# Комбинаторни алгоритми

ИТ Кариера

Учителски екип
Обучение за ИТ кариера
https://it-kariera.mon.bg/e-learning

# Съдържание

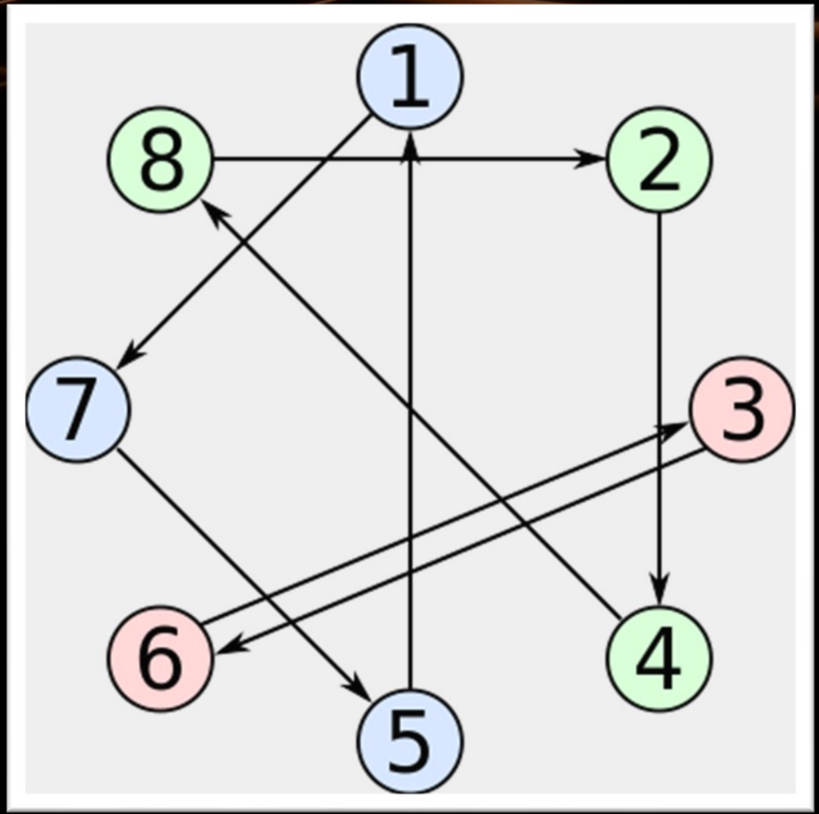
- Генериране на вариации, комбинации, пермутации
- Упражнения: генериране на комбинации и вариации
- Упражнения: генериране на пермутации и други комбинаторни обекти
- Упражнения: комбинаторни задачи

# Множества

- Съвкупност от обекти, обединени по някакъв общ признак.
- Обектите, от които се състои множеството, се наричат елементи.
- Символният запис a∈A означава, че елементът a принадлежи на множеството A.

# Множества

- Множеството A се нарича крайно, ако се състои от краен брой елементи.
- Означаваме $A=\{a_1, a_2, \ldots, a_n\}$ и пишем $|A|=n$.
- Множеството Ø, което не съдържа нито един елемент, се нарича празно множество.




# Пермутации

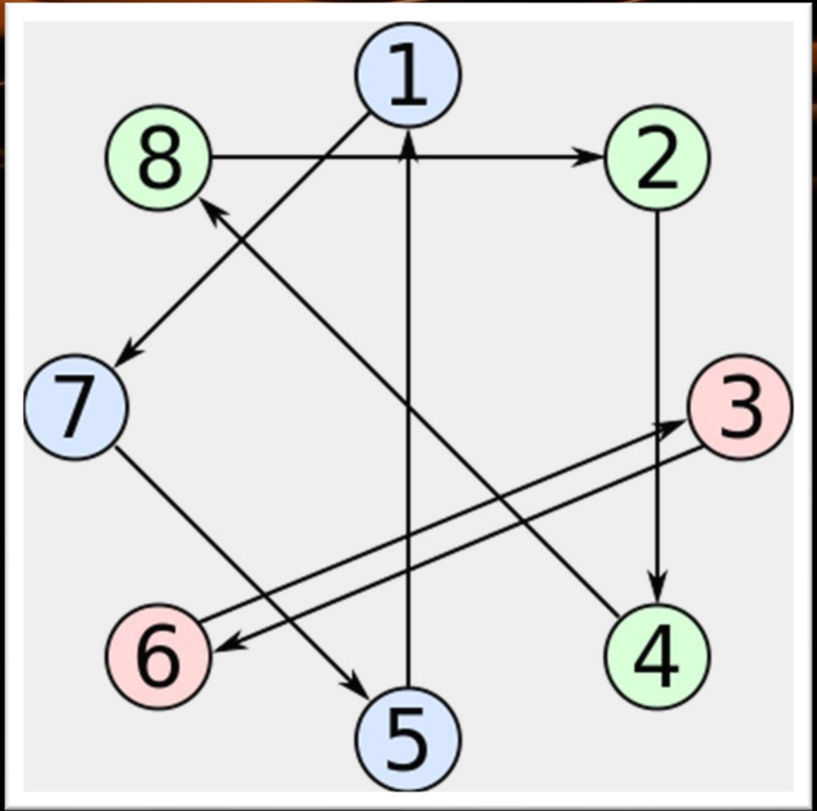
# Пермутации без повторение

Определение: Нека A е множество и |A|=n. Всяко подреждане на всичките n елемента на A (или всички различни подреждания на първите n естествени числа) се нарича пермутация без повторение от n-ти ред. Две пермутации се различават една от друга по реда на елементите, участващи в тях.

Теорема: Броят на всички различни пермутации от n-ти ред е:

$$P_n = 1.2.3\ldots n = n!$$

По определение се приема, че 0!=1.

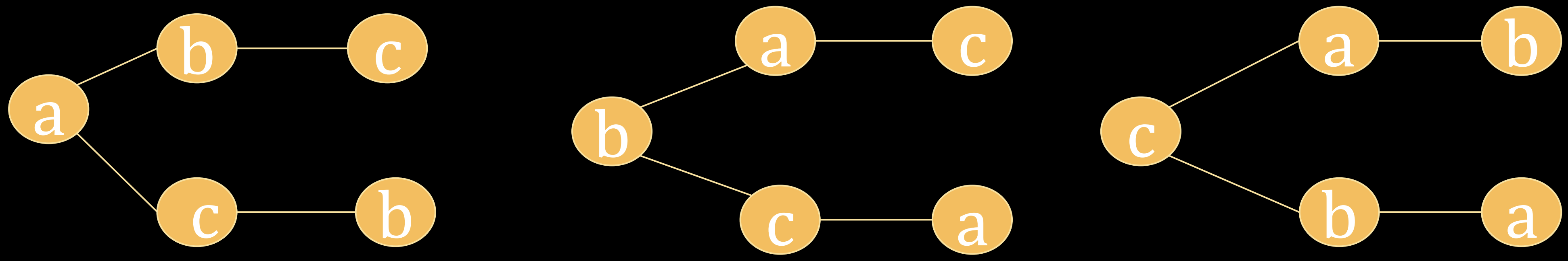



# Пермутации

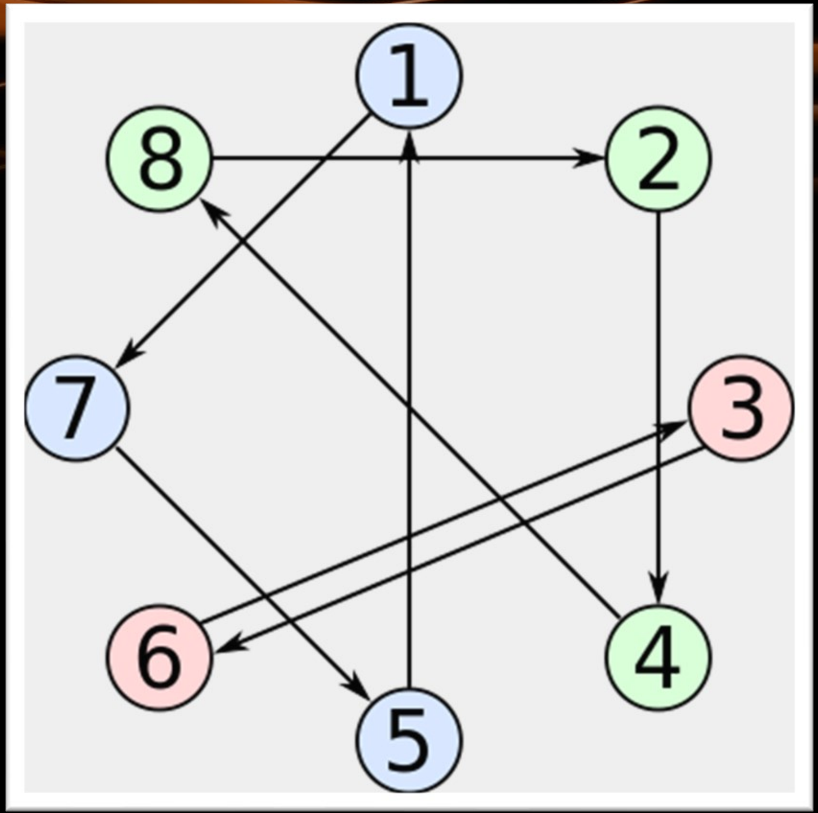
Задача

# Пермутации без повторение

По колко различни начина могат да седнат трима приятели на един ред в киносалон?




|A|=|{abc, acb, bac, bca, cab, cba}|=3.2.1=6.




# Пермутации

Задача

# Пермутации без повторение

Колко различни петцифрени числа могат да се запишат с цифрите 0, 1, 2, 3, и 4, ако всички цифри участват?

0 1 2 3 4 NO!

1 0 2 3 4 4!

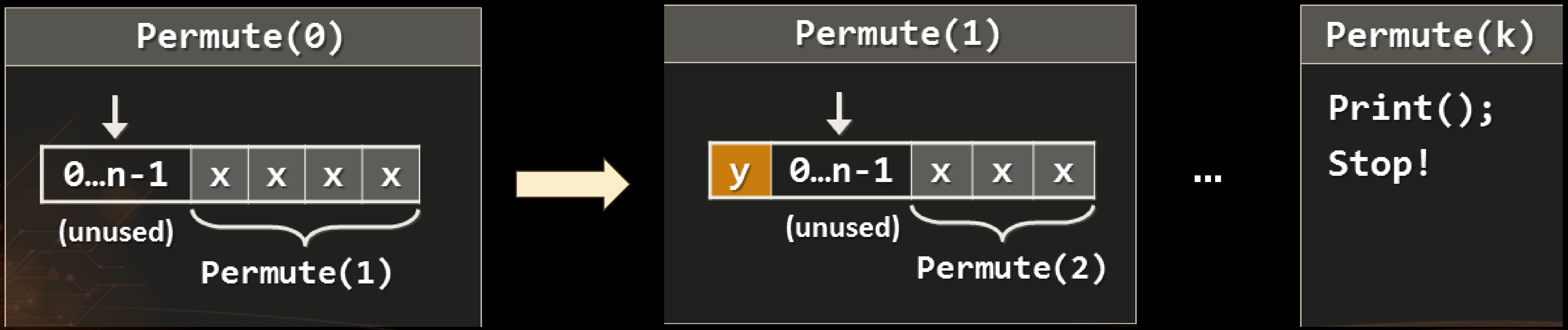
# Пермутации без повторение

| | | | | | |
|---|---|---|---|---|---|
| 2 | 1 | 0 | 3 | 4 | 4! |
| 3 | 1 | 2 | 0 | 4 | 4! |
| 4 | 1 | 2 | 3 | 0 | 4! |

общо:
96

# Пермутации: алгоритъм

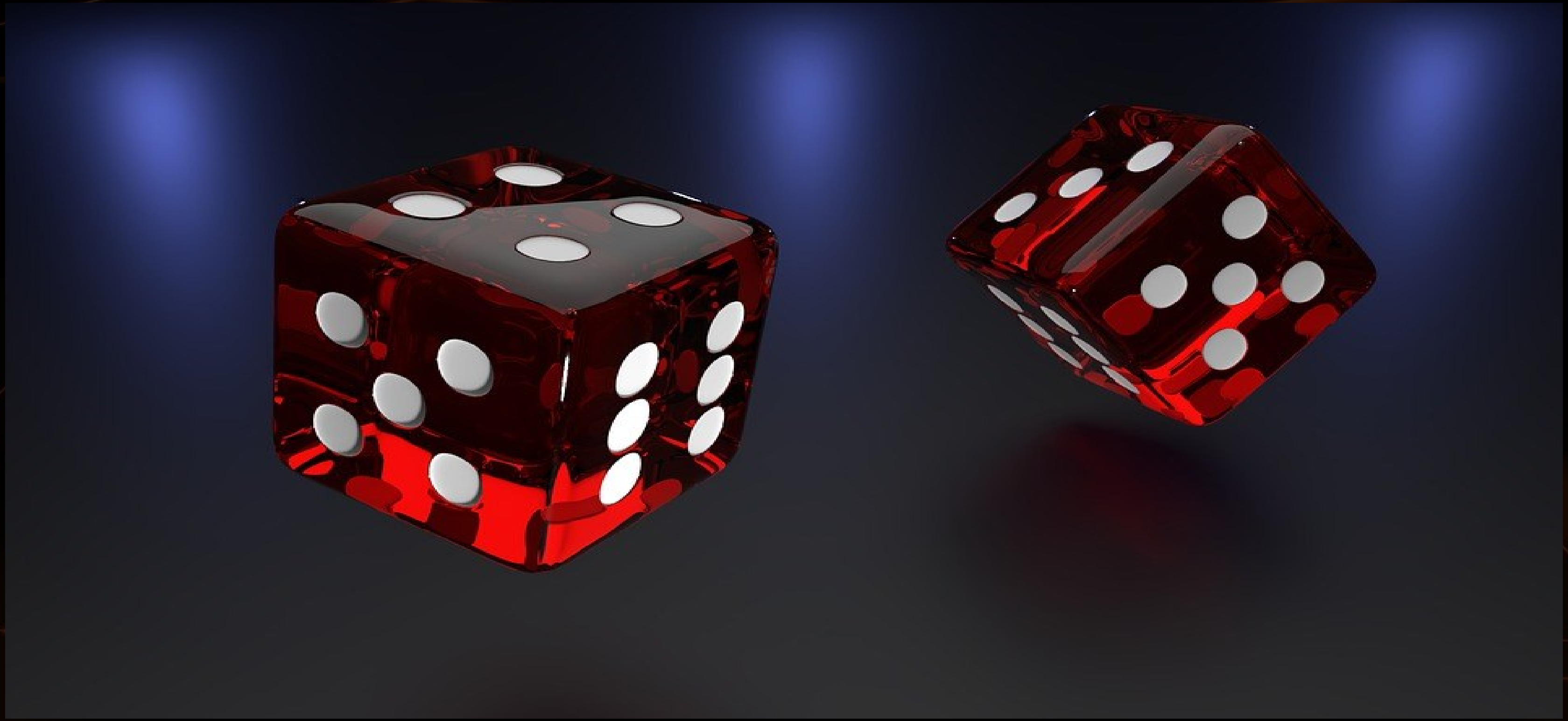
- Използва се функция с един параметър Permute(index)
- Индекса i ще пази неизползваните елементи i=0…n-1
- Маркират се всички използвани елементи
- Извиква се рекурсивно функцията comb(index + 1), за да се генерира останалата част от масива



# Генериране на пермутации

```
public static void Gen(int index)
{
  if (index >= elements.Length)
    Console.WriteLine(string.Join(" ", perm));
  else
    for (int i = 0; i < elements.Length; i++)
      if (!used[i])
      {
        used[i] = true;
        perm[index] = elements[i];
        Gen(index + 1);
        used[i] = false;
      }
}
```

```
elements = new int[n];
used = new bool[n];
perm = new int[n];
Permute(0);
```



# Комбинации

# Комбинации без повторения

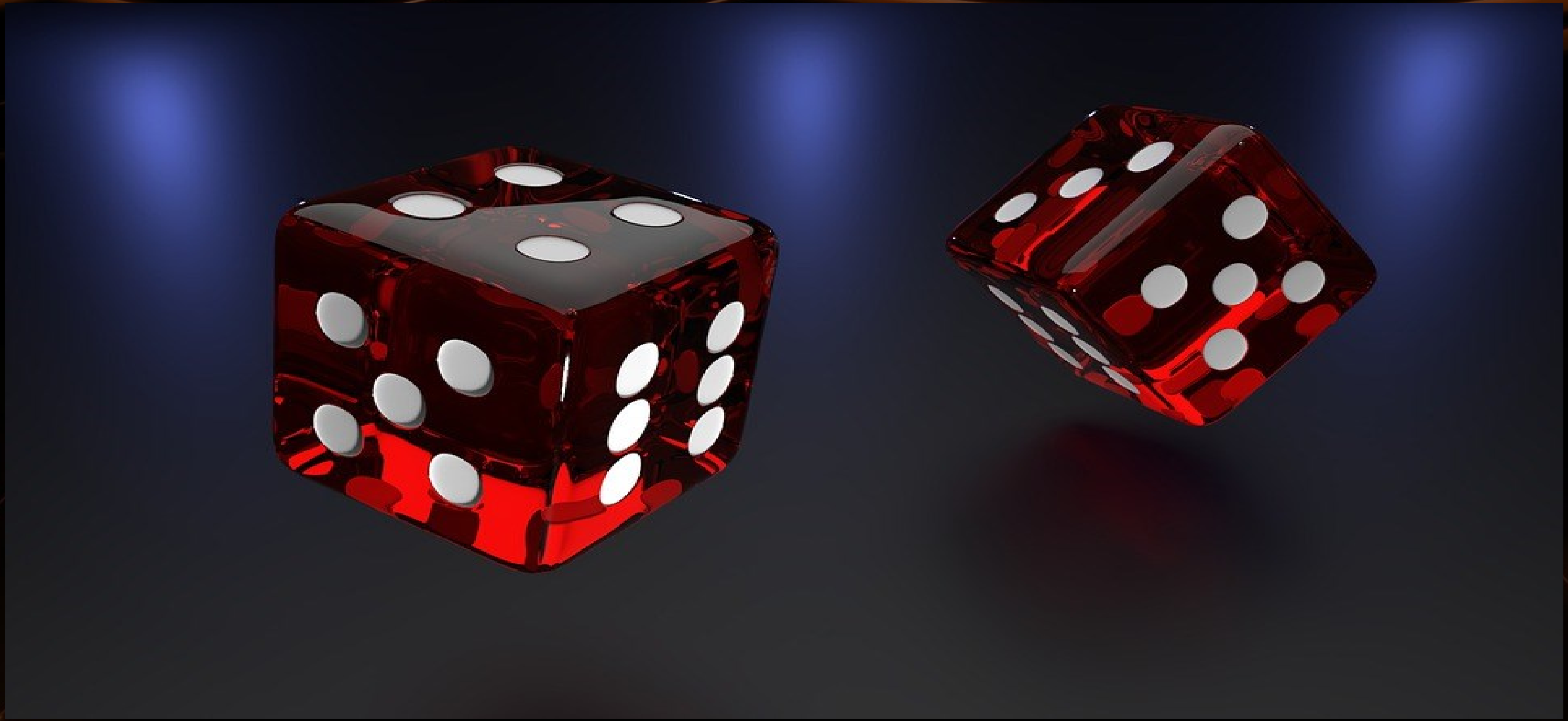
Определение: Всички различни не наредени извадки без повторение на n елемента от k-ти клас се наричат комбинации без повторение на n елемента от k-ти клас. Две комбинации без повторение се различават една от друга по елементите, участващи в тях.

Теорема: Броят на всички различни комбинации на n елемента от k-ти клас е:

$$C_n^k = \binom{n}{k} = \frac{n!}{k!(n-k)!} = \frac{n(n-1)\dots(n-k+1)}{k!}$$

Някои по-често използвани свойства на биномните коефициенти са:

$$C_n^0 = C_n^n = 1 \qquad C_n^1 = C_n^{n-1} = n$$

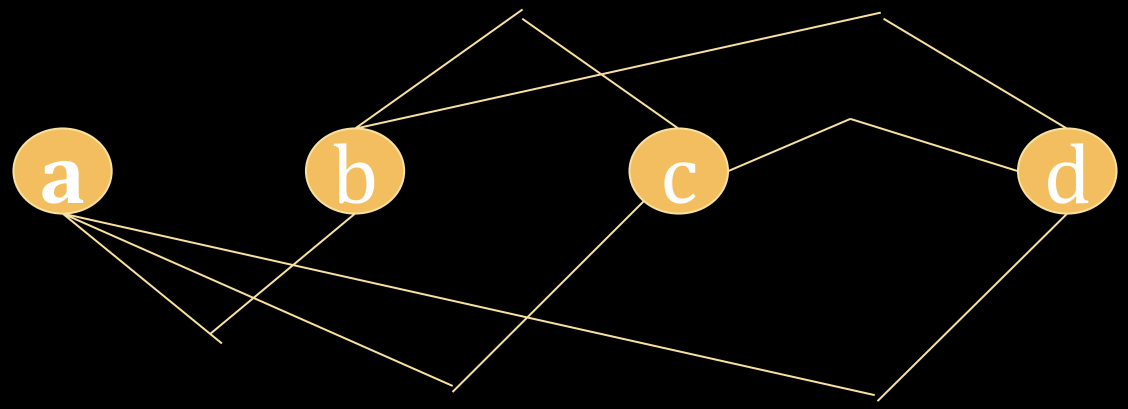


# Комбинации

## Задача

# Комбинации без повторения

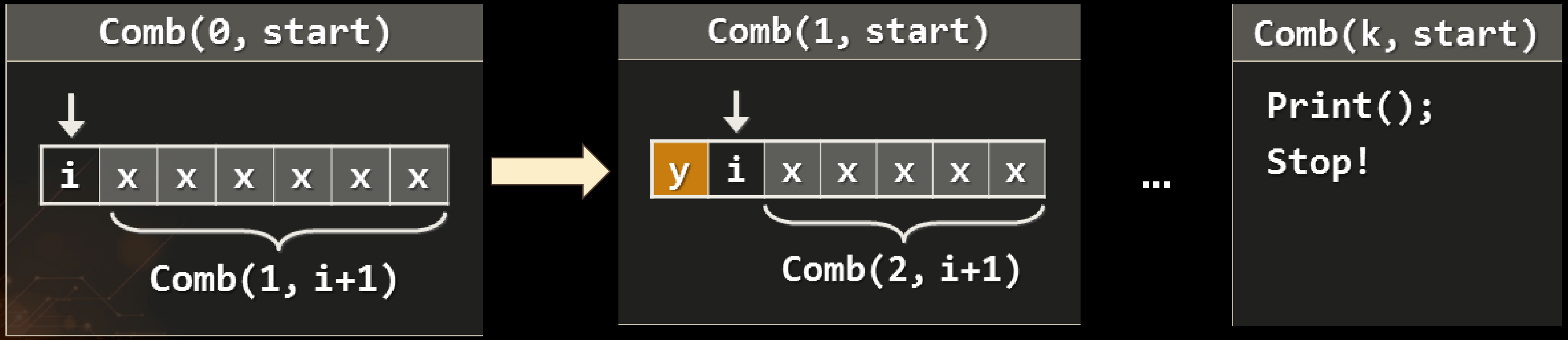
Да се опишат всички комбинации без повторение на четири елемента от втори клас от елементите a, b, c и d и да се определи броят им.

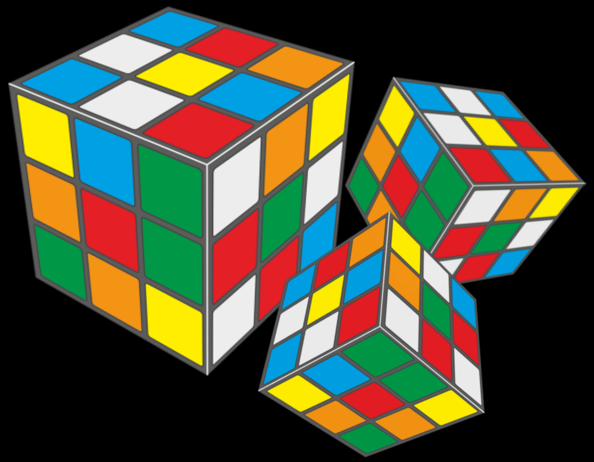



|A|=|{ab, ac, ad, bc, bd, cd}|=6.

# Генериране на комбинации без повторения

- Използва се функция с два параметъра comb(index, start)
- Индекса i има за начална стойност start, и крайна n-1
- Извиква се рекурсивно функцията comb(index + 1, i + 1), за да се генерира останалата част от масива





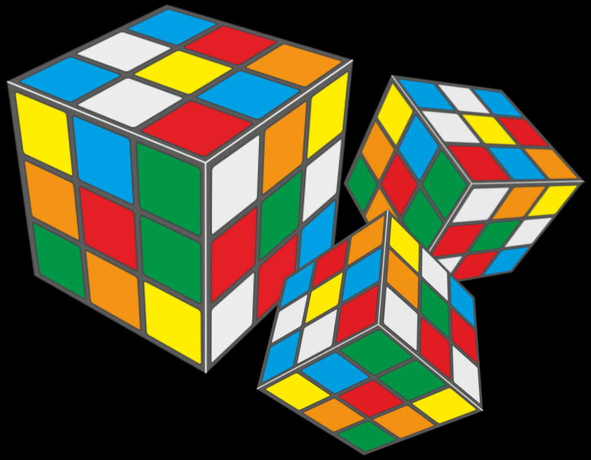
# Вариации

# Вариации без повторения

Определение: Всички различни наредени извадки без повторение на n елемента от k-ти клас наричаме вариации без повторение на n елемента от k -ти клас. Две вариации без повторение се различават една от друга или по реда на участващите в тях елементи или по елементите, участващи в тях.

Теорема: Броят на всички различни вариации без повторение на n елемента от k-ти клас е:

$$V_n^k = n(n-1)\ldots(n-k+1)$$



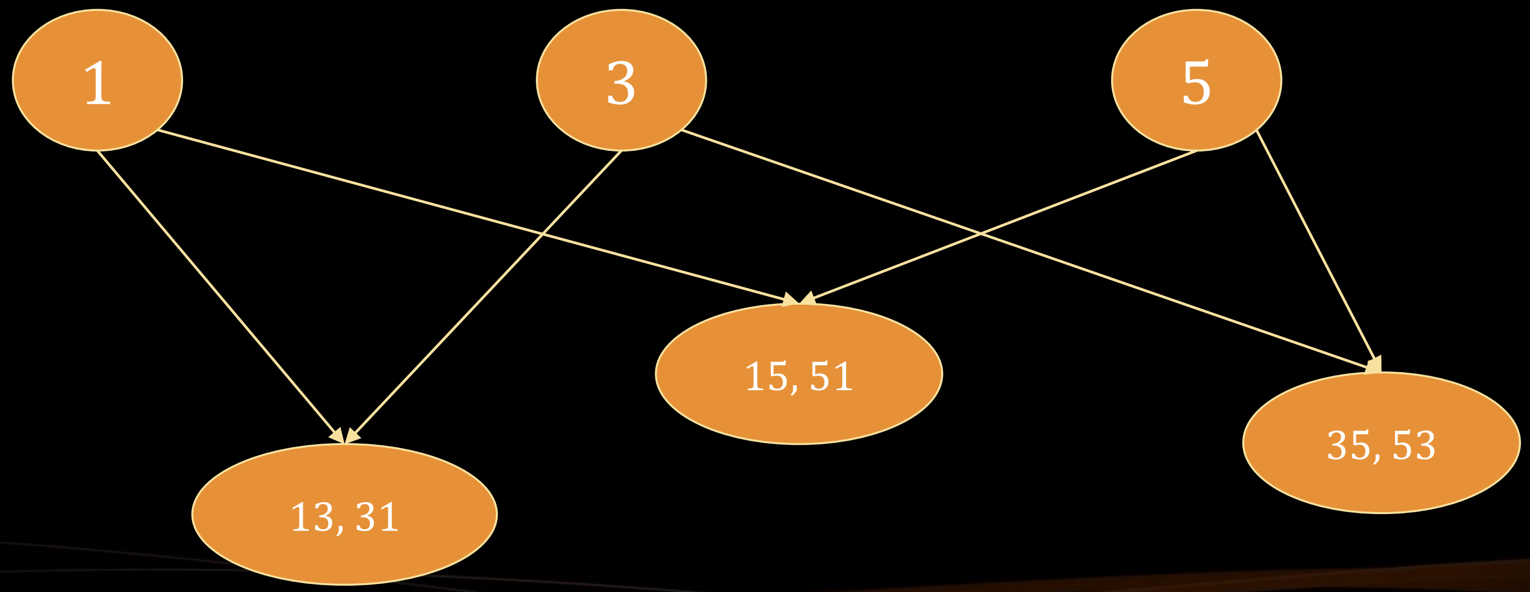
# Вариации

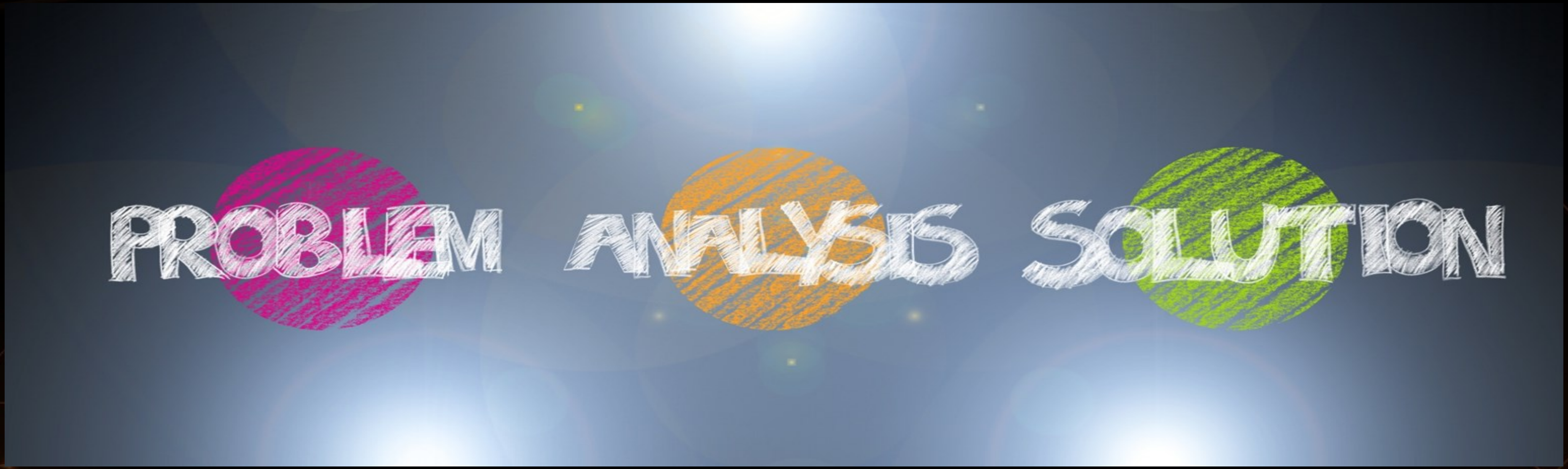
## Задача

# Вариации без повторения

Колко различни двуцифрени числа, съставени от различни цифри могат да бъдат образувани с цифрите 1, 3 и 5.

NB! Редът на подреждането на цифрите в числото има значение.






# Упражнения:
# генериране на комбинации и вариации

# Упражнения: генериране на комбинации и вариации

Задача: Да се напише програма, която въвежда от клавиатурата две естествени числа n - броя на елементите и k - от колко елемента да се състои подмножеството, което ще се генерира. Програмата да извежда всички комбинации от n елемента k-ти клас.

Примерен вход: 4 2

Примерен изход: 1 2

1 3

1 4

2 3

2 4

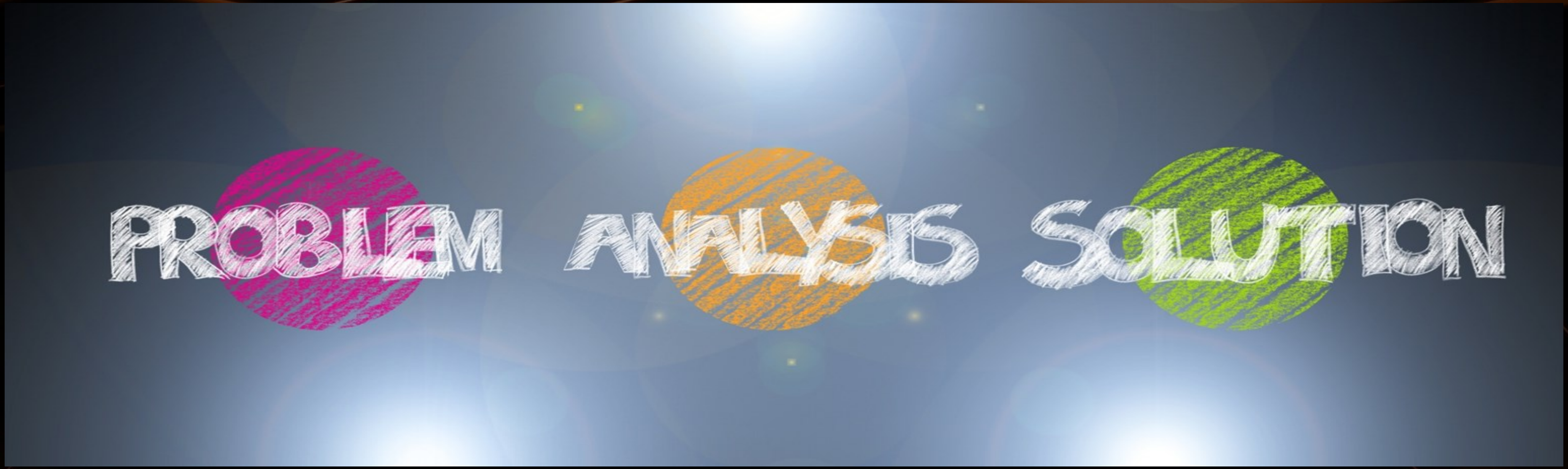
3 4

# Упражнения: генериране на комбинации и вариации

Задача: Да се напише програма, която въвежда от клавиатурата две естествени числа n - броя на елементите и k - от колко елемента да се състои подмножеството, което ще се генерира. Програмата да извежда всички комбинации от n елемента k-ти клас, като редът на елементите е от значение.

Примерен вход: 4 2

Примерен изход: 1 2 1 3 1 4 2 1 2 3 2 4 3 1 3 2 3 4 4 1 4 2 4 3




# Упражнения:
# генериране на пермутации и други комбинаторни обекти

# Упражнения: генериране на пермутации и други комбинаторни обекти

Задача: Да се напише програма, която въвежда от клавиатурата едно цяло число n - броя на елементите от дадено множество. Програмата да извежда всички възможни подреждания на тези елементи. Всеки елемент участва веднъж и мястото му е съществено.

Примерен вход: 3

Примерен изход: 1 2 3

1 3 2

2 1 3

2 3 1

3 1 2

3 2 1

# Упражнения: генериране на пермутации и други комбинаторни обекти

Задача: Напишете програма, която намира биномните коефициенти на израза

$$(x+y)^n$$

където n се въвежда от клавиатурата .

Примерен вход: 2

Примерен изход: 1 2 1

# Упражнения: генериране на пермутации и други комбинаторни обекти

При повдигане на степен n на израза (x+y) коефициентите пред съответните степени на x и y са всъщност биномните коефициенти.

Например:

$$(x+y)^2 = \binom{2}{0}.x^2 + \binom{2}{1}.x.y + \binom{2}{2}.y^2 = 1.x^2 + 2.x.y + 1.y^2$$

# Биномни коефициенти

За комбинация на n елемента от k- ти клас използваме означението $\binom{n}{k}$

В литературата е прието да се означава $C_n^k$

Теорема: Броят на всички различни комбинации на n елемента от k-ти клас е:

$$C_n^k = \binom{n}{k} = \frac{n!}{k!(n-k)!} = \frac{n(n-1)\dots(n-k+1)}{k!}$$

# Биномни коефициенти

- Тази формула е еквивалентна на:

$$\binom{n}{k} = \binom{n}{n-k}$$

- Тези числа са известни като биномни коефициенти, тъй като те участват в Нютоновия бином (математическа теорема за разлагане на двучлен, повдигнат на степен).

$$\left(x+y\right)^n = \sum_{k=0}^{n} \binom{n}{k} x^k y^{n-k}$$

# Биномни коефициенти

- В частен случай, когато x=y=1 се получава:

$$\sum_{k=0}^{n} \binom{n}{k} = 2^n$$

- Някой най-често използвани свойства на биномните коефициенти са:

$$C_n^0 = C_n^n = 1 \qquad C_n^1 = C_n^{n-1} = n$$

# Биномни коефициенти

- Рекурсивната дефиниция за броя на всички комбинации от n елемента k-ти клас е:

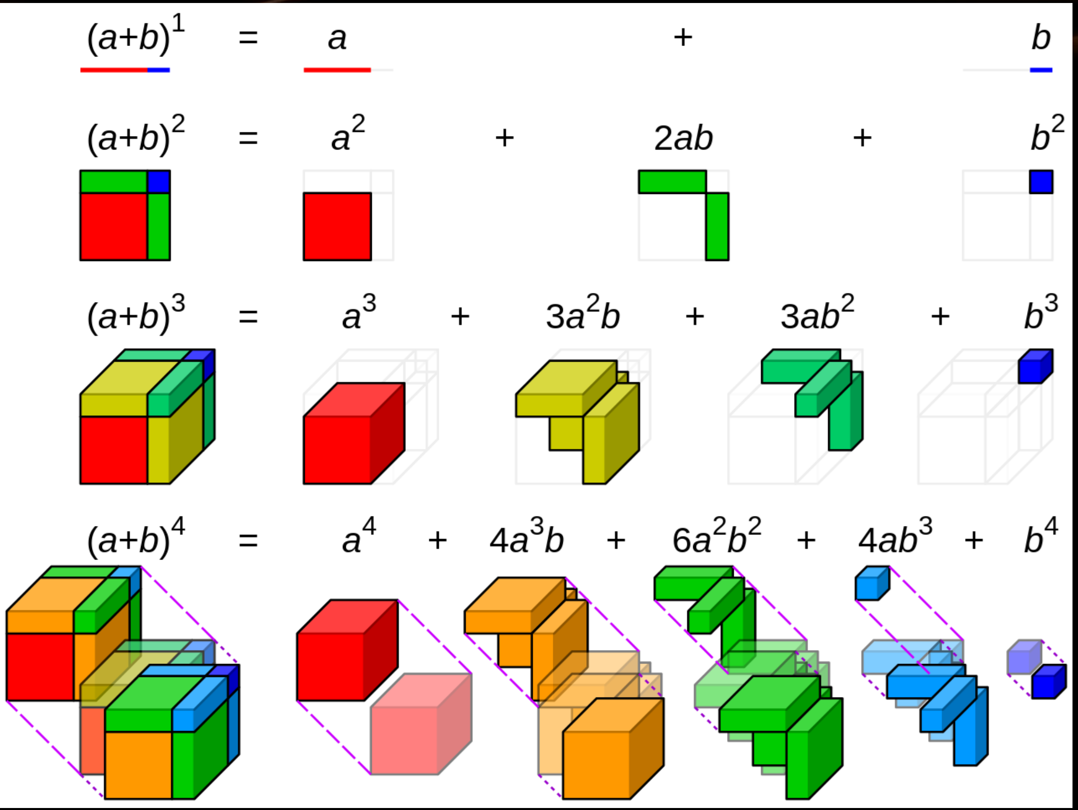
$$\binom{n}{k} = \binom{n-1}{k} + \binom{n-1}{k-1}$$

- Частни случаи:

$$\binom{n}{0} = 1 \qquad \binom{n}{n} = 1$$

# Биномни коефициенти – решение

```
// Рекурсивна функция
int binom(int n, int k)
{
    if (n==k || k==0) return 1;
    return binom(n-1, k) + binom(n-1, k-1);
}
```




# Триъгълник на Паскал

# Триъгълник на Паскал

- Аритметичен триъгълник, съдържащ биномните коефициенти.
- Позволява да разположите биномните коефициенти, като всяко число е равно на сумата от двете числа над него.

n=0 → 1

n=1 → 1 1

n=2 → 1 2 1

# Триъгълник на Паскал

| | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | 1 | | | | | | | |
| | | | | | | 1 | | 1 | | | | | | |
| | | | | | 1 | | 2 | | 1 | | | | | |
| | | | | 1 | | 3 | | 3 | | 1 | | | | |
| | | | 1 | | 4 | | 6 | | 4 | | 1 | | | |
| | | 1 | | 5 | | 10 | | 10 | | 5 | | 1 | | |
| | 1 | | 6 | | 15 | | 20 | | 15 | | 6 | | 1 | |
| 1 | | 7 | | 21 | | 35 | | 35 | | 21 | | 7 | | 1 |

# Триъгълник на Паскал

- Ако сте наблюдателни в триъгълника на Паскал ще забележите:
  - Естествените числа 1, 2, 3, 4, ...
  - Триъгълните числа 1, 3, 6, 10, ...
  - Петоъгълните числа 1, 5, 15, 35, 70, ...
  - Шестоъгълни числа 1, 6, 15, 28, ...

# Триъгълник на Паскал

- Ако сте наблюдателни в триъгълника на Паскал ще забележите:
  - Пирамидалните числа 1, 4, 10, 20, ...
  - Числа на Фибоначи 1, 1, 2, 3, 5, 8, ...
  - Числа на Каталан 1, 2, 5, 14, 42, ...

# Триъгълник на Паскал

Задача: Напишете програма, която извежда триъгълника на Паскал. От клавиатурата се въвеждат две цели числа N- броя на елементите и K - подмножествата. На изхода изведете триъгълника на Паскал, подравнен в ляво.

| $k \backslash n$ | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | | | | | | | | | |
| 1 | 1 | 1 | | | | | | | | |
| 2 | 1 | 2 | 1 | | | | | | | |
| 3 | 1 | 3 | 3 | 1 | | | | | | |
| 4 | 1 | 4 | 6 | 4 | 1 | | | | | |
| 5 | 1 | 5 | 10 | 10 | 5 | 1 | | | | |
| 6 | 1 | 6 | 15 | 20 | 15 | 6 | 1 | | | |
| 7 | 1 | 7 | 21 | 35 | 35 | 21 | 7 | 1 | | |
| 8 | 1 | 8 | 28 | 56 | 70 | 56 | 28 | 8 | 1 | |
| 9 | 1 | 9 | 36 | 84 | 126 | 126 | | | | |
| 10 | 1 | 10 | 45 | 120 | 210 | 252 | | | | |
| 11 | 1 | 11 | 55 | 165 | 330 | 462 | | | | |

Binomial coefficients $\binom{n}{k}$ for $0 \le k \le 11$.

# Обобщение

- Пермутации - начини за подреждане на N елемента
- Вариации - начини за подреждане на K от N елемента
- Комбинации - начини за избор на K от N елемента
- Триъгълник на Паскал
- Биномни коефициенти

# Министерство на образованието и науката (МОН)

- Настоящият курс (презентации, примери, задачи, упражнения и др.) е разработен за нуждите на Национална програма "**Обучение за ИТ кариера**" на МОН за подготовка по професия "Приложен програмист"

Министерство на образованието и науката

Национална програма „Обучение за ИТ кариера"

- Курсът е базиран на учебно съдържание и методика, предоставени от **фондация "Софтуерен университет"** и се разпространява под свободен лиценз **CC-BY-NC-SA**

SoftUni Foundation