

Перестановка элементов в массиве

Перестановками называются выборки элементов, отличающиеся только порядком расположения элементов, но не самими элементами.

Пример.



после перестановки:



Число перестановок на множестве из n элементов определяется по формуле: $P_n = n \cdot (n-1) \cdot (n-2) \dots 3 \cdot 2 \cdot 1 = n!$

$n!$ (n -факториал) – обозначение для краткой записи произведения всех натуральных чисел от 1 до n включительно. Например, перестановка пяти книг будет равна: $P_5 = 5! = 1 \cdot 2 \cdot 3 \cdot 4 \cdot 5 = 120$.

1. Способ (использование дополнительной переменной)

Наглядный способ для понимания – представить переменную в виде кружки с чаем. Чтобы поменять содержимое кружки (рис. 4.9), нужно использовать третью пустую кружку в качестве посредника или дополнительной переменной.



Рис. 4.9. Перестановка местами

Алгоритм перестановки:

Вход: $a[1]=4$
 $a[2]=10$

Перестановка: $k=a[1]$
 $a[1]=a[2]$
 $a[2]=k$

Выход: $\text{print}(a[1], a[2])$

4 10

10 4

2. Способ. (использование специальной функции **itertools**)

Методом генерирования комбинаций элементов последовательности **abc**. Этот метод принимает список в качестве входных данных и возвращает список объектов, которые содержат все возможные перестановки в форме списка.

Программа:

```
from itertools import permutations      # подключение библиотеки
                                           # перестановки с использованием
                                           # библиотечной функции

a = permutations([1, 2, 3])              # получить все перестановки из
                                           # [1, 2, 3]

for i in list(a): print i                # Распечатать полученные
                                           # перестановки
```

до:

1	2	3
---	---	---

после:

2	3	1
3	1	2
1	3	2

Если вы хотите получить перестановки **длины L**, то реализуйте это следующим образом.

Программа:

```
from itertools import permutations      # подключение библиотеки
                                           # перестановки с использованием
                                           # библиотечной функции

a = permutations([1, 2, 3],2)            # получить все перестановки из
                                           # [1, 2, 3], но вывести только 2
                                           # элемента

for i in list(a): print i                # Распечатать полученные
                                           # перестановки
```

до:

1	2	3
---	---	---

после:

3	2
2	1

Задание

1. Определите возможное количество перестановок:

Количество элементов	2	5	3	8
Число перестановок				

2. Напишите оператор:

генерация комбинаций

--

генерация чисел

--

3. Напишите программу:

Введите три числа. Выполните перестановку чисел (используя дополнительную переменную)

Необходимо создать пароль состоящий из 4 символов (числа от 0 до 3) и вывести на экран все возможные варианты данного пароля.

Необходимо создать пароль состоящий из чисел от 0 до 9 и вывести на экран все возможные варианты пароля в котором будут использоваться 4 цифры.