Министерство образования Республики Беларусь

Учреждение образования

«Брестский государственный технический университет»

Кафедра ИИТ

Лабораторная работа №4

за 1 семестр

По дисциплине: «Дискретная математика»

Тема: «Комбинаторика»

Выполнил:

Студент 2 курса

Группы ПО-4(1)

Иваненко И.Л.

Проверил:

Глущенко Т.А.

2020

Цель работы: “Научиться решать основные задачи комбинаторики”

Вариант 10

***Задание.***

1. *Номер вашего варианта равен остатку от деления вашего порядкового номера по списку на 12. Если ваш порядковый номер кратен 12, выполняете 12 вариант.*
2. *По пункту 4: решить рекуррентное соотношение и программно реализовать вычисление 5 члена числовой последовательности через рекурсию и полученное решение.*



Код программы:

def recurrence\_ratio(a, n, k1, k2):

a.append(k1 \* a[n - 1] - k2 \* a[n - 2])

print(a)

if n == 4:

print("5 член числовой последовательности: " + str(a[n]))

return

recurrence\_ratio(a, n + 1, k1, k2)

recurrence\_ratio([-3, 1], 2, 2, 1)

Результат выполнения:

[-3, 1, 5]

[-3, 1, 5, 9]

[-3, 1, 5, 9, 13]

5 член числовой последовательности: 13

1. *При разложении степени*  *коэффициенты при произведениях* *,* *, рассчитываются по формуле:* *(число перестановок с повторениями) и носят название полиномиальных или мультиномиальных коэффициентов.  
   Например, вычислим коэффициент при произведении*  *в разложении**.  
   Он равен:* *.*
2. *Написать программу генерации перестановок (рекурсивный вариант). Четные номера – реализуют лексикографический порядок генерации. Нечетные номера – реализуют антилексикографический порядок генерации.*

Код программы:

def permutations(arr):

for perm in itertools.permutations(arr):

print(perm)

def antilex\_permutations(arr):

arr = arr[::-1]

for perm in itertools.permutations(arr):

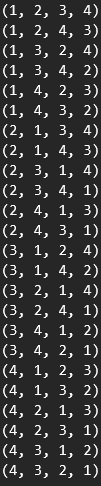
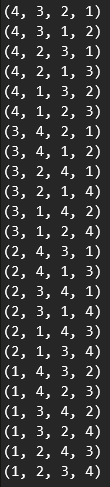
print(perm)

permutations([1, 2, 3, 4])

print("\n\n\n")

antilex\_permutations([1, 2, 3, 4])

Результат выполнения:

1. *Написать программу генерации k- элементных подмножеств n-элементного множества в лексикографическом порядке. Вход: k – мощность подмножества, n - мощность множества.*

Код программы:

def permutations(arr, k = None):

if k == None:

for perm in itertools.permutations(arr):

print(perm)

else:

for perm in itertools.permutations(arr, k):

print(perm)

def antilex\_permutations(arr, k = None):

arr = arr[::-1]

if k == None:

for perm in itertools.permutations(arr):

print(perm)

else:

for perm in itertools.permutations(arr, k):

print(perm)

permutations([1, 2, 3, 4], 2)

print("\n\n\n")

antilex\_permutations([1, 2, 3, 4], 2)

Результат выполнения:

*Вариант 10.*

1. Позывные американских радиостанций состоят из трех или четырех букв и начинаются с *k* или *w*. Сколько существует различных позывных?

В английском алфавите 26 букв. У нас есть 4 разных варианта позывных:

1) Начинается с буквы k и состоит из трех букв, Например: kab

1 \* 26 \* 26

2) Начинается с буквы w и состоит из трех букв, Например: wab

1 \* 26 \* 26

3) Начинается с буквы k и состоит из четырех букв, Например: kabc

1 \* 26 \* 26 \* 26

4) Начинается с буквы w и состоит из четырех букв, Например: wabc

1 \* 26 \* 26 \* 2

Ответ: (26^2 + 26^2 + 26^3 + 26^3 = 2\*26^2 + 3\*26^2 = 1352 + 35152 = 36504)

1. Сколько целых чисел между 1 и 2003 делится на 5, 7 или 11?

Делится на 5: 2003//5 = 400

Делится на 7: 2003//7 = 286

Делится на 11: 2003//11 = 182

Делится на 5 и 7: 2003//5\*7 = 57

Делится на 5 и 11: 2003//5\*11 = 36

Делится на 7 и 11: 2003//7\*11 = 26

Делится на 5, 7 и 11: 2003//5\*7\*11 = 5

1. К несчастью, судья на выставке цветов не разбирается в орхидеях. Если он выбирает победителей случайным образом среди 18 участниц, то сколько имеется способов вручить первый, второй и третий приз?

18 \* 17 \* 16 = 48 96

1. Разложить .



(a+b)10 = a10 + 10a9b + 45a8b2 + 120a7b3 + 210a6b4 + 252a5b5 + 210a4b6 + 120a3b7 + 45a2b8 + 10ab9 + ab10

1. Решить линейное однородное рекуррентное соотношение *2-го* порядка:  
   

an - 2an-1 + an-2 = 0

p2 – 2p + 1 = 0

p = 1

an = c + dn

a0 = c + d0 = -3

c = -3

a1 = c + d = 1

-3 + d = 1

d = 4

Ответ: -3 + 4n