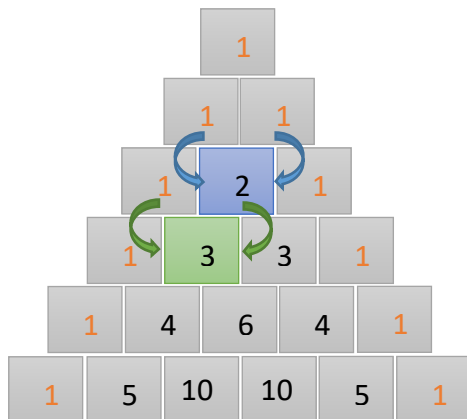


Rezolvări și explicații – set 4

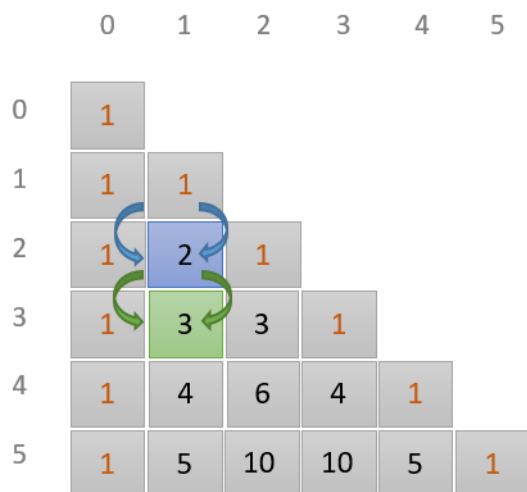
1. Se citește un număr natural n . Să se genereze coeficienții dezvoltării binomului $(x+1)^n$ (Triunghiul lui Pascal).

Rezolvare:

Triunghiul lui Pascal este un tablou triunghiular cu numere naturale, tablou în care fiecare element aflat pe laturile triunghiului are valoarea 1, iar restul elementelor sunt egale cu suma celor două elemente vecine, situate pe linia de deasupra.



În implementare, rearanjăm elementele tabloului astfel:



Observație: Triunghiul lui Pascal din figura alăturată se va afișa pentru $n=6$ (6 linii în tabloul triunghiular)

În soluție, vom folosi un tablou bidimensional cu doi indici i și j , cu ajutorul cărora vom parcurge matricea punând 1 pe prima coloana și 1 pe diagonala principală (construim laturile triunghiului lui Pascal), iar în rest, vom pune 0 (inițializare):

	$j \rightarrow$	0	1	2	3	4	5
$i \downarrow$ 0		1	0	0	0	0	0
1		1	1	0	0	0	0
2		1	0	1	0	0	0
3		1	0	0	1	0	0
4		1	0	0	0	1	0
5		1	0	0	0	0	1

După care, calculăm linia curentă i folosind linia precedentă $i-1$ (instrucțiunile 14-16).

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  int main ()
4  {
5      int n,i,j,a[20][20];
6      printf("n=");
7      scanf("%d",&n); //numar de linii in triunghiul lui Pascal
8      //vom pune 1 in prima coloana (coloana j=0) si 0 in rest
9      for(i=0;i<=n;i++)
10         for(j=0;j<=n;j++)
11             if(j==0)
12                 a[i][j]=1; //pune 1 pe prima coloana
13             else a[i][j]=0; //0 in rest
14      for(i=1;i<=n;i++)
15         for(j=1;j<=i;j++)
16             a[i][j]=a[i-1][j]+a[i-1][j-1];
17      printf("Triunghiul lui Pascal pentru n=%d este\n",n);
18      for(i=0;i<n;i++)
19      {
20         for(j=0;j<=i;j++)
21             printf("%5d",a[i][j]);
22         putchar('\n');
23     }
24     return 0;
25 }
```

2. Se citesc 2 numere naturale n și k , $k \leq n$ și un șir a_0, a_1, \dots, a_n reprezentând coeficienții unui polinom de gradul n . Să se afișeze toate derivatele până la ordinul k ale acestui polinom.

Rezolvare:

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  int main()
4  {
5      // n ordinul polinomului
6      int n,k,i,a[10],nrder;
7      printf("Introduceti gradul polinomului: ");
8      scanf("%d",&n);
9      do{
10         printf("Introduceti ordinul derivatei (<%d): ",n);
11         scanf("%d",&k);
12     }while (k>=n); // citim k cu validare
13     // fiindca k trebuie sa fie strict mai mic ca n
14     printf("Introduceti coeficientii polinomului\n");
15     for (i=0;i<=n;i++)//n+1 coeficienti
16     {
17         //polinomul a[0]*x^n+a[1]*x^(n-1)+...+a[i]*x^(n-i)+...+a[n-1]*x+a[n]
18         printf("a[%d]=",i);
19         scanf("%d",&a[i]);
20     }
21     for(nrder=1;nrder<=k;nrder++) //contorul nrder tine evidenta derivatelor
22     // de ordin mai mic sau egal cu k
23     {
24         for(i=0;i<n;i++)
25             a[i]=a[i]*(n-i);
26         n--; // dupa ce construim o derivata, decrementam n cu o unitate
27         printf("\n\nDerivata de ordinul %d a polinomului este:\n",nrder);
28         for (i=0;i<=n;i++)
29         {
30             if(i<n)
31                 printf("%d*X^%d+",a[i],n-i);
32             else
33                 printf("%d",a[i]);
34         }
35     }
36     return 0;
37 }

```