**7.6 Практическая работа**

Задача

Цель практической работы

Научиться:

* решать задачи с использованием циклов;
* решать задачи при помощи кортежей (tuple);
* уменьшать повторяемость кода с помощью функции range (начальным и конечным значением).

Что входит в работу

* Задача 1. Должники.
* Задача 2. Посчитай чужую зарплату...
* Задача 3. Факториал.
* Задача 4. Успеваемость в классе.
* Задача 5. Отрезок.
* Задача 6. Замечательные числа.
* Задача 7. Пропавшая карточка.

Задача 1. Должники

Что нужно сделать

В базе банка хранятся данные и должников, и законопослушных граждан. Каждому человеку система присваивает свой номер. У нас есть случайная выборка из десяти номеров. Правда, иногда база глючит и выдаёт номер с отрицательным значением. А ещё по статистике, которую собрал наш «МирПрогБанк», каждый второй пользователь брал кредит и не выплатил его вовремя, то есть является должником.

Напишите программу, которая при вводе десяти чисел определяет, сколько из них являются одновременно чётными и положительными.

Что оценивается

Задание считается успешно выполненным, если:

* результат вывода соответствует условию;
* input содержит корректное приглашение для ввода;
* переменные имеют значащие имена, не только a, b, c, d;
* задача решена с помощью конструкции for.

Задача 2. Посчитай чужую зарплату...

Что нужно сделать

Бухгалтер устала постоянно считать вручную среднегодовую зарплату сотрудников компании и, чтобы облегчить себе жизнь, обратилась к программисту.

Напишите программу, которая принимает от пользователя зарплату сотрудника за каждый из 12 месяцев и выводит на экран среднюю зарплату за год.

Что оценивается

Задание считается успешно выполненным, если:

* результат вывода соответствует условию;
* input содержит корректное приглашение для ввода;
* вывод содержит описание результата (не выведено одно число без описания);
* переменные имеют значащие имена, не только a, b, c, d;
* задача решена с помощью конструкции for.

Задача 3. Факториал

Что нужно сделать

Мы всё ближе и ближе подбираемся к серьёзной математике. Одна из классических задач — задача на нахождение факториала числа. И в будущем мы с ней ещё встретимся.

Дано натуральное число n. Напишите программу, которая находит n! (n-факториал).

Запись n! означает следующее:

n! = 1 \* 2 \* 3 \* 4 \* 5 \* … \* n

**Пример**:

Введите число: 5

Факториал числа 5 равен 120

Что оценивается

Задание считается успешно выполненным, если:

* результат вывода соответствует условию;
* input содержит корректное приглашение для ввода;
* формат вывода соответствует примеру;
* вывод содержит описание, не только результат в виде числа;
* для решения используется цикл for, а не встроенные функции или рекурсия;
* переменные имеют значащие имена, не только a, b, c, d.

Задача 4. Успеваемость в классе

Что нужно сделать

В классе N человек. Каждый из них получил за урок по информатике оценку: 3, 4 или 5, двоек сегодня не было. Напишите программу, которая получает список оценок — N чисел — и выводит на экран сообщение о том, кого сегодня больше: отличников, хорошистов или троечников.

Замечание: можно предположить, что количество отличников, хорошистов, троечников различно.

Что оценивается

Задание считается успешно выполненным, если:

* результат вывода соответствует условию;
* input содержит корректное приглашение для ввода;
* в выводе присутствует сообщение о том, кого больше;
* для решения используется цикл for, а не встроенные функции или рекурсия;
* переменные имеют значащие имена, не только a, b, c, d.

Задача 5. Отрезок

Что нужно сделать

Напишите программу, которая считывает с клавиатуры два числа: a и b, — считает и выводит в консоль среднее арифметическое всех чисел из отрезка [a; b], кратных числу 3.

Что оценивается

Задание считается успешно выполненным, если:

* результат вывода соответствует условию;
* input содержит корректное приглашение для ввода;
* в выводе присутствует описание результата;
* для решения используется конструкция for.

Задача 6. Замечательные числа

Что нужно сделать

Напишите программу, которая находит и выводит все двузначные числа, равные утроенному произведению своих цифр. К таким относятся, например, 15 и 24.

Что оценивается

Задание считается успешно выполненным, если:

* результат вывода соответствует условию;
* допускается вывод без описания, достаточно вывести числа;
* переменные имеют значащие имена, не только a, b, c, d.

Задача 7. Пропавшая карточка

Что нужно сделать

Для настольной игры используются карточки с номерами от 1 до N. Одна карточка потерялась. Напишите программу, которая поможет найти потерянную карточку, если номера оставшихся известны. Найдите её, зная номера оставшихся карточек.

Введите число карточек — N.

Затем введите N − 1 номера оставшихся карточек. Они могут быть введены в любом порядке.

**Пример:**

* Введите количество карточек: 5
* Введите номер оставшейся карточки: 1
* Введите номер оставшейся карточки: 4
* Введите номер оставшейся карточки: 5
* Введите номер оставшейся карточки: 3
* Номер пропавшей карточки: 2

Что оценивается

Задание считается успешно выполненным, если:

* результат вывода соответствует условию;
* input содержат корректные приглашения для ввода;
* переменные имеют значащие имена, не только a, b, c, d.

Что оценивается (общее)

* Использование именованных индексов, не просто i (видео 7.2).
* Ввод правильных чисел, без дополнительных действий со стороны пользователя, без использования +1 (видео 7.4).
* Правильно оформленный input, без пустого приветствия для ввода (видео 2.3).
* Переменные имеют значащие имена, не только a, b, c, d (видео 2.3).
* Пробелы после запятых, пробелы при бинарных операциях.
* Отсутствие пробелов после имён функций и перед скобками: print (), input () — неверно, print() — верно.
* Правильно оформлены блоки if-elif-else, отступы одинаковы во всех блоках одного уровня.

**Правильно**:

if a > 1:

b = 3

else:

b = 5

**Неправильно**:

If a > 1:

b = 3

else:

b = 5

* Переменные имеют корректные названия, и в качестве имён не используются имена встроенных функций ([список встроенных функций](https://docs.python.org/3.7/library/functions.html) — официальная документация).

Советы и рекомендации

* Помните о приоритете арифметических операций [PEP 8](https://docs.python.org/3.7/reference/expressions.html#operator-precedence) (мы вводим and, or).
* Почитайте руководство по написанию кода на Python:
* [PEP 8](https://www.python.org/dev/peps/pep-0008/) (на английском языке),
* [PEP 8](https://letpy.com/python-guide/pep8/) (на русском языке).
* Обратите внимание на [список встроенных функций](https://docs.python.org/3.7/library/functions.html) (официальная документация).
* Прочитайте описание конструкции [for-else](https://docs.python.org/3/tutorial/controlflow.html" \l "break-and-continue-statements-and-else-clauses-on-loops" \t "_blank).
* Прочитайте описание [enumerate](https://docs.python.org/3/library/functions.html" \l "enumerate" \t "_blank) для исключения конструкций вида for i in range(len(array)).