

2. Generarea aranjamentelor

Se citesc două numere naturale n și c , cu $1 \leq c \leq n$. Se cere să se tipărească toate aranjamentele de n elemente luate câte c , în ordine lexicografică. **De exemplu**, pentru $n=3$ și $c=2$ avem
1 2, 1 3, 2 1, 2 3, 3 1, 3 2.

Definiție: Fie A o mulțime cu n elemente. **Aranjamentele** de n elemente luate câte c sunt toate submulțimile ordonate ale lui A care au fiecare câte c elemente.

Reprezentarea soluției: - fiecare componentă are valori în mulțimea $\{1,2,\dots,n\}$
-soluția are c componente

Condiția de validare: - fiecare componentă apare o singură dată

Numărul aranjamentelor: $A_n^c = \frac{n!}{(n-c)!}$

Varianta iterativă

```
1. #include<iostream>
2. using namespace std;
3. int x[30];
4. int n,c,nr;
5.
6. void tipar()
7. {
8.     int i;
9.     for(i=1;i<=c;i++)
10.         cout<<x[i]<<' ';
11.     cout<<'\n';
12.     nr++;
13. }
14.
15. int valid(int k)
16. {
17.     int i;
18.     for(i=1;i<k;i++)
19.         if(x[i]==x[k])return 0;
20.     return 1;
21. }
22.
23. void back()
24. {
25.     int i,k=1;
26.     x[1]=0;
27.     while(k>0)//stiva nu este vida
28.     {
29.         while(x[k]<n)
30.         {
31.             x[k]=x[k]+1;
32.             if(valid(k))
33.                 if(k==c) tipar();
34.                 else{k=k+1;x[k]=0;}
35.         }
36.         k=k-1;
37.     }
38. }
39.
40. int main()
41. {
42.     cin>>n;
43.     cin>>c;
44.     back();
45.     cout<<nr;
46. }
```

Varianta recursivă

```
1. #include<iostream>
2. using namespace std;
3. int x[30];
4. int n,c,nr;
5.
6. void tipar()
7. {
8.     int i;
9.     for(i=1;i<=c;i++)
10.         cout<<x[i]<<' ';
11.     cout<<'\n';
12.     nr++;
13. }
14.
15. int valid(int k)
16. {
17.     int i;
18.     for(i=1;i<k;i++)
19.         if(x[i]==x[k])return 0;
20.     return 1;
21. }
22.
23. void back(int k)
24. {
25.     int i;
26.     if(k==c+1)tipar();
27.     else
28.         for(i=1;i<=n;i++)
29.         {
30.             x[k]=i;
31.             if(valid(k)) back(k+1);
32.         }
33. }
34.
35. int main()
36. {
37.     cin>>n;
38.     cin>>c;
39.     back(1);
40.     cout<<nr;
41. }
```