# Упражнения: Комбинаторни алгоритми

Тествайте решението в Judge: <https://judge.softuni.org/Contests/3641/Combinatorial-Algorithms-Exercises>

Използвайте дадения скелет:

A picture containing table

Description automatically generated

## Пермутация без повторения

Даден ви е **набор** от елементи. Намерете всички **пермутации без повторение**.

### Примери

|  |  |
| --- | --- |
| **Вход** | **Изход** |
| A B C | A B C  A C B  B A C  B C A  C B A  C A B |

### Насоки

За да напишем алгоритъм, трябва да направим дъно на рекурсията. Дъното ще проверява дали **индексът** е **по-голям или равен** на **дължината на пермутацията**:

Graphical user interface, text, application

Description automatically generated

Алгоритъмът започва от 0 до дължината на пермутацията. Ако не сме използвали сегашната пермутация, трябва да:

* Маркираме **сегашният индекс** като използван, за да сме сигурни, че **не се повтаря**
* Задаваме текущата **пермутация** като тази от входа, за да актуализираме нашата **последователност**
* Извикваме GeneratePermutations() с **index+1**, за да отидем на следващата възможна пермутация
* Маркираме сегашния индекс като неизползван, за да може да го използваме отново за следваща пермутация

Graphical user interface, text, application

Description automatically generated

## Пермутация с повторения

Даден ви е **набори от елементи** с възможност за повтаряне. Намерете **всички пермутации**.

### Примери

|  |  |
| --- | --- |
| **Вход** | **Изход** |
| A B B | A B B  B A B  B B A |

## Вариация без повторения

Даден ви е **набор** от елементи. Намерете **вариациите** от **k елемента без повторение**.

### Примери

|  |  |
| --- | --- |
| **Вход** | **Изход** |
| A B C  2 | A B  A C  B A  B C  C A  C B |

### Насоки

Каква е разликата от предишния алгоритъм? Къде влиза **K**? **Преди** или **по време** на алгоритъма?

## Вариация с повторения

Даден ви е набор от елементи. Намерете всички **вариации** от **k елемента с повторения**.

### Примери

|  |  |
| --- | --- |
| **Вход** | **Изход** |
| A B C  2 | A A  A B  A C  B A  B B  B C  C A  C B  C C |

## Кино

Напишете програма, която отпечатва всички **възможни** **разпределения** на **група от приятели** в кино салон.

### Вход

* На първия ред ще получите **всички имена на приятелите**, разделени със **запетая** и **интервал** .
* Докато не получите командата **"generate"**, ще получите имената на приятелите и номерата на местата им, където искат да сядат във формата: **"{name} - {place}"**.
  + Тези приятели искат да седнат само на мястото, което са избрали. Те не могат да седят на други места. За повече информация вижте примерите по-долу.

### Изход

Отпечатайте **всички** **възможни** **начини** за **разпределяне на приятелите**, имайки предвид, че някои от тях искат определено място да седят само там. **Редът** на изхода **няма значение**.

### Бележки

* **Имената** на приятелите и **номерата** на местата ще бъдат валидни

### Примери

|  |  |  |
| --- | --- | --- |
| **Вход** | **Изход** | **Обяснения** |
| Peter, Amy, George, Rick  Amy - 1  Rick - 4  generate | Amy Peter George Rick  Amy George Peter Rick | Amy иска да седи на първата седалка и Rick иска да седи на четвъртата седалка, така че сменяме само другите приятели |
| Garry, Liam, Teddy, Anna, Buddy, Simon  Buddy - 3  Liam - 5  Simon - 1  generate | Simon Garry Buddy Teddy Liam Anna  Simon Garry Buddy Anna Liam Teddy  Simon Teddy Buddy Garry Liam Anna  Simon Teddy Buddy Anna Liam Garry  Simon Anna Buddy Garry Liam Teddy  Simon Anna Buddy Teddy Liam Garry |  |

### Насоки

Във вашият **метод** Generate() може да проверите дали индексът е по-голям или равен на имената. Ако е така, просто отпечатайте **резултата** и се **върнете**.

След това извикайте пермутация за **index + 1.**

За **обратното проследяване** от **index + 1** до броя на имената:

* Разменяме **index** с i.
* Извикваме пермутация за **index + 1.**
* Разменяме **index** с i.

## Комбинации без повторение

Даден ви е **набор** от елементи. Генерирайте всички **комбинации** от **k елемента** **без повторения**.

### Примери

|  |  |
| --- | --- |
| **Вход** | **Изход** |
| A B C  2 | A B  A C  B C |

### Решение

Алгоритъмът се различава от пермутацията по следния начин:

For-цикъла започва от startIndex. За всяко обхождане задаваме текущата комбинация да е равна на тази от входа и извикваме generate() с **index + 1** и **currentIndex + 1**

Graphical user interface, text, application

Description automatically generated

## Комбинации без повторение

Даден ви е **набор** от елементи. Генерирайте всички **комбинации** от **k елементи с повторение**.

### Примери

|  |  |
| --- | --- |
| **Вход** | **Изход** |
| A B C  2 | A A  A B  A C  B B  B C  C C |

## N избора K пъти

Дадени са ви числата **n** и **k**. Пресметнете всички възможните **комбинации n-k** (**без повторение**).

### Примери

|  |  |
| --- | --- |
| **Вход** | **Изход** |
| 3  2 | 3 |
| 5  3 | 10 |

## Училищни отбори

Напишете програма, която получава имена на момчета и момичета в клас. Генерирайте всички възможни отбори с 3 момичета и 2 момчета. Отпечатайте всеки отбор разделен със запетая и интервал **", "** (първо момичетата и след това момчетата). За повече информация вижте примерите.

### Вход

* На първия ред ще получите имената на **момичетата**, разделение със запетая и интервал **", "**.
* На следващия ред ще получите имената на **момчетата** разделени със запетая и интервал **", "**.

### Изход

* Отпечатайте всички **възможни отбори** на нов ред с точно **3 момичета** и **2 момчета**, разделени със запетая и интервал. Започнете с момичетата.

### Бележки

* Ще получите поне **3 момичета** и **2 момчета**.

### Примери

|  |  |
| --- | --- |
| **Вход** | **Изход** |
| Linda, Amy, Katty  John, Bill | Linda, Amy, Katty, John, Bill |
| Lisa, Yoana, Marta, Rachel  George, Garry, Bob | Lisa, Yoana, Marta, George, Garry  Lisa, Yoana, Marta, George, Bob  Lisa, Yoana, Marta, Garry, Bob  Lisa, Yoana, Rachel, George, Garry  Lisa, Yoana, Rachel, George, Bob  Lisa, Yoana, Rachel, Garry, Bob  Lisa, Marta, Rachel, George, Garry  Lisa, Marta, Rachel, George, Bob  Lisa, Marta, Rachel, Garry, Bob  Yoana, Marta, Rachel, George, Garry  Yoana, Marta, Rachel, George, Bob  Yoana, Marta, Rachel, Garry, Bob |

## Плодови напитки

Напишете програма, която получава имена на **плодове** и **топинги**. Генерирайте всички възможни начина да създадете напитка с **2 плода** и **1 топинг**.Отпечатайте всички напитки на нов ред.

### Вход

* На първия ред ще получите имената на плодовете, разделени със запетая и интервал.
* На следващия ред ще получите **топингите**, разделени със запетая и интервал.

### Изход

* Отпечатайте всички възможни напитки с **2 плода** и **1 топинг**, разделени със **запетая** и **интервал**.

### Примери

|  |  |
| --- | --- |
| **Вход** | **Изход** |
| Apple Peach Pear  Mint OrangeSlice | Apple, Apple, Mint  Apple, Apple, OrangeSlice  Apple, Peach, Mint  Apple, Peach, OrangeSlice  Apple, Pear, Mint  Apple, Pear, OrangeSlice  Peach, Peach, Mint  Peach, Peach, OrangeSlice  Peach, Pear, Mint  Peach, Pear, OrangeSlice  Pear, Pear, Mint  Pear, Pear, OrangeSlice |
| Apple Peach Cherries Orange  Mint Straw OrangeSlice NoTopping | Apple, Apple, Mint  Apple, Apple, Straw  Apple, Apple, OrangeSlice  Apple, Apple, NoTopping  Apple, Peach, Mint  Apple, Peach, Straw  Apple, Peach, OrangeSlice  Apple, Peach, NoTopping  Apple, Cherries, Mint  Apple, Cherries, Straw  Apple, Cherries, OrangeSlice  Apple, Cherries, NoTopping  Apple, Orange, Mint  Apple, Orange, Straw  Apple, Orange, OrangeSlice  Apple, Orange, NoTopping  Peach, Peach, Mint  Peach, Peach, Straw  Peach, Peach, OrangeSlice  Peach, Peach, NoTopping  Peach, Cherries, Mint  Peach, Cherries, Straw  Peach, Cherries, OrangeSlice  Peach, Cherries, NoTopping  Peach, Orange, Mint  Peach, Orange, Straw  Peach, Orange, OrangeSlice  Peach, Orange, NoTopping  Cherries, Cherries, Mint  Cherries, Cherries, Straw  Cherries, Cherries, OrangeSlice  Cherries, Cherries, NoTopping  Cherries, Orange, Mint  Cherries, Orange, Straw  Cherries, Orange, OrangeSlice  Cherries, Orange, NoTopping  Orange, Orange, Mint  Orange, Orange, Straw  Orange, Orange, OrangeSlice  Orange, Orange, NoTopping |