1922–ben kiadott Izotópok című könyve már figyelmesztetett az atomenergia jövőbeni alkalmazásának hasznára és veszélyeire.

**50 éve** halt meg **Hans GEIGER** (Neustadt, 1882. 9. 30. - Potsdam, 1945. 9. 24.) : német fizikus. 1912 - ben Nuttallal együtt felfedezték a Geiger - Nuttall - törvényt. 1928 - ban egyik tanítványával, W. Müllerrel elkészítette a csúcsszámlálónál is érzékenyebb Geiger - Müller - féle számlálócsövet.

**25 éve** halt meg **Max BORN** (Breslau, 1882. 12. 11. - Göttingen, 1970. 1. 5.) : német elméleti fizikus. 1954–ben Nobel–díjat kapott " alapvető kvantummechanikai munkásságáért ". Fő kutatási területe a kvantummechanika, a kristályrácsok dinamikája, a kristályok termodinamikája, a folyadékok és gázok kinetikus elmélete, a relativitáselmélet és az atomfizika volt.

**25 éve** halt meg **Chandrasekhara Venkata RAMAN** (Tiruchirapalli, India, 1888. 11. 7. - Bangalore, 1970. 11. 24.) : indiai fizikus. 1930–ban az ázsiai fizikusok között elsőként kapott Nobel–díjat „a fény szóródásával kapcsolatos munkásságáért és a róla elnevezett hatás felfedezéséért".

**Cseh Gyopárka**

## *Kísérlet, labor, műhely*

# Permutációk, variációk, kombinációk előállítása – II. rész

### Variációk előállítása

Most térjünk át a variációkra: n elem m-ed osztályú variációja megkapható n elem (m-1)-ed osztályú variációjából, ha annak (mondjuk, hogy) az első helyére beillesztjük az (m-1)-esek között még nem variált elemet. Ezt fejezi ki a

$$V_m^n = ( n . m + 1 ) V_{m-1}^n$$

képlet is, vagyis minden egyes (m-1)-ed osztályú varriációból (n-m+1) új állítható elő m-ed osztályúvá.

Itt a permutációhoz képest a felfejlesztés bonyolultabb, mert n-szer kell a főprogramból is meghívni a *varia* rekurzív eljárást, egyszerűsödik ellenben az új elemek elhelyezése a régebbi generációhoz, mert csak az első helyre tesszük a még nem variált elemet. Ezt elegánsan úgy oldjuk

meg, hogy kiszitáljuk a már variált elemeket és a megmaradt elemeket helyezzük az első helyre, majd átadjuk az eggyel nagyobb osztály generálását a rekurzív eljárásnak. A H halmaz tölti be a szita szerepét, benne maradnak a még nem variált elemek. A technika egyébként ugyanaz mint a permutációnál. Itt is vigyázni kell arra, hogy a kiírást csak, a kívánt osztály elérésekor végezzük el. Ellenörzésként külön kiszámítottuk a variációk számát és a jj változóval követni tudjuk, hogy eljárásunk helyesen működik, előállítván az összes variációt.

```
program vvv;
uses crt;
const ml = 3;
   n = 5;
type t = array (1..ml) of integer;
   halmaz = set of 1..n;
var p : t;
   i : integer;
   jj : integer;
funkcion vv (n,m : integer) : integer;
var t, j : integer;
begin
t := 1;
 for j := 1 to m do t := t* (n - j + 1) ;
VV := t ;
end;
procedure varia (n,m : integer var p : t) ;
var elem, k, l, i : integer;
   v : t ;
   H : halmaz;
begin
 if m = ml + i then
 else
   begin
    H := [ 1..n] ;
    for k := 1 to m-i do
        H := H - [ p[ k]] ;
    for elem := 1 to n do
        if elem in H then
        begin
         v[ i] := elem;
         for l := 2 to m do
             v[ l] := p[ l - 1] ;
         varia (n,m + 1,v);
         if m = ml then
         begin
             for l := i to m do write (v [ l] : 2,' , ');
             writeln (' az' , jj,' ,' )  jj := jj + 1;
             readln;
         end;
        end;
 end;
begin
 clrscr;
 writeln
(' ' , n : 1,' elem -' , ml : 1,' osztályú variációja' , VV (n,mi) : 4);
 jj := i;
 for i := i to n do
```

```
    begin
      p[1] := i;
      varia (n,2,p);
    end;
end.
```

### Kombinációk előállítása

A variációtól már csak egy lépés választ el a kombinációk előállításáig. Az iskolai matematikában azt tanultuk, hogy a variációkból kihagyván a permutációkat, megkapjuk a kombinációkat. A

$$C_m^n = \frac{V_m^n}{P_n}$$

képlet is ezt fejezi ki. Ezért első ötletünk az lehetne, hogy probáljuk meg kiszitálni a permutációkat a variációkból. De ez technikás és hosszas megoldást követel. Ha rájövünk, hogy a kombinációk rendezett variációk, akkor a problémát meg is oldottuk. Ugyanis H-ból még kiszitálva azokat az elemeket, amelyek kisebbek a már előállított, egy ranggal kisebb variációk elemeinél és ezzel dolgozva tovább megkapjuk az összes variációk származtatását a variációk származtatásának algoritmusával.

A *kombio* eljárásban

```
for k := 1 to m - 1 do
   H := H - [p[k]];
 for i:=1 to n do
  if i in H then
   begin
     for k:= 1 to m - 1 do
          if i <p [k] then H := H - [i];
   end;
```

részt kicserélhetjük azzal, hogy:

```
for i:=1 to n do
 if i in H then
   begin
     for k:=1 to m-1 do
          if i <= p[k] then H:= H-[i];
   end;
```

de a jobb megértésért meghagytuk a "variációk" változat kibővítését.

```
program kombi;
uses crt;
const m1=3;
   n=5;
type t=array[1..m1] of integer;
   halmaz = set of 1..n;
var p:t;
   i: integer;
   jj: integer;
function CC (n,m:integer): integer;
var t, u, j : integer;
```

```
begin
t := 1;
u := 1;
 for j := 1 to m do t:= t * (n-j+1);
 for j := 1 to m do u:= u:= u * j;
CC:= t div u;
end;
procedure kombio (n,m: integer var p:t);
var elem, k, k, i: integer;
   v: t;
   H : halmaz;
 begin
  if m = m1 + 1 then
  else
   begin
    H : = [ 1..n] ;
    for k:= 1 to m - 1 do
        H:= H-[ p[ k]] ;
    for i:= 1 to n do
     if i in H then
        begin
         for k:= 1 to m-1 do
             if i <p[ k] then H:= H - [ i] ;
         end;
    for elem:= 1 to n do
     if elem in H then
     begin
        v [ 1] := elem;
        for 1:= 2 to m do
             v [ 1] := p [ 1 - 1] ;
        kombio (n, m + 1, v) ;
        if m = m1 then
        begin
             for 1:= i to m do write (v[ 1] : 2,' ,' ) ;
                      writeln (' az' , jj, ' ,' ) ;
             jj := jj + 1 readln;
        end;
     end;
   end;
end;
BEGIN
   clrscr;
   writeln
(' ' , n:1,' elem --' m1 : 1,' osztályú kombinációja CC (n,mi) : 2) ;
   jj : = 1;
   for i:= 1 to n do
     begin
        p [ 1] := i;
        kombio (n,2, p) ;
     end;
END.
```

*Irodalom*

H. Georgescu, O.Bâscă: Programe în limbajul FORTRAN, Ed. Albatros, 1975.

**Oláh Gál Róbert**