m; i++){
        for(int j=1; j<=k; j++) cout<<f[i][j]<<" ";
        cout<<"\n";
    }
    cout<<res;
}
void tim(){
    sum=0;
    for(int i=1; i<=k; i++) sum+=1LL*b[a[i]];
    ans=min(ans,sum);
    if(sum>=res){
        m++; res=sum; d[m]=sum;
        for(int i=1; i<=k; i++) f[m][i]=b[a[i]];
    }
}
void xuly(){
    int i;
    for(i=1; i<=k; i++) a[i]=i;
    do{
        tim(); i=k;
        while(i>0 && a[i]==n-k+i) i--;
        if(i>0){
            a[i]++;
            for(int j=i+1; j<=k; j++) a[j]=a[j-1]+1;
        }
    }while(i!=0);
}
int main(){
    ios_base::sync_with_stdio(0);

```

```

        cin.tie(0); cout.tie(0);
        cin>>n>>k;
        xuly();
        if(res==ans) xuat();
        return 0;
    }

```

Code Python

```

from math import sqrt
fi = open("dqckn5.inp","r")
fo = open("dqckn5.out","w")
sn = fi.readline().split()
n = int(sn[0])
k = int(sn[1])
a = n * [0]
b = n * [0]
sa = fi.readline().split()
for i in range(n):
    b[i] = int(sa[i])
def timmin():
    tmp = round(1000000)
    for i in b:
        if (i < tmp):
            tmp = i
    return tmp
def timmax():
    tmp = -round(1000000)
    for i in b:
        if (i > tmp):
            tmp = i
    return tmp
rmin = timmin()
rmax = timmax()

def _xuli():
    p1 = 0
    p2 = 0
    global k

```

```

# for i in range(k):
#     print(a[i],end = " ", file = fo)
# print("\n",file = fo)
for i in range(k):
    if (b[a[i]] == rmax):
        p1 = 1
    if (b[a[i]] == rmin):
        p2 = 1
if (p1*p2 != 0):
    for i in range(k):
        print(b[a[i]], end = " ", file = fo)
    print("", file = fo)

def _lietke():
    for i in range(k):
        a[i] = i
    i = 0
    while (i >= 0):
        _xuli()
        i = k-1
        while (i >= 0) and (n - k + i == a[i]):
            i = i - 1
        if (i >= 0):
            a[i] = a[i] + 1
            for j in range (i + 1, k):
                a[j] = a[j-1] + 1
    _lietke()

```

27. DQCKN52.* Tập con K có độ chênh lệch nhỏ nhất

Hãy đưa ra các cách chọn K phần tử trong **dãy b** sao cho dãy được chọn có độ chênh lệch **nhỏ nhất**. Đưa ra các cách chọn: Ghi chỉ số của các phần tử được chọn và độ chênh lệch nhỏ nhất ($|a_{\max i} - a_{\min i}|$).

DQCKN52.INP	DQCKN52.OUT
5 3 1 8 2 1 3	1 2 1

Thuật toán:

Chọn K phần tử trong dãy b bằng cách: Sinh các cấu hình chứa chỉ số các phần tử được chọn là a[i].

Với mỗi cấu hình, tìm rmax, rmin.

Tính độ chênh lệch $|a_{\max} - a_{\min}| = r_{\max} - r_{\min}$

Cập nhật nhỏ nhất và lưu cấu hình.

Đưa các cấu hình ra và đưa giá trị nhỏ nhất ra.

Code C++

```
#include<bits/stdc++.h>

using namespace std;

int n,k,b[100000],a[100000],c[100000],g[1000][1000],res;
int amax = INT_MIN, amin = INT_MAX;

void in(){
    cin >> n >> k;
    for(int i = 1; i <= n; i++) cin >> b[i];
}

void out(){
    for(int i = 1; i <= d; i++){
        if(c[i] == res){
            for(int j = i + 1; j <= k; j++){
                if(g[i][j]) cout << b[j] << ' ';
                cout << "\n";
            }
        }
    }
    cout << res;
}

void go(){
    amax = max(amax, b[a[i]]); //for trên mảng
    amin = min(amin, b[a[i]]);
    res = min(res, amax - amin); // thay bằng if
    c[d] = amax - amin;
    d++;
    for(int i = 1; i <= d; i++) g[d][i] = a[i];
}
```

```

void sinh(){
    int i;
    for(int i = 1; i <= k; i++) a[i] = i;
    do{
        go(); i = k;
        while((i > 0) && (a[i] == n-k+1)) i--;
        if(i > 0){
            a[i]++;
            for(int j = i+1; j <= k; j++){
                a[j] = a[j-1]+1;
            }
        }
    }while(i != 0);
    out();
}

int main(){
    ios_base::sync_with_stdio(NULL);
    cin.tie(0); cout.tie(0);
    in();
    sinh();
    return 0;
}
}

#include <bits/stdc++.h>

using namespace std;

int n, k, a[24], b[24], Max = INT_MIN, Min = INT_MAX, t[10000][24], c[100000],
d = 0, res = INT_MAX;

void handle () {
    int Max = INT_MIN, Min = INT_MAX;
    for (int i = 1; i <= k; ++i) {
        Max = max (Max, b[a[i]]);
        Min = min (Min, b[a[i]]);
        t[d][i] = a[i];
    }
    if (Max - Min <= res) {

```

```

        res = Max - Min;
        c[d] = Max - Min;
        ++d;
    }
}

void gen () {
    for (int i = 1; i <= k; ++i) a[i] = i;
    int i;
    do {
        handle ();
        i = k;
        while ((i > 0) && (a[i] == n - k + i)) --i;
        if (i > 0) {
            ++a[i];
            for (int j = i + 1; j <= k; ++j) a[j] = a[j - 1] + 1;
        }
    } while (i > 0);
}

void output () {
    for (int i = 0; i < d; ++i) {
        if (c[i] == res) {
            for (int j = 1; j <= k; ++j) cout << b[t[i][j]] << ' ';
            cout << '\n';
        }
    }
}

int main () {
    ios_base::sync_with_stdio(NULL);
    cin.tie(NULL); cout.tie(NULL);
    cin >> n >> k;
    for (int i = 1; i <= n; ++i) cin >> b[i];
    gen ();
    output ();
    return 0;
}

```

}

Code Python

```

from math import sqrt
fi = open("CQCKN52.INP","r")
fo = open("CQCKN52.OUT","w")
sn = fi.readline().split()
n = int(sn[0])
k = int(sn[1])
a = n * [0]
b = n * [0]
c = n * [0]
sa = fi.readline().split()
for i in range(n):
    b[i] = int(sa[i])
res = 1000000
def _xuli():
    p1 = 0
    p2 = 0
    global k
    global res
    global c
    # for i in range(k):
    #     print(a[i],end = " ", file = fo)
    # print("\n",file = fo)
    rmin = 1000000
    rmax = -1000000
    for i in range(k):
        if (b[a[i]] > rmax):
            rmax = b[a[i]]
        if (b[a[i]] < rmin):
            rmin = b[a[i]]
    if (res > abs(rmax - rmin)):
        res = abs(rmax - rmin)
        for i in range(k):
            c[i] = b[a[i]]
def _lietke():
    for i in range(k):

```

```

    a[i] = i
    i = 0
    while (i >= 0):
        _xuli()
        i = k-1
        while (i >= 0) and (n - k + i == a[i]):
            i = i - 1
        if (i >= 0):
            a[i] = a[i] + 1
            for j in range (i + 1, k):
                a[j] = a[j-1] + 1
        _lietke()
    for i in range(k):
        print(c[i],end = " ",file = fo)

```

High:

Nhập N, K và dãy b_1, b_2, \dots, b_n , dãy c_1, c_2, \dots, c_N .

28. DQCKN6.* Bài toán cái túi 2

Hãy đưa ra các cách chọn **K phần tử** để **tổng các $b[i]$** được chọn $\leq M$ và **tổng $c[i]$** được chọn **lớn nhất**. Và đưa ra giá trị tổng $c[i]$ lớn nhất tìm được.

Thuật toán:

Chọn K phần tử trong dãy b bằng cách: Sinh các cấu hình chứa chỉ số các phần tử được chọn là $a[i]$. //mảng $a[]$ lưu chỉ số các phần tử được chọn.

Với mỗi cấu hình, tìm $tb += b[a[i]]$; $tc += c[a[i]]$.

Nếu $tb \leq M$ thì cập nhật res, lưu cấu hình.

Đưa các cấu hình ra và đưa giá trị lớn nhất ra.

Code C++

```

#include <bits/stdc++.h>

using namespace std;

int n, k, m, a[24], b[24], c[24], val[10000], t[10000][24], d = 0, res = INT_MIN;

void output () {
    for (int i = 0; i < d; ++i) {
        if (val[i] == res) {

```



```

        for (int j = 1; j <= k; ++j) cout << t[i][j] << ' ';
        cout << '\n';
    }
}
if (res > INT_MIN) cout << res;
    else cout << "Ko co cau hinh thoa man";
}

void handle () {
    int sumc = 0, sumb = 0;
    for (int i = 1; i <= k; ++i) {
        t[d][i] = a[i];
        sumc += c[a[i]];
        sumb += b[a[i]];
        if (sumb > m) return ;
    }
    if (sumc >= res) {
        res = sumc;
        val[d] = sumc;
        ++d;
    }
}

void gen () {
    for (int i = 1; i <= k; ++i) a[i] = i;
    int i;
    do {
        handle ();
        i = k;
        while ((i > 0) && (a[i] == n - k + i)) --i;
        if (i > 0) {
            ++a[i];
            for (int j = i + 1; j <= k; ++j) a[j] = a[j - 1] + 1;
        }
    } while (i > 0);
}

```

```

int main () {
    ios_base::sync_with_stdio(NULL);
    cin.tie(NULL); cout.tie(NULL);
    cin >> n >> k >> m;
    for (int i = 1; i <= n; ++i) cin >> b[i];
    for (int i = 1; i <= n; ++i) cin >> c[i];
    gen ();
    output ();
    return 0;
}

#include<bits/stdc++.h>
#define N 101
#define ll long long
#define h round(1e18+1)
using namespace std;
int a[N],b[N],c[N],d[N],f[N][N]; int n,w=0,k; ll sb,sc,m,res=-h;
void nhap(){
    cin>>n>>k>>m;
    for(int i=1; i<=n; i++) cin>>b[i];
    for(int i=1; i<=n; i++) cin>>c[i];
}
int timkiem(ll x){ //Tìm kiếm nhị phân: vị trí của x trong mảng D
    int l=1,r=w,mid;
    while(l<=r){
        mid=(l+r)/2;
        if(d[mid]==x){
            int i=mid-1;
            while(d[i]==d[mid]) i--;
            return i+1;
        }else if(d[mid]<x) l=mid+1;
        else r=mid-1;
    }
    return -1;
}
void xuat(){
    int i=timkiem(res);
    if(i<1) return;

```

```

        for( ; i<=w; i++){
            for(int j=1; j<=k; j++)
                if(f[i][j]) cout<<b[j]<<" "<<c[j]<<" ";
            cout<<"\n";
        }
    }
void tim(){
    sb=sc=0;
    for(int i=1; i<=k; i++){
        sb+=1LL*b[a[i]];
        sc+=1LL*c[a[i]];
        if(sb>m) return;
    }
    if(sc>=res){
        w++; res=sc; d[w]=res;
        for(int i=1; i<=k; i++) f[w][i]=a[i];
    }
}
void xuly(){
    int i;
    for(i=1; i<=k; i++) a[i]=i;
    do{
        tim(); i=k;
        while(i>0 && a[i]==n-k+i) i--;
        if(i>0){
            a[i]++;
            for(int j=i+1; j<=k; j++) a[j]=a[j-1]+1;
        }
    }while(i!=0);
}
int main(){
    ios_base::sync_with_stdio(0);
    cin.tie(0); cout.tie(0);
    nhap();
    xuly();
    xuat();
    return 0;
}

```

```

}
#include<bits/stdc++.h>

using namespace std;

int n,k,m,d,a[100000],b[100000],c[100000],g[1000][1000];
long long sb,sc,res = LL_MIN;

void in(){
    cin >> n >> k >> m;
    for(int i = 1; i <= n; i++) cin >> b[i];
    for(int i = 1; i <= n; i++) cin >> c[i];
}
void out(){
    for(int i = 1; i <= d; i++){
        for(int j = i + 1; j <= k; j++){
            if(g[i][j]) cout << b[j] << ' ';
            cout << '\n';
        }
        cout << res;
    }
}
void go(){
    sb = sc = 0;
    for(int i = 1; i <= k; i++){
        sb += 1LL*b[a[i]];
        sc += 1LL*c[a[i]];
        if(sb > m) return;
    }
    if(sb <= m){
        if(sc >= res){
            res = sc;
            d++;
            for(int i = 1; i <= d; i++) g[d][i] = a[i];
        }
    }
}
void sinh(){
    int i;

```

```

for(int i = 1; i <= k; i++) a[i] = i;
do{
    go(); i = k;
    while((i > 0) && (a[i] == n-k+1)) i--;
    if(i > 0){
        a[i]++;
        for(int j = i+1; j <= k; j++){
            a[j] = a[j-1]+1;
        }
    }
}while(i != 0);
out();
}
int main(){
    ios_base::sync_with_stdio(NULL);
    cin.tie(0); cout.tie(0);
    in();
    sinh();
    return 0;
}

```

Code Python

```

from math import sqrt
fi = open("CQCKN6.INP","r")
fo = open("CQCKN6.OUT","w")
sn = fi.readline().split()
n = int(sn[0])
k = int(sn[1])
m = int(sn[2])
a = n * [0]
b = n * [0]
c = n * [0]
sa = fi.readline().split()
for i in range(n):
    b[i] = int(sa[i])
sb = fi.readline().split()
for i in range(n):
    c[i] = int(sb[i])

```

```

res = -1000000
def _xuli():
    p1 = 0
    p2 = 0
    global k
    global res
    global c
    # for i in range(k):
    #     print(a[i],end = " ", file = fo)
    # print("\n",file = fo)
    rmin = 1000000
    rmax = -1000000
    for i in range(k):
        p1 += b[a[i]]
        p2 += c[a[i]]
        if (p1 < m) and (p2 > res):
            res = p2
def _lietke():
    for i in range(k):
        a[i] = i
    i = 0
    while (i >= 0):
        _xuli()
        i = k-1
        while (i >= 0) and (n - k + i == a[i]):
            i = i - 1
        if (i >= 0):
            a[i] = a[i] + 1
            for j in range (i + 1, k):
                a[j] = a[j-1] + 1
    _lietke()
print(res, end = "", file = fo)

```

29. DQCKN7.* Tập con K có tổng lớn nhất

Hãy đưa ra **các cách chọn đúng K** phần tử để tổng tất cả các $(b[i] + c[i])$ được chọn là lớn nhất. Đưa tổng lớn nhất của $b[i]$ và $c[i]$ ra.

DQCKN7.INP	DQCKN7.OUT
------------	------------

6 3	31
2 2 3 4 7 2	
5 1 3 1 6 9	

Thuật toán:

Chọn K phần tử trong dãy b bằng cách: Sinh các cấu hình chứa chỉ số các phần tử được chọn là a[i]. //mảng a[] lưu chỉ số các phần tử được chọn.

Với mỗi cấu hình, tìm $S += b[a[i]] + c[a[i]]$.

Nếu $res \leq S$ thì cập nhật res, lưu cấu hình.

Đưa các cấu hình ra và đưa giá trị lớn nhất ra.

Code C++

```
#include<bits/stdc++.h>
#define N 101
#define ll long long
#define h round(1e18+1)
using namespace std;
int a[N],b[N],c[N],d[N],f[N][N]; int n,w=0,k, ll s,res=-h;
void nhap(){
    cin>>n>>k;
    for(int i=1; i<=n; i++) cin>>b[i];
    for(int i=1; i<=n; i++) cin>>c[i];
}
int timkiem(ll x){
    int l=1,r=w,mid;
    while(l<=r){
        mid=(l+r)/2;
        if(d[mid]==x){
            int i=mid-1;
            while(d[i]==d[mid]) i--;
            return i+1;
        }else if(d[mid]<x) l=mid+1;
        else r=mid-1;
    }
    return -1;
}
void xuat(){
    int i=timkiem(res);
```

```

        if(i<1) return;
        for( ; i<=w; i++){
            for(int j=1; j<=k; j++)
                if(f[i][j]) cout<<b[j]<<" "<<c[j]<<" ";
            cout<<"\n";
        }
        cout<<res;
    }
    void tim(){
        s=0;
        for(int i=1; i<=k; i++) s+=1LL*(b[a[i]]+c[a[i]]);
        if(s>=res){
            w++; res=s; d[w]=res;
            for(int i=1; i<=k; i++) f[w][i]=a[i];
        }
    }
    void xuly(){
        int i;
        for(i=1; i<=k; i++) a[i]=i;
        do{
            tim(); i=k;
            while(i>0 && a[i]==n-k+i) i--;
            if(i>0){
                a[i]++;
                for(int j=i+1; j<=k; j++) a[j]=a[j-1]+1;
            }
        }while(i!=0);
    }
    int main(){
        ios_base::sync_with_stdio(0);
        cin.tie(0); cout.tie(0);
        nhap();
        xuly();
        xuat();
        return 0;
    }

```

```
#include<bits/stdc++.h>
```



```

using namespace std;

int n,k,d,a[100000],b[100000],c[100000],g[1000][1000];
long long sb, sc, res = LL_MIN;

void in(){
    cin >> n >> k;
    for(int i = 1; i <= n; i++) cin >> b[i];
    for(int i = 1; i <= n; i++) cin >> c[i];
}

void out(){
    for(int i = 1; i <= d; i++){
        for(int j = i + 1; j <= k; j++){
            if(g[i][j]) cout << b[j] << ' ';
            cout << "\n";
        }
        cout << res;
    }
}

void go(){
    sb = sc = 0;
    for(int i = 1; i <= k; i++){
        sb += 1LL*a[i]*b[i];
        sc += 1LL*a[i]*c[i];
    }

    res = max(res, sb+sc); //Tham khảo thêm phần lưu nghiệm của anh Kiên
    d++;
    for(int i = 1; i <= d; i++) g[d][i] = a[i];
}

void sinh(){
    int i;
    for(int i = 1; i <= k; i++) a[i] = i;
    do{
        go(); i = k;
        while((i > 0) && (a[i] == n-k+1)) i--;
        if(i > 0){

```

```

        a[i]++;
        for(int j = i+1; j <= k; j++){
            a[j] = a[j-1]+1;
        }
    }
} while(i != 0);
out();
}
int main(){
    ios_base::sync_with_stdio(NULL);
    cin.tie(0); cout.tie(0);
    in();
    sinh();
    return 0;
}

```

```
#include <bits/stdc++.h>
```

```
using namespace std;
```

```
int n, k, a[24], b[24], c[24], val[10000], t[10000][24], d = 0, res = INT_MIN;
```

```

void output () {
    for (int i = 0; i < d; ++i) {
        if (val[i] == res) {
            for (int j = 1; j <= k; ++j) cout << t[i][j] << ' ';
            cout << '\n';
        }
    }
    if (res > INT_MIN) cout << res;
    else cout << "Ko co cau hinh thoa man";
}

```

```

void handle () {
    int sumc = 0, sumb = 0;
    for (int i = 1; i <= k; ++i) {
        t[d][i] = a[i];
        sumc += c[a[i]];
    }
}

```

```

        sumb += b[a[i]];
    }
    if (sumc + sumb >= res) {
        res = sumc + sumb;
        val[d] = res;
        ++d;
    }
}

void gen () {
    for (int i = 1; i <= k; ++i) a[i] = i;
    int i;
    do {
        handle ();
        i = k;
        while ((i > 0) && (a[i] == n - k + i)) --i;
        if (i > 0) {
            ++a[i];
            for (int j = i + 1; j <= k; ++j) a[j] = a[j - 1] + 1;
        }
    } while (i > 0);
}

int main () {
    ios_base::sync_with_stdio(NULL);
    cin.tie(NULL); cout.tie(NULL);
    cin >> n >> k;
    for (int i = 1; i <= n; ++i) cin >> b[i];
    for (int i = 1; i <= n; ++i) cin >> c[i];
    gen ();
    output ();
    return 0;
}

```

Code Python

```

from math import sqrt
fi = open("CQCKN7.INP", "r")
fo = open("CQCKN7.OUT", "w")

```

```

#Doc du lieu vao
sn = fi.readline().split()
n = int(sn[0])
k = int(sn[1])
a = n * [0]
b = n * [0]
c = n * [0]
sa = fi.readline().split()
for i in range(n):
    b[i] = int(sa[i])
sb = fi.readline().split()
for i in range(n):
    c[i] = int(sb[i])
# Het doc du lieu

res = -1000000
# Voi moi cach chon, tim cach chon co tong lon nhat
def _xuli():
    p1 = 0
    global k
    global res
    for i in range(k):
        p1 += b[a[i]] + c[a[i]]
    res = max(res, p1)
# Giai thuat lien ke tap con K phan tu, luu vao mang a
def _lietke():
    for i in range(k):
        a[i] = i
    i = 0
    while (i >= 0):
        _xuli()
        i = k-1
        while (i >= 0) and (n - k + i == a[i]):
            i = i - 1
        if (i >= 0):
            a[i] = a[i] + 1
            for j in range (i + 1, k):

```

```
a[j] = a[j-1] + 1
```

```
_lietke()
```

```
print(res, end = "", file = fo)
```

Hard:

Nhập N và dãy b_1, b_2, \dots, b_n , dãy c_1, c_2, \dots, c_N , dãy d_1, d_2, \dots, d_N .

30. DQCKN8.* Bài toán cái túi 3

Hãy đưa ra các cách K phần tử để tổng $b[i]$ được chọn $\leq M$, tổng $d[i]$ được chọn $\leq V$ và tổng $c[i]$ được chọn là lớn nhất. Hãy đưa ra tổng S_c lớn nhất tìm được.

DQCKN8.INP	DQCKN8.INP
6 3 10 8	24
1 2 4 5 1 3	
7 2 8 9 1 6	
1 3 2 1 1 3	

Thuật toán:

Chọn K phần tử trong dãy b bằng cách: Sinh các cấu hình chứa chỉ số các phần tử được chọn là $a[i]$. //mảng $a[]$ lưu chỉ số các phần tử được chọn.

Với mỗi cấu hình, ta tính tổng: sb, sc, sd .

Nếu $sb \leq M$ và $sd \leq v$ thì cập nhật res với sc , lưu cấu hình.

Thêm nhánh cận khi tính sb và tính sc : nếu $sb > M$ hoặc $sd > v$ thì return luôn.

Đưa các cấu hình ra và đưa giá trị lớn nhất ra.

Code C++

```
#include<bits/stdc++.h>
```

```
using namespace std;
```

```
int n,k,m,v,x,a[100000],b[100000],c[100000],d[100000],g[1000][1000];
```

```
long long sb, sc, sd, res = LL_MIN;
```

```
void in(){
```

```
    cin >> n >> k >> m >> v;
```

```
    for(int i = 1; i <= n; i++) cin >> b[i];
```

```
    for(int i = 1; i <= n; i++) cin >> c[i];
```

```
    for(int i = 1; i <= n; i++) cin >> d[i];
```

```
}
```

```

void out(){
    for(int i = 1; i <= x; i++){
        for(int j = i + 1; j <= k; j++){
            if(g[i][j]) cout << b[j] << ' ';
            cout << '\n';
        }
        cout << res;
    }
}

void go(){
    sb = sc = sd = 0;
    for(int i = 1; i <= k; i++){
        sb += 1LL*b[a[i]];
        sc += 1LL*c[a[i]];
        sd += 1LL*d[a[i]];
        if((sb > m) || (sd > v)) return;
    }
    if((sb <= m) && (sd <= v) && (res >= sc)){
        res = sc;
        x++;
        for(int i = 1; i <= x; i++) g[x][i] = a[i];
    }
}

void sinh(){
    int i;
    for(int i = 1; i <= k; i++) a[i] = i;
    do{
        go(); i = k;
        while((i > 0) && (a[i] == n-k+1)) i--;
        if(i > 0){
            a[i]++;
            for(int j = i+1; j <= k; j++){
                a[j] = a[j-1]+1;
            }
        }
    }while(i != 0);
    out();
}

```

```

int main(){
    ios_base::sync_with_stdio(NULL);
    cin.tie(0); cout.tie(0);
    in();
    sinh();
    return 0;
}

#include <bits/stdc++.h>

using namespace std;

int n, k, m, v, a[24], b[24], c[24], d[24], val[10000], t[10000][24], dem = 0, res = INT_MIN;

void output () {
    for (int i = 0; i < dem; ++i) {
        if (val[i] == res) {
            for (int j = 1; j <= k; ++j) cout << t[i][j] << ' ';
            cout << '\n';
        }
    }
    if (res > INT_MIN) cout << res;
    else cout << "Ko co cau hinh thoa man";
}

void handle () {
    int sumc = 0, sumb = 0, sumd = 0;
    for (int i = 1; i <= k; ++i) {
        t[dem][i] = a[i];
        sumc += c[a[i]];
        sumb += b[a[i]];
        sumd += d[a[i]];
        if (sumb > m || sumd > v) return ;
    }
    if (sumc >= res) {
        res = sumc;
        val[dem] = sumc;
        ++dem;
    }
}

```

```

    }
}

void gen () {
    for (int i = 1; i <= k; ++i) a[i] = i;
    int i;
    do {
        handle ();
        i = k;
        while ((i > 0) && (a[i] == n - k + i)) --i;
        if (i > 0) {
            ++a[i];
            for (int j = i + 1; j <= k; ++j) a[j] = a[j - 1] + 1;
        }
    } while (i > 0);
}

int main () {
    ios_base::sync_with_stdio(NULL);
    cin.tie(NULL); cout.tie(NULL);
    cin >> n >> k >> m >> v;
    for (int i = 1; i <= n; ++i) cin >> b[i];
    for (int i = 1; i <= n; ++i) cin >> c[i];
    for (int i = 1; i <= n; ++i) cin >> d[i];
    gen ();
    output ();
    return 0;
}

#include<bits/stdc++.h>
#define N 101
#define ll long long
#define h round(1e18+1)
using namespace std;
int a[N],b[N],c[N],t[N],d[N],f[N][N]; int n,w=0,k; ll sb,sc,sd,m,v,res=-h;
void nhap(){
    cin>>n>>k>>m>>v;
    for(int i=1; i<=n; i++) cin>>b[i];

```



```

        for(int i=1; i<=n; i++) cin>>c[i];
        for(int i=1; i<=n; i++) cin>>d[i];
    }
    int timkiem(ll x){
        int l=1,r=w,mid;
        while(l<=r){
            mid=(l+r)/2;
            if(t[mid]==x){
                int i=mid-1;
                while(t[i]==t[mid]) i--;
                return i+1;
            }else if(t[mid]<x) l=mid+1;
            else r=mid-1;
        }
        return -1;
    }
    void xuat(){
        int i=timkiem(res);
        if(i<1) return;
        for( ; i<=w; i++){
            for(int j=1; j<=k; j++)
                if(f[i][j]) cout<<b[j]<<" "<<c[j]<<" "<<d[j]<<" ";
            cout<<"\n";
        }
        cout<<res;
    }
    void tim(){
        sb=sc=sd=0;
        for(int i=1; i<=k; i++){
            sb+=1ll*b[a[i]];
            sc+=1ll*c[a[i]];
            sd+=1ll*d[a[i]];
            if(sb>m || sd>v) return;
        }
        if(sc>=res){
            w++; res=sc; t[w]=res;
            for(int i=1; i<=k; i++) f[w][i]=a[i];
        }
    }
}

```

```

    }
}
void xuly(){
    int i;
    for(i=1; i<=k; i++) a[i]=i;
    do{
        tim(); i=k;
        while(i>0 && a[i]==n-k+i) i--;
        if(i>0){
            a[i]++;
            for(int j=i+1; j<=k; j++) a[j]=a[j-1]+1;
        }
    }while(i!=0);
}
int main(){
    ios_base::sync_with_stdio(0);
    cin.tie(0); cout.tie(0);
    nhap();
    xuly();
    xuat();
    return 0;
}

```

Code Python

```

from math import sqrt
fi = open("CQCKN8.INP","r")
fo = open("CQCKN8.OUT","w")
#Doc du lieu vao
sn = fi.readline().split()
n = int(sn[0])
k = int(sn[1])
m = int(sn[2])
v = int(sn[3])
a = n * [0]
b = n * [0]
c = n * [0]
d = n * [0]
sa = fi.readline().split()

```

```

for i in range(n):
    b[i] = int(sa[i])
sb = fi.readline().split()
for i in range(n):
    c[i] = int(sb[i])
sc = fi.readline().split()
for i in range(n):
    c[i] = int(sb[i])
# Het doc du lieu

res = -1000000
# Voi moi cach chon, tim cach chon co tong SC lon nhat
def _xuli():
    p1 = 0
    p2 = 0
    p3 = 0

    global k
    global m
    global v
    global res
    for i in range(k):
        p1 += b[a[i]]
        p2 += c[a[i]]
        p3 += d[a[i]]
    if (p1 > m) or (p3 > v):
        return
    # print(p2, ": ", a, file = fo)
    res = max(res, p2)
# Giai thuat lien ke tap con K phan tu, luu vao mang a
def _lietke():
    for i in range(k):
        a[i] = i
    i = 0
    while (i >= 0):
        _xuli()
        i = k-1

```

```

while (i >= 0) and (n - k + i == a[i]):
    i = i - 1
if (i >= 0):
    a[i] = a[i] + 1
    for j in range (i + 1, k):
        a[j] = a[j-1] + 1
_lietke()
print(res, end = "", file = fo)

```

31. DQCKN9.* Tập con K có tổng bằng P

Hãy đưa ra cách chọn K phần tử để tổng $b[i] \leq M$, tổng $d[i] \leq V$ và tổng $c[i]$ là bằng P.

DQCKN9.INP	DQCKN9.OUT
6 3 10 8 17	7 2 8
1 2 4 5 1 3	7 9 1
7 2 8 9 1 6	2 9 6
1 3 2 1 1 3	

Thuật toán:

Chọn K phần tử trong dãy b bằng cách: Sinh các cấu hình chứa chỉ số các phần tử được chọn là a[i]. //mảng a[] lưu chỉ số các phần tử được chọn.

Với mỗi cấu hình, ta tính: sb, sc, sd.

Nếu $sb \leq M$ và $sd \leq v$ và $sc = P$ thì cập nhật lưu cấu hình.

Thêm nhánh cận khi tính sb và tính sc.

Đưa các cấu hình ra.

Code C++

```

#include<bits/stdc++.h>

using namespace std;

int n, k, m, v, p, x, a[100000], b[100000], c[100000], d[100000], g[1000][1000];
long long sb, sc, sd, res = LL_MIN;

void in(){
    cin >> n >> k >> m >> v >> p;
    for(int i = 1; i <= n; i++) cin >> b[i];

```

```

        for(int i = 1; i <= n; i++) cin >> c[i];
        for(int i = 1; i <= n; i++) cin >> d[i];
    }
    void out(){
        for(int i = 1; i <= x; i++){
            for(int j = i + 1; j <= k; j++){
                if(g[i][j]) cout << b[j] << ' ';
                cout << "\n";
            }
            cout << res;
        }
    }
    void go(){
        sb = sc = sd = 0;
        for(int i = 1; i <= k; i++){
            sb += 1LL*a[i]*b[i];
            sc += 1LL*a[i]*c[i];
            sd += 1LL*a[i]*d[i];
            if((sb > m) || (sd > v) || (sc > p)) return;
        }
        if((sc == p) && (sb <= m) && (sd <= v)){
            //res = maxx(sb, sc, sd);
            x++;
            for(int i = 1; i <= x; i++) g[x][i] = a[i];
        }
    }
    void sinh(){
        int i;
        for(int i = 1; i <= k; i++) a[i] = i;
        do{
            go(); i = k;
            while((i > 0) && (a[i] == n-k+1)) i--;
            if(i > 0){
                a[i]++;
                for(int j = i+1; j <= k; j++){
                    a[j] = a[j-1]+1;
                }
            }
        }
    }

```

```

        }while(i != 0);
        out();
    }
int main(){
    ios_base::sync_with_stdio(NULL);
    cin.tie(0); cout.tie(0);
    in();
    sinh();
    return 0;
}

#include<bits/stdc++.h>
#define N 101
#define ll long long
using namespace std;
int a[N],b[N],c[N],d[N]; int n,k; ll sb,sc,sd,m,v,p;
void nhap(){
    cin>>n>>k>>m>>v>>p;
    for(int i=1; i<=n; i++) cin>>b[i];
    for(int i=1; i<=n; i++) cin>>c[i];
    for(int i=1; i<=n; i++) cin>>d[i];
}
void xuat(){
    for(int i=1; i<=k; i++)
        cout<<b[a[i]]<<" "<<c[a[i]]<<" "<<d[a[i]]<<" ";
    cout<<"\n";
}
void tim(){
    sb=sc=sd=0;
    for(int i=1; i<=k; i++){
        sb+=1ll*b[a[i]];
        sc+=1ll*c[a[i]];
        sd+=1ll*d[a[i]];
        if(sb>m || sd>v || sc>p) return;
    }
    if(sc==p) xuat();
}
void xuly(){

```

```

    int i;
    for(i=1; i<=k; i++) a[i]=i;
    do{
        tim(); i=k;
        while(i>0 && a[i]==n-k+i) i--;
        if(i>0){
            a[i]++;
            for(int j=i+1; j<=k; j++) a[j]=a[j-1]+1;
        }
    }while(i!=0);
}
int main(){
    ios_base::sync_with_stdio(0);
    cin.tie(0); cout.tie(0);
    nhap();
    xuly();
    return 0;
}

```

```

#include <bits/stdc++.h>

```

```

using namespace std;

```

```

int n, k, m, v, p, a[24], b[24], c[24], d[24], val[10000], t[10000][24], dem = 0, res = INT_MIN;

```

```

void output () {
    for (int i = 0; i < dem; ++i) {
        if (val[i] == res) {
            for (int j = 1; j <= k; ++j) cout << t[i][j] << ' ';
            cout << '\n';
        }
    }
    if (res > INT_MIN) cout << res;
    else cout << "Ko co cau hinh thoa man";
}

```

```

void handle () {

```

```

int sumc = 0, sumb = 0, sumd = 0;
for (int i = 1; i <= k; ++i) {
    t[dem][i] = a[i];
    sumc += c[a[i]];
    sumb += b[a[i]];
    sumd += d[a[i]];
    if (sumb > m || sumd > v) return ;
}
if (sumc != p) return ;
if (sumb + sumc + sumd >= res) {
    res = sumb + sumc + sumd;
    val[dem] = sumb + sumc + sumd;
    ++dem;
}
}

void gen () {
    for (int i = 1; i <= k; ++i) a[i] = i;
    int i;
    do {
        handle ();
        i = k;
        while ((i > 0) && (a[i] == n - k + i)) --i;
        if (i > 0) {
            ++a[i];
            for (int j = i + 1; j <= k; ++j) a[j] = a[j - 1] + 1;
        }
    } while (i > 0);
}

int main () {
    ios_base::sync_with_stdio(NULL);
    cin.tie(NULL); cout.tie(NULL);
    cin >> n >> k >> m >> v >> p;
    for (int i = 1; i <= n; ++i) cin >> b[i];
    for (int i = 1; i <= n; ++i) cin >> c[i];
    for (int i = 1; i <= n; ++i) cin >> d[i];
}

```



```

gen ();
output ();
return 0;
}

```

Code Python

```

from math import sqrt
fi = open("CQCKN9.INP","r")
fo = open("CQCKN9.OUT","w")
#Doc du lieu vao
sn = fi.readline().split()
n = int(sn[0])
k = int(sn[1])
m = int(sn[2])
v = int(sn[3])
p = int(sn[4])
a = n * [0]
b = n * [0]
c = n * [0]
d = n * [0]
sa = fi.readline().split()
for i in range(n):
    b[i] = int(sa[i])
sb = fi.readline().split()
for i in range(n):
    c[i] = int(sb[i])
sc = fi.readline().split()
for i in range(n):
    c[i] = int(sb[i])
# Het doc du lieu

res = -1000000
# Voi moi cach chon, tim cach chon co tong SC = P
def _xuli():
    p1 = 0
    p2 = 0
    p3 = 0
    global k

```

```

global m
global v
global res
for i in range(k):
    p1 += b[a[i]]
    p2 += c[a[i]]
    p3 += d[a[i]]
if (p1 > m) or (p3 > v):
    return
# print(p2, ": ", a, file = fo)
if (p2 == p):
    for i in range(k):
        print(c[a[i]], end = " ", file = fo)
    print("", file = fo)
# Giai thuat lien ke tap con K phan tu, luu vao mang a
def _lietke():
    for i in range(k):
        a[i] = i
    i = 0
    while (i >= 0):
        _xuli()
        i = k-1
        while (i >= 0) and (n - k + i == a[i]):
            i = i - 1
        if (i >= 0):
            a[i] = a[i] + 1
            for j in range (i + 1, k):
                a[j] = a[j-1] + 1
    _lietke()

```

LIỆT KÊ HOÁN VỊ

32. DQHV1.* Liệt kê hoán vị

Cho N. Hãy liệt kê các hoán vị của tập (1, 2, 3, ..., N).

Ví dụ:

DQHV1.INP	DQHV1.OUT
3	1 2 3 1 3 2 2 1 3 2 3 1 3 1 2 3 2 1

Thuật toán:

Các giá trị có thể gán cho $a[i]$ thuộc đoạn $[1, N]$.

Vậy ta thử điền vào ô $a[i]$ giá trị j thuộc đoạn $[1, N]$. Khi điền j thì ta đánh dấu j đã sử dụng.

Nếu điền đầy đủ N ô của mảng a thì ta thu được một cấu hình, ta đưa cấu hình đó ra.

Tiếp đến, giải phóng j để thử vào cấu hình khác.

Code C++

```
#include<bits/stdc++.h>
using namespace std;
int a[30]; int n;
void xuat(){
    for(int i=1; i<=n; i++) cout<<a[i]<<" ";
    cout<<"\n";
}
void xuly(){
    int i,k,d,c;
    for(i=1; i<=n; i++) a[i]=i;
    do{
        xuat(); i=n-1;
        while(i>0 && a[i]>a[i+1]) i--;
        if(i>0){
            k=n;
            while(k>0 && a[k]<a[i]) k--;
            swap(a[k],a[i]);
        }
    } while(1);
}
```

```

                d=i+1; c=n;
                while(d<c){
                    swap(a[d],a[c]);
                    d++; c--;
                }
            }
        }while(i!=0);
    }
int main(){
    ios_base::sync_with_stdio(0);
    cin.tie(0); cout.tie(0);
    freopen("dqhv1.inp","r",stdin);
    freopen("dqhv1.out","w",stdout);
    cin>>n;
    xuly();
    return 0;
}

```

```
#include <bits/stdc++.h>
```

```
using namespace std;
```

```
int n, a[24];
```

```
void show () {
    for (int i = 1; i <= n; ++i) cout << a[i] << ' ';
    cout << '\n';
}

```

```
void gen () {
    for (int i = 1; i <= n; ++i) a[i] = i;
    int i;
    do {
        show ();
        i = n - 1;
        while ((i > 0) && (a[i] > a[i + 1])) --i;
        if (i > 0) {
            int k = n;

```

```

        while (a[i] > a[k]) --k;
        swap (a[i], a[k]);
        int x = i + 1, y = n;
        while (x < y) {
            swap(a[x], a[y]);
            ++x; --y;
        }
    }
} while (i > 0);
}

```

```

int main () {
    ios_base::sync_with_stdio(NULL);
    cin.tie(NULL); cout.tie(NULL);
    freopen ("DQHV1.INP", "r", stdin);
    freopen ("DQHV1.OUT", "w", stdout);
    cin >> n;
    gen ();
    return 0;
}

```

```

#include<iostream>
#define nmax 100
using namespace std;
int n;
int b[nmax]={0};
int a[nmax];

void xuat()
{
    for(int i=1;i<=n;i++)
        cout<<a[i]<<" ";
    cout<<endl;
}

void thu(int k)
{
    for(int i = 1; i <= n; i++)
    {

```

```

        if (!b[i])
        {
            a[k]=i;
            b[i]=1;
            if(k==n)
                xuat();
            else
                thu(k);
            b[i]=0;
        }
    }
}

int main()
{
    cin>>n;
    thu(1);
}
#include<bits/stdc++.h>

using namespace std;

int n, i, k, p, q, ok = 1, a[100000];

void in(){
    cin >> n;
    for(i = 1; i <= n; i++) a[i] = i;
}

void out(){
    for(i = 1; i <= n; i++) cout << a[i] << ' ';
    cout << '\n';
}

void sinh(){
    i = n - 1;
    while((i > 0) && (a[i] > a[i+1])) i--;
    if(i > 0){
        k = n;

```

```

        while(a[k] < a[i]) k--;
        swap(a[k], a[i]);
        p = i + 1; q = n;
        while(p <= q){
            swap(a[p], a[q]);
            p++;
            q--;
        }
    }
    else (ok = 0);
}

void next(){
    while(ok){
        out();
        sinh();
    }
}

int main(){
    ios_base::sync_with_stdio(NULL);
    cin.tie(0); cout.tie(0);
    freopen("DQHV1.INP", "r", stdin);
    freopen("DQHV1.OUT", "w", stdout);
    in();
    next();
    return 0;
}

```

Code Pascal

```

var
    n, i, k, p, q: longint;
    a: array[-7..100007] of longint;
procedure swap(var u, v: longint);
var tam: longint;
begin
    tam := u;
    u := v;
    v := tam;
end;

```

```

procedure xuli();
var j: longint;
begin
  for j := 1 to n do write(a[j],#32);
  writeln;
end;
begin
  assign(input,'lkhv.inp');reset(input);
  assign(output,'lkhv.out');rewrite(output);
  readln(n);
  for i := 1 to n do a[i] := i;
  i := n - 1;
  while i > 0 do
    begin
      xuli;
      i := n - 1;
      while (i > 0) and (a[i] > a[i + 1]) do dec(i);
      if i > 0 then
        begin
          k := n;
          while a[k] < a[i] do dec(k);
          swap(a[k], a[i]);
          p := i + 1; q := n;
          while p < q do
            begin
              swap(a[p], a[q]);
              inc(p);
              dec(q);
            end;
          end;
        end;
      close(input);close(output);
    end.

```

Code Python

```

from math import sqrt
fi = open("DQHV1.INP","r")
fo = open("DQHV1.OUT","w")

```



```

#Doc du lieu vao
sn = fi.readline().split()
n = int(sn[0])
a = n * [0]
fre = n * [0]
for i in range(n):
    fre[i] = True
print(n, file = fo)
# Het doc du lieu
# Voi moi hoan vi ta dua no ra
def _xuli():
    global a
    print(a, file = fo)

# Giai thuat lien ke hoan vi, luu vao mang a
def _lietke(i):
    global fre
    for j in range(n):
        if (fre[j]):
            a[i] = j
            fre[j] = False
            if (i == n - 1):
                _xuli()
            else:
                _lietke(i+1)
            fre[j] = True
    _lietke(0)

```

33.DQHV2.* Số thứ tự của hoán vị

Cho N và một hoán vị của N (a_1, a_2, \dots, a_N). Hãy cho biết số thứ tự của hoán vị.

DQHV2.INP	DQHV2.OUT
3 2 1 3	3
5 3 5 2 4 1	70

Thuật toán:

Thuật toán sinh hoán vị sẽ sinh ra các hoán vị theo thứ tự từ điển.

Lần lượt sinh ra các hoán vị, đến hoán vị thứ res ta kiểm tra nó có bằng hoán vị đã cho hay không? Nếu có thì dừng và đưa res ra.

Ngược lại thì sinh hoán vị kế tiếp.

Thêm nhánh cận để thuật toán chạy nhanh hơn.

```
#include<bits/stdc++.h>
#define ll long long
using namespace std;
int a[30]; ll f[30]; bool kt[25]; int n; ll res=0;
void nhap(){
    cin>>n; f[1]=1;
    for(int i=1; i<=n; i++) cin>>a[i];
    for(int i=1; i<=n; i++) kt[i] = false;
    for(int i=2; i<=n; i++) f[i]=f[i-1]*(1ll*i);
}
ll tim(int u){
    ll ans=0;
    for(int i=1; i<=u; i++) kt[a[i]]=true;
    for(int i=1; i<=n; i++)
        if(!kt[i] && i<a[u]) ans++; //đếm các phần tử nhỏ hơn a[i] hiện tại và chưa
    sử dụng đến.
    return ans;
}
void xuly(){
    for(int i=1; i<n; i++) res+=tim(i)*f[n-i]; // công thức
    cout<<res+1;
}
int main(){
    ios_base::sync_with_stdio(0);
    cin.tie(0); cout.tie(0);
    freopen("dqhv1.inp","r",stdin);
    freopen("dqhv1.out","w",stdout);
    nhap();
    xuly();
    return 0;
}
```

```
#include <bits/stdc++.h>

using namespace std;

int n, a[24];
int b[24];

bool check () {
    for (int i = 1; i <= n; ++i) if (a[i] != b[i]) return false;
    return true;
}

int gen () {
    for (int i = 1; i <= n; ++i) a[i] = i;
    int i, d = 1;
    do {
        if (check()) return d;
        ++d;
        i = n - 1;
        while ((i > 0) && (a[i] > a[i + 1])) --i;
        if (i > 0) {
            int k = n;
            while (a[i] > a[k]) --k;
            swap (a[i], a[k]);
            int x = i + 1, y = n;
            while (x < y) {
                swap(a[x], a[y]);
                ++x; --y;
            }
        }
    } while (i > 0);
    return -1;
}

int main () {
    ios_base::sync_with_stdio(NULL);
    cin.tie(NULL); cout.tie(NULL);
```

```

freopen ("DQHV2.INP", "r", stdin);
freopen ("DQHV2.OUT", "w", stdout);
cin >> n;
for (int i = 1; i <= n; ++i) cin >> b[i];
cout << gen ();
return 0;
}

```

Code Python

```

from math import sqrt
fi = open("DQHV2.INP","r")
fo = open("DQHV2.OUT","w")
#Doc du lieu vao

sn = fi.readline().split()
n = int(sn[0])
sb = fi.readline().split()
b = n * [0]
for i in range(n):
    b[i] = int(sb[i]) - 1
# Het doc du lieu
a = n * [0]
res = 0
p = 0
fre = n * [0]
for i in range(n):
    fre[i] = True
# Voi moi hoan vi takiem tra no bang hoan vi da cho khong?
def _xuli():
    global a
    global res
    res += 1
    if (a == b):
        print(res,file = fo)
        p = 1
# Giai thuat lien ke hoan vi, luu vao mang a
def _lietke(i):
    global fre

```

```

global p
for j in range(n):
    if (fre[j]):
        a[i] = j
        fre[j] = False
        if (i == n - 1):
            _xuli()
            if (p == 1):
                return
        else:
            _lietke(i+1)
        fre[j] = True
_lietke(0)

```

34. DQHV3.* Hoán vị thứ K

Hãy tìm hoán vị thứ K của tập N.

DQHV3.INP	DQHV3.OUT
3 4	2 3 1

Thuật toán:

Lần lượt sinh ra các cấu hình, tăng res.

Nếu res = K thì đưa cấu hình đó ra.

Các cấu hình được đánh số thứ tự từ 1, 2, 3, N!

```

#include<bits/stdc++.h>
#define ll long long
using namespace std;
int a[30],b[30]; ll f[30]; bool kt[30]; int n,w=0; ll k,res=0; bool ok;
void nhap(){
    cin>>n>>k; f[1]=1;
    for(int i=2; i<=n; i++) f[i]=f[i-1]*(1ll*i); // số lượng hoán vị có i chữ số
}
void xuat(){
    for(int i=1; i<=n; i++) cout<<b[i]<<" ";
}

```

```

}
void xuly(){
    int j;
    for(int i=1; i<=n; i++){
        ok=true;
        for(j=1; j<=n; j++){
            if(!kt[j])
                if(k>f[n-i]) k=f[n-i];
            else{
                w++; b[i]=j; kt[j]=true; ok=false; break;
            }
        }
        if(ok)
            if(w==n-1){
                for(j=1; j<=n; j++){
                    if(!kt[j]){
                        b[i]=j; kt[j]=true; break;
                    }
                }
            }else{
                w++; b[i]=j-1; kt[j-1]=true;
            }
        }
    }
    xuat();
}
int main(){
    ios_base::sync_with_stdio(0);
    cin.tie(0); cout.tie(0);
    freopen("dqhv3.inp","r",stdin);
    freopen("dqhv3.out","w",stdout);
    nhap();
    xuly();
    return 0;
}

```

```
#include <bits/stdc++.h>
```

```
using namespace std;
```

```
int n, k, a[24];
```

```

void gen () {
    for (int i = 1; i <= n; ++i) a[i] = i;
    int i, d = 1;
    do {
        if (d == k) {
            for (int i = 1; i <= n; ++i) cout << a[i] << ' ';
            return ;
        }
        ++d;
        i = n - 1;
        while ((i > 0) && (a[i] > a[i + 1])) --i;
        if (i > 0) {
            int k = n;
            while (a[i] > a[k]) --k;
            swap (a[i], a[k]);
            int x = i + 1, y = n;
            while (x < y) {
                swap(a[x], a[y]);
                ++x; --y;
            }
        }
    } while (i > 0);
}

```

```

int main () {
    ios_base::sync_with_stdio(NULL);
    cin.tie(NULL); cout.tie(NULL);
    freopen ("DQHV3.INP", "r", stdin);
    freopen ("DQHV3.OUT", "w", stdout);
    cin >> n >> k;
    gen ();
    return 0;
}

```

Code Python

```

from math import sqrt
fi = open("DQHV3.INP","r")

```

```

fo = open("DQHV3.OUT","w")
#Doc du lieu vao

sn = fi.readline().split()
n = int(sn[0])
k = int(sn[1])

# Het doc du lieu
a = n * [0]
res = 0
p = 0
fre = n * [0]
for i in range(n):
    fre[i] = True
# Dua hoan vi thu k ra
def _xuli():
    for i in range(n):
        print(a[i]+1, end = " ", file = fo)

# Giai thuat lien ke hoan vi, luu vao mang a
def _lietke(i):
    global fre
    global p
    for j in range(n):
        if (fre[j]):
            a[i] = j
            fre[j] = False
            if (i == n - 1):
                p += 1
                if (p == k):
                    _xuli()
                    return
            else:
                _lietke(i+1)
            fre[j] = True
    _lietke(0)

```


35. DQHV4.* Hoán vị kế tiếp

Cho N và một hoán vị. Hãy tìm hoán vị kế tiếp của hoán vị đã cho.

4	1 3 2
1 2 3	

Thuật toán:

Bài toán 1.1: Cho hoán vị, tìm vị trí của hoán vị đã cho (res).

Bài toán 1.2: Tìm hoán vị kế tiếp (có thứ tự là: res + 1). Cho K tìm hoán vị thứ K của tập N.

Code C++

```
#include<bits/stdc++.h>
#define ll long long
using namespace std;
int a[30],b[30]; ll f[30]; bool kt1[30],kt2[30]; int n,w=0; ll k,res=0; bool ok;
void nhap(){
    cin>>n; f[1]=1;
    for(int i=1; i<=n; i++) cin>>a[i];
    for(int i=2; i<=n; i++) f[i]=f[i-1]*(1ll*i);
}
void xuat(){
    for(int i=1; i<=n; i++) cout<<b[i]<<" ";
}
ll tim(int u){
    ll ans=0;
    for(int i=1; i<=u; i++) kt1[a[i]]=true;
    for(int i=1; i<=n; i++)
        if(!kt1[i] && i<a[u]) ans++;
    return ans;
}
void xuly1(){
    for(int i=1; i<n; i++) res+=tim(i)*f[n-i];
    res+=2;
}
void xuly2(){
    int j;
    for(int i=1; i<=n; i++){
        ok=true;
        for(j=1; j<=n; j++)
            if(!kt2[j])
```

```

        if(res>f[n-i]) res-=f[n-i];
        else{
            w++; b[i]=j; kt2[j]=true; ok=false; break;
        }
    if(ok)
        if(w==n-1){
            for(j=1; j<=n; j++)
                if(!kt2[j]){
                    b[i]=j; kt2[j]=true; break;
                }
            }else{
                w++; b[i]=j-1; kt2[j-1]=true;
            }
        }
    xuất();
}
int main(){
    ios_base::sync_with_stdio(0);
    cin.tie(0); cout.tie(0);
    freopen("dqhv4.inp","r",stdin);
    freopen("dqhv4.out","w",stdout);
    nhap();
    xuly1();
    xuly2();
    return 0;
}

```

36. DQHV5.*

Cho N và dãy a_1, a_2, \dots, a_N . Thực hiện đổi chỗ một số phần tử sao cho tổng.

$S = |a_1 - a_2| + |a_2 - a_3| + \dots + |a_{n-1} - a_n|$ đạt giá trị nhỏ nhất.

Input: từ tệp DQHV5.INP gồm

- Dòng 1: Ghi số nguyên dương N
- Dòng 2: Ghi N số nguyên a_i .

Output: ghi ra tệp DQHV5.OUT gồm:

- Dòng 1: Ghi giá trị nhỏ nhất của S tìm được.
- Dòng 2: Ghi một cách sắp xếp dãy đã cho để có S nhỏ nhất.

Thuật toán:

Sinh các hoán vị vào mảng $b[i]$ của tập $\{1, 2, \dots, N\}$. Mảng $b[i]$ là mảng chỉ số các phân tử. Với mỗi cấu hình, truy cập $a[b[1]], a[b[2]], \dots, a[b[n]]$.

For (1, N) $S += |a[b[i]] - a[b[i+1]]|$;

Nhánh cận $S > res$ thì return sinh cấu hình kế tiếp.

Cập nhật lại res và lưu cấu hình.

```
#include<bits/stdc++.h>
#define h round(1e18+1)
#define ll long long
using namespace std;
int a[30],d[30],b[30]; int n; ll res=h,tmp;
void nhap(){
    cin>>n;
    for(int i=1; i<=n; i++) cin>>a[i];
}
void xuat(){
    cout<<res<<"\n";
    for(int i=1; i<=n; i++) cout<<d[i]<<" ";
}
void tim(){
    tmp=0;
    for(int i=2; i<=n; i++) tmp+=1ll*abs(a[b[i]]-a[b[i-1]]);
    if(res>tmp){
        res=tmp;
        for(int i=1; i<=n; i++) d[i]=a[b[i]];
    }
}
void xuly(){
    int i,k,d,c;
    for(i=1; i<=n; i++) b[i]=i;
    do{
        tim(); i=n-1;
        while(i>0 && b[i]>b[i+1]) i--;
        if(i>0){
            k=n;
            while(k>0 && b[k]<b[i]) k--;
            swap(b[k],b[i]);
            d=i+1; c=n;
        }
    } while(1);
}
```

```

        while(d<c){
            swap(b[d],b[c]);
            d++; c--;
        }
    }
} while(i!=0);
}
int main(){
    ios_base::sync_with_stdio(0);
    cin.tie(0); cout.tie(0);
    freopen("dqhv5.inp","r",stdin);
    freopen("dqhv5.out","w",stdout);
    nhap();
    xuly();
    xuất();
    return 0;
}

```

37.DQHV6.*

Cho N và dãy a_1, a_2, \dots, a_N . Thực hiện đổi chỗ một số phần tử sao cho tổng.

$S = |a_1 - a_2 + a_3| + |a_2 - a_3 + a_4| + \dots + |a_{n-2} - a_{n-1} + a_n|$ đạt giá trị lớn nhất.

Input: từ tệp DQHV6.INP gồm

- Dòng 1: Ghi số nguyên dương N
- Dòng 2: Ghi N số nguyên a_i .

Output: ghi ra tệp DQHV6.OUT gồm:

- Dòng 1: Ghi giá trị nhỏ nhất của S tìm được.
- Các dòng tiếp theo, mỗi dòng ghi một cách sắp xếp dãy đã cho để có S lớn nhất.

THUẬT TOÁN QUAY LUI

Mô hình thuật toán quay lui:

```

procedure quaylui(i: longint);
{

    For (mọi giá trị V có thể gán cho a[i])
    {
        Thử gán a[i] := V;
        Nếu a[i] là phần tử cuối cùng thì gọi thủ tục Xuli()
        Ngược lại,
            { // ghi nhận việc đã sử dụng giá trị V cho a[i];
              gọi thủ tục quaylui(i + 1) ;
              // bỏ ghi nhận việc đã sử dụng giá trị V và thử giá trị khác cho a[i];
            }
        }
    }
}

void quaylui(int i)
{

    For (mọi giá trị V có thể gán cho a[i])
    {
        Thử gán a[i] = V;
        Nếu a[i] là phần tử cuối cùng thì gọi thủ tục Xuli()
        Ngược lại,
            { // ghi nhận việc đã sử dụng giá trị V cho a[i];
              gọi thủ tục quaylui(i + 1) ;
              // bỏ ghi nhận việc đã sử dụng giá trị V và thử giá trị khác cho a[i];
            }
        }
    }
}

```

38.QLBIN1.*

Cho N. Liệt kê các dãy nhị phân có độ dài N.

Thuật toán:

Tổ chức thủ tục xử lí cấu hình thu được.

Thử điền giá trị 0 hoặc 1 vào vị trí thứ I của mảng a.

Nếu điền đủ N phần tử thì xử lí cấu hình thu được.

```
#include <bits/stdc++.h>
#define nmax 107
using namespace std;
int a[nmax], n;
void xl()
{
    for(int i=1;i<=n;i++) cout<<a[i]<<" ";
    cout<<"\n";
}
void thu(int i)
{
    for(int j=0;j<=1;j++)
    {
        a[i]=j;
        if(i==n) xl();
        else thu(i+1);
    }
}
int main()
{
    cin>>n;
    thu(1);
    return 0;
}
```

39. QLBIN2.*

Liệt kê dãy nhị phân có số lượng số 0 bằng số lượng số 1.

Thuật toán:

Tổ chức thủ tục xử lí các cấu hình, đếm số lượng bit 0.

Thêm nhánh cận: Nếu số lượng bit 0 nhiều hơn $N/2$ thì dừng và chuyển sang sinh cấu hình kế tiếp.

Nếu số lượng bit 0: $dem*2 == N$ thì đưa cấu hình đó ra.

Tổ chức thủ tục quay lui sinh các cấu hình.

```
#include <bits/stdc++.h>
#define nmax 107
```

```

using namespace std;
int a[nmax], n,i;
void xuli()
{
    int d1=0, d2=0;
    for(int i=1;i<=n;i++)
    {
        if (a[i]==0) d1++;
        else d2++;
    }
    if (d1==d2)
    {
        for(int i=1;i<=n;i++)
        cout << a[i] << " ";
        cout << endl;
    }
}
void thu(int i)
{
    for(int j=0;j<=1;j++)
    {
        a[i]=j;
        if(i==n)
            xuli();
        else thu(i+1);
    }
}
int main()
{
    cin>>n;
    if (n%2==0)
        thu(1);
    return 0;
}

```

40. QLCKN1.*

Liệt kê các tập con có K phần tử của tập N phần tử.

Thuật toán:

Với phần tử thứ i : giá trị cần thử thuộc đoạn $[a[i-1] + 1; N-K+1]$

Hay $A[i-1] + 1 \leq A[i] \leq N - K + i$;

Khi gán đủ K giá trị thì đưa cấu hình ra.

```

var
  n, i, k: longint;
  a: array[-7..100007] of longint;
procedure xuli();
var p: longint;
begin
  for p := 1 to k do write(a[p],#32);
  writeln();
end;
procedure thu(i: longint);
var j: longint;
begin
  for j := a[i - 1] + 1 to n -
    k + i do
    begin
      a[i] := j;
      if i = k then xuli
      else thu(i + 1);
    end;
end;
begin
  assign(input,'qlbin3.inp');reset(input);
  assign(output,'qlbin3.out');rewrite(output);
  readln(n, k);
  a[0] := 0;
  thu(1);
  close(input);close(output);
end.

```

```
#include <bits/stdc++.h>
```

```
using namespace std;
```

```
int a[101];
```



```

int n, k;

void xuat()
{
    for(int i = 1 ; i <= k ; i++) cout << a[i] << " ";
    cout << "\n";
}

void xuly(int i)
{
    for(int j = a[i-1]+1 ; j <= n-k+i ; j++)
    {
        a[i] = j;
        if(i == k) xuat();
        else xuly(i+1);
    }
}

int main()
{
    freopen("DQCKN1.inp", "r", stdin);
    freopen("DQCKN1.out", "w", stdout);
    ios_base::sync_with_stdio(0);
    cin.tie(0); cout.tie(0);
    cin >> n >> k;
    a[0] = 0;
    xuly(1);
    return 0;
}

```

41.QLHV1.*

Cho N. Hãy liệt kê các hoán vị của tập (1, 2, 3, ...N).

N = 3

1 2 3

1 3 2

2 1 3

2 3 1

3 1 2

3 2 1

Thuật toán: $O(N!)$

- Tập tất cả các giá trị của tập N được sắp xếp theo thứ tự khác nhau.
- Sử dụng mảng đánh dấu các phần tử chưa sử dụng: `int fre[100];`
- Khởi tạo mảng `fre[i] = 1` với i từ 1 tới N .
- `fre[i] = 1` thì i chưa sử dụng/còn free; ngược lại `fre[i] = 0` thì i đã sử dụng.
- ở ô thứ i thì có thể nhận giá trị $[1, N]$.
- For ($j, 1, N$)
 Nếu `fre[j] = 1` thì
 {

 Nếu đã điền đủ N phần tử thì gọi thủ tục xử lí ();

 Ngược lại

 { Đánh dấu j đã sử dụng ~ `fre[j] = 0;`
 `quaylui(i+1);`
 giải phóng j : `fre[j] = 1;`
 }

}

```
#include<bits/stdc++.h>
#define nmax 100
using namespace std;
int n;
int fre[nmax]={0};
int a[nmax];

void xl()
{
    for(int i=1;i<=n;i++)
        cout<<a[i]<<" ";
    cout<<"\n";
}

void thu(int i)
{
    for(int j=1;j<=n;j++)
    {
        if(fre[j]==1)
        {
            a[i]=j;
```

```

        if(i==n)
            xl();
        else
        {
            fre[j]=0;
            thu(i+1);
            fre[j]=1;
        }
    }
}
int main()
{
    cin>>n;
    for(int i=1;i<=n;i++)
        fre[i]=1;
    thu(1);
    return 0;
}

```

42. QLCHKN.*

Liệt kê các chỉnh hợp không lặp chập K phần tử của N.

Thuật toán: $O(N!)$

- Tập tất cả các giá trị của tập N được sắp xếp theo thứ tự khác nhau.
- Sử dụng mảng đánh dấu các phần tử chưa sử dụng: int fre[100];
- Khởi tạo mảng fre[i] = 1 với i từ 1 tới N.
- fre[i] = 1 thì i chưa sử dụng/còn free; ngược lại fre[i] = 0 thì i đã sử dụng.
- ở ô thứ i thì có thể nhận giá trị [1,N].
- For (j,1,N)

Nếu fre[j] = 1 thì

{

Nếu đã điền đủ K phần tử thì gọi thủ tục xử lí ();

Ngược lại

{ Đánh dấu j đã sử dụng ~ fre[j] = 0;
quaylui(i+1);

```

        giải phóng j: fre[j] = 1;
    }
}

```

```

#include<bits/stdc++.h>
#define nmax 100
using namespace std;
int n,k;
int fre[nmax]={0};
int a[nmax];

void xl()
{
    for(int i=1;i<=k;i++)
        cout<<a[i]<<" ";
    cout<<"\n";
}

void thu(int i)
{
    for(int j=1;j<=n;j++)
    {
        if(fre[j]==1)
        {
            a[i]=j;
            if(i==k)
                xl();
            else
            {
                fre[j]=0;
                thu(i+1);
                fre[j]=1;
            }
        }
    }
}

int main()
{

```

```

        cin>>n >>k;
        for(int i=1;i<=k;i++)
            fre[i]=1;
        thu(1);
        return 0;
    }

```

43. QLPTS.*

Cho N. Liệt kê các cách phân tích N thành tổng các số nguyên dương. Các cách phân tích có hoán vị giống nhau thì tính là một cách.

Thuật toán:

- $A[i-1] \leq A[i] \leq (N-T[i-1]) / 2$
- Phần tử cuối cùng $a[i] = N - T[i-1]$
- Cấu hình có I phần tử.

```

#include<bits/stdc++.h>
#define nmax 32
using namespace std;
int a[nmax],t[nmax],n; // a[nmax] = x[nmax] như trong vd :)

void xl(int k)
{
    for(int i=1;i<=k;i++) cout<<a[i]<<" ";
    cout<<"\n";
}

void thu(int i)
{
    for(int j=a[i-1];2*j<=n-t[i-1];j++)
    {
        a[i]=j;
        t[i]=t[i-1]+j;
        thu(i+1);
    }
    a[i]=n-t[i-1];
    xl(i);
}

int main()
{

```

```

    cin>>n;
    a[0]=1; t[0]=0;
    thu(1);
    return 0;
}

```

```

var
  n, i: longint;
  a, t: array[-7..100007] of longint;
procedure xuli(k: longint);
var p :longint;
begin
  for p := 1 to k do write(a[p],#32);
  writeln();
end;
procedure thu(i: longint);
var j: longint;
begin
  for j := a[i - 1] to (n - t[i - 1]) div 2 do
    begin
      a[i] := j;
      t[i] := t[i - 1] + j;
      thu(i + 1);
    end;
  a[i] := n - t[i - 1];
  xuli(i);
end;
begin
  assign(input,'qlpts.inp');reset(input);
  assign(output,'qlpts.out');rewrite(output);
  readln(n);
  a[0] := 1;t[0] := 0;
  thu(1);
  close(input);close(output);
end.

```

44. QLPTSPR.*

Cho N. Liệt kê các cách phân tích N thành *tổng các số nguyên tố khác nhau đôi một*.

Các cách phân tích có hoán vị giống nhau thì tính là một cách.

$$N = 7 = 2 + 5$$

Thuật toán:

Sub1: Giả sử $N = C_1 + C_2 + \dots + C_j$ với dãy C_i là các số nguyên tố đôi một khác nhau.

Dãy tăng C_1, C_2, \dots, C_j là các số nguyên tố $< N$.

- Sàng nguyên tố tới $N - 1$ để được mảng $F[i] = \text{true/false}$;
- Duyệt $(1, N)$ nếu $F[i] = \text{true}$ thì đẩy i vào mảng C có K phần tử.

$N = 20$ thì mảng $C \{2, 3, 5, 7, 11, 13, 17, 19\}$

Bài toán quy về: Cho mảng C có K phần tử, hãy chọn một số các phần tử để có tổng bằng N .

Áp dụng phương pháp sinh dãy nhị phân có độ dài K vào mảng b . Với mỗi cấu hình ta thực hiện: *void xuli()*

For(1,K) nếu $b[i] = 1$ thì

{

$S += C[i]$;

Thêm nhánh cận, nếu $S > N$ thì return;

}

Nếu $S == N$ thì

{ //đưa các phần tử được chọn ra.

For(1,K) nếu $b[i] = 1$ thì cout << $c[i]$ << " ";

Cout << endl;

}

ở chương trình chính:

Đọc N ;

Sangnt($N-1$);

Duyệt $(1, N)$ nếu $F[i] = \text{true}$ thì đẩy i vào mảng C có K phần tử.

Quaylui(1);

// sinh dãy nhị phân vào mảng B , khi sinh đủ K phần tử thì gọi thủ tục xuli()

Sub2: Làm tương tự **bài 6 – Liệt kê các cách phân tích N thành tổng các số nguyên dương**.

- Sàng nguyên tố tới $N-1$ thu được mảng $F[i]$.

void xl(int u) //cải tiến thủ tục xử lí

{

//kiểm tra mảng a có u phần tử có phải là mảng có u phần tử là số nguyên tố đôi một khác nhau.

$k = 0$; // số lượng phần tử của mảng C chứa các số nguyên tố khác nhau của a

```

for(int i=1;i<=u;i++)
    nếu f[a[i]] == true thì
        Nếu a[i] != c[k-1] thì  đẩy a[i] vào mảng C có K phần tử;
        Nếu K = u thì đưa mảng C ra; //mảng a chỉ chứa các số nguyên tố khác nhau, đây là
        một nghiệm.
    }

```

45. QLQHAU.*

Cho bàn cờ kích thước $N \times N$. Hãy đặt N quân hậu lên bàn cờ để không quân hậu nào ăn được nhau. Đưa ra các cách sắp xếp các quân hậu.

Thuật toán:

Mảng $a[i] = \text{true}$ nếu hàng i chưa bị chiếm.

Mảng $c[i+j] = \text{true}$ nếu đường chéo 1 chưa bị chiếm.

Mảng $d[i-j] = \text{true}$ nếu đường chéo 2 chưa bị chiếm.

Tư tưởng quay lui:

- Thử đặt quân hậu vào hàng thứ i
 Nếu điền đủ N quân hậu thì thực hiện xuli()
 Ngược lại
 - Nếu hàng i chưa bị chiếm thì đánh dấu hàng i , chéo $i+j$, chéo $i-j$ bị chiếm và thử điền quân hậu thứ $i + 1$;
 - Giải phóng hàng i , chéo $i+j$, chéo $i-j$ để tìm cấu hình kế tiếp.

46. QLBIN3.*

Cho N và dãy a_1, a_2, \dots, a_N . Hãy liệt kê tất cả các tập con của dãy.

Ví dụ: $N = 3$ và dãy 3, 5, 6.

Kết quả:

3; 5; 6; 3 5; 3 6; 5 6; 3 5 6;

Thuật toán:

Áp dụng phương pháp sinh dãy nhị phân 0/1 có độ dài N .

Với mỗi cấu hình thu được, ta duyệt $(1, N)$ nếu $b[i] = 1$ thì đưa $a[i]$ ra.

47. QLHVHT.*

Một dãy a_1, a_2, \dots, a_N gọi là **một hoán vị hoàn toàn** của tập $\{1, 2, \dots, N\}$ nếu nó là một hoán vị và thoả mãn $a[i] \neq i$.

Cho N . Hãy liệt kê các hoán vị hoàn toàn của tập N phần tử.

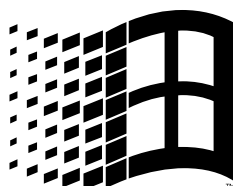
Ví dụ: $N = 3$

~~1-2-3~~; ~~1-3-2~~; ~~2-1-3~~; **2 3 1**; **3 1 2**; ~~3-2-1~~

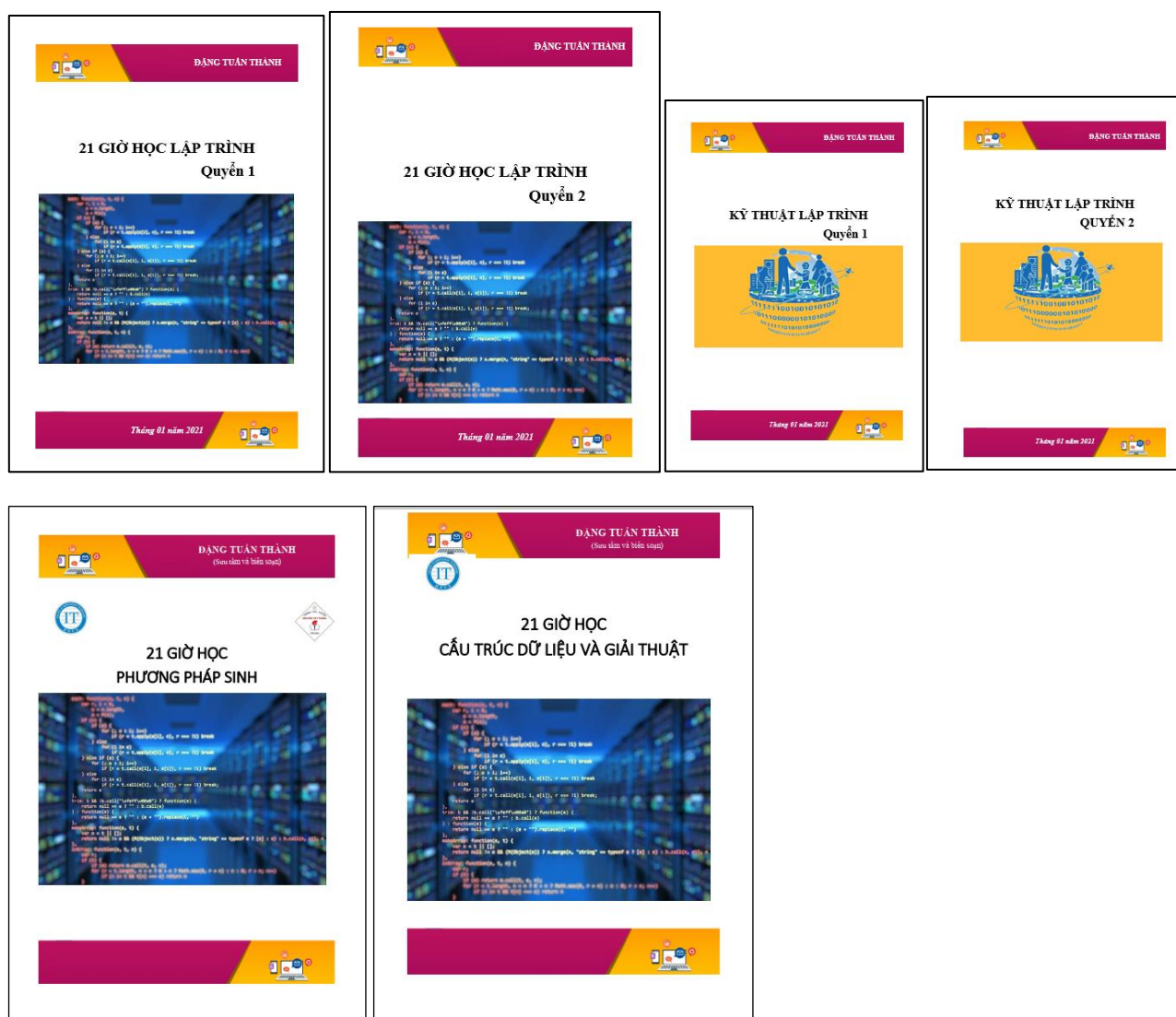
Kết quả: 2 3 1; 3 1 2

Thuật toán:

“Mỗi ngày tôi chọn một niềm vui,
Chọn những bông hoa, chọn những nụ cười...”



Các bạn tìm đọc tài liệu của cùng tác giả:



Zalo hỗ trợ: 0854518333