

```
/*
1. Generarea produsului cartezian pentru multimi standard.
*/
```

```
#include <iostream>
```

```
using namespace std;
```

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

```
void produs_cartezian(int m[], int n) {
    int c[101], k;
    for(int i = 1; i <= n; i++)
        c[i] = 1;
    afisare(c,n);
    do{
        k = n;
        while(c[k] == m[k] && k > 0) k--;
        if(k > 0) {
            c[k]++;
            for(int i = k + 1; i <= n; i++)
                c[i] = 1;
            afisare(c,n);
        }
    } while(k > 0);
}
```

```
int main() {
    int m[101], n;
    cin >> n;
    for(int i = 1; i <= n; i++)
        cin >> m[i];
    produs_cartezian(m,n);
    return 0;
}
```

```
/*
2. Generarea produsului cartezian pentru multimi oarecare.
*/
```

```
#include <iostream>
```

```
using namespace std;
```

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

```
void afisare(int c[], int a[][101], int n) {
    for(int i = 1; i <= n; i++)
        cout << a[i][c[i]] << ' ';
```

```

        cout << '\n';
    }

void produs_cartezian(int m[], int n) {
    int c[101], k;
    for(int i = 1; i <= n; i++)
        c[i] = 1;
    afisare(c,a,n);
    do{
        k = n;
        while(c[k] == m[k] && k > 0) k--;
        if(k > 0) {
            c[k]++;
            for(int i = k + 1; i <= n; i++)
                c[i] = 1;
            afisare(c,a,n);
        }
    } while(k > 0);
}

int main() {
    int m[101], n;
    cout << "Nr multimi: ";
    cin >> n;
    cout << "Nr elemente multimi: ";
    for(int i = 1; i <= n; i++)
        cin >> m[i];
    cout << "Elementele multimilor: ";
    for(int i = 1; i <= n; i++)
        for(int j = 1; j <= m[i]; j++)
            cin >> a[i][j];
    produs_cartezian(m,n);
    return 0;
}
/*
3. Generarea submultimilor unei multimi standard.
*/

```

```
#include <iostream>
```

```
using namespace std;
```

```

void afisare(int c[], int a[], int n) {
    for(int i = 1; i <= n; i++)
        if(c[i] == 1)
            cout << a[i] << ' ';
    cout << '\n';
}

```

```

void submultimi(int a[], int n) {
    int c[101], k;
    for(int i = 1; i <= n; i++)
        c[i] = 1;
    afisare(c, a, n);
}

```

```

    do {
        k = n;
        while((c[k] == 2) && (k >= 0)) k--;
        if(k>=0) {
            c[k] = 2;
            for(int i = k + 1; i <= n; i++)
                c[i] = 1;
            afisare(c,a,n);
        }
    } while(k > 0);
}

int main() {
    int a[101], n;
    cin >> n;
    for(int i = 1; i <= n; i++)
        a[i] = i;
    submultimi(a,n);
    return 0;
}
/*
4. Generarea submultimilor unei multimi oarecare.
*/

#include <iostream>

using namespace std;

void afisare(int c[], int a[], int n) {
    for(int i = 1; i <= n; i++)
        if(c[i] == 1)
            cout << a[i] << ' ';
    cout << '\n';
}

void submultimi(int a[], int n) {
    int c[101], k;
    for(int i = 1; i <= n; i++)
        c[i] = 1;
    afisare(c, a, n);
    do {
        k = n;
        while((c[k] == 2) && (k >= 0)) k--;
        if(k>=0) {
            c[k] = 2;
            for(int i = k + 1; i <= n; i++)
                c[i] = 1;
            afisare(c,a,n);
        }
    } while(k > 0);
}

int main() {
    int a[101], n;

```

```

    cin>>n;
    for(int i = 1; i <= n; i++)
        cin >> a[i];
    submultimi(a,n);
    return 0;
}
/*
5. Generarea aranjamentelor cu repetitie pentru multimi standard.
*/

```

```

#include <iostream>

using namespace std;

void afisare(int c[], int n) {
    for(int i = 1; i <= n; i++)
        cout << c[i] << ' ';
    cout << '\n';
}

void aranjamente_cu_repetitie(int a[], int n, int m) {
    int c[101], k;
    for(int i = 1; i <= n; i++)
        c[i] = 1;
    afisare(c,n);
    do{
        k = n;
        while(c[k] == m && k > 0) k--;
        if(k > 0) {
            c[k]++;
            for(int i = k + 1; i <= n; i++)
                c[i] = 1;
            afisare(c,n);
        }
    } while(k > 0);
}

int main() {
    int a[101], n, m;
    cin >> n >> m;
    aranjamente_cu_repetitie(a, n, m);
    return 0;
}
/*
6. Generarea aranjamentelor cu repetitie pentru multimi oarecare.
*/

```

```

#include <iostream>

using namespace std;

void afisare(int c[], int a[], int n) {
    for(int i = 1; i <= n; i++)
        cout << a[c[i]] << ' ';
}

```

```

        cout << '\n';
    }

    void aranjamente_cu_repetitie(int a[], int n, int m) {
        int c[101], k;
        for(int i = 1; i <= n; i++)
            c[i] = 1;
        afisare(c, a, n);
        do{
            k = n;
            while(c[k] == m && k > 0) k--;
            if(k > 0) {
                c[k]++;
                for(int i = k + 1; i <= n; i++)
                    c[i] = 1;
                afisare(c, a, n);
            }
        } while(k > 0);
    }

    int main() {
        int a[101], n, m;
        cin >> n >> m;
        for(int i = 1; i <= m; i++)
            cin >> a[i];
        aranjamente_cu_repetitie(a, n, m);
        return 0;
    }

```