Практическое задание №11.

Тема: составление программ в функциональном стиле в IDE PyCharm Community.

Цель: закрепить усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрести навыки составление программ с использованием списковых включений, итераторов, генераторов в IDE PyCharm Community.

Постановка задачи:
1.В последовательности на n целых элементов найти количество пар, для которых произведение элементов делится на 3 (элементы пары в последовательности являются соседними).

Тип алгоритма: Циклический.

Текст программы:
```
# В последовательности на n целых элементов найти количество пар,
# для которых произведение элементов делится на 3
# (элементы пары в последовательности являются соседними).
import random

lst = [random.randint(1, 1000) for x in range(random.randint(5, 10))]  # Создаем список

count = 0  # Обозначаем количество пар
for x, y in enumerate(lst):
    try:  # Проверка нужна для того, что бы пройтись по всем числам
        if (lst[x] * lst[x + 1]) % 3 == 0:
            count += 1
    except IndexError:  # При попытке взять элемент, с индексом которого нет, программа попадет суда
        continue
print(f"Сгенерированный список: " + ", ".join(list(map(str, lst))))
print(f"Количество пар, произведение которых элементов делится на 3: {count}")
```

Протокол работы программы:
Сгенерированный список: 782, 827, 58, 736, 719, 991, 342

Количество пар, произведение которых элементов делится на 3: 1

Process finished with exit code 0

Постановка задачи:
2. Составить генератор (yield), который преобразует все буквенные символы в заглавные.

Тип алгоритма: Циклический.

Текст программы:
```
# Составить генератор (yield), который преобразует все буквенные символы в
```

```
# заглавные.
import random  # Нужен нам для генерации строки
from string import ascii_lowercase as ascii_lower  # Воспользуемся буквами
ascii


def my_gen(sybols):  # Объявление функции
    for symb in sybols:
        yield symb.upper()  # Изменения символов


gen_string = ''.join(  # Создадим строку
    [ascii_lower[random.randint(0, len(ascii_lower) - 1)]
     for x in range(random.randint(5, 25))
     ])
print(f"Строка до использования генератора: {gen_string}")
g = my_gen(gen_string)
print(f"Строка после использования генератора: " + ''.join(g))  # Вывод с
помощью join, что бы не использовать next()
```

Протокол работы программы:
Строка до использования генератора: knnfepftdsbosyunewkel

Строка после использования генератора: KNNFEPFTDSBOSYUNEWKEL

Process finished with exit code 0

Вывод: закрепила усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрела навыки составления программ с использованием списковых включений, итераторов, генераторов в IDE PyCharm Community.