**ЛАБОРАТОРНАЯ РАБОТА №3**

**Работа со строками. Работа со списками.**

Цель работы: приобрести практический опыт работы при использовании строк, а также одномерных и двумерных списков.

**Содержание работы**

1. Разработал алгоритм, описал его в виде блок-схемы и составил программу для решения задачи соответствующего варианта из каждой части (I – задание со строкой; II – задание с списком).

2. Набрал текст программы и отладил её в среде разработки PyCharm.

3. Оформил отчет о проделанной работе.

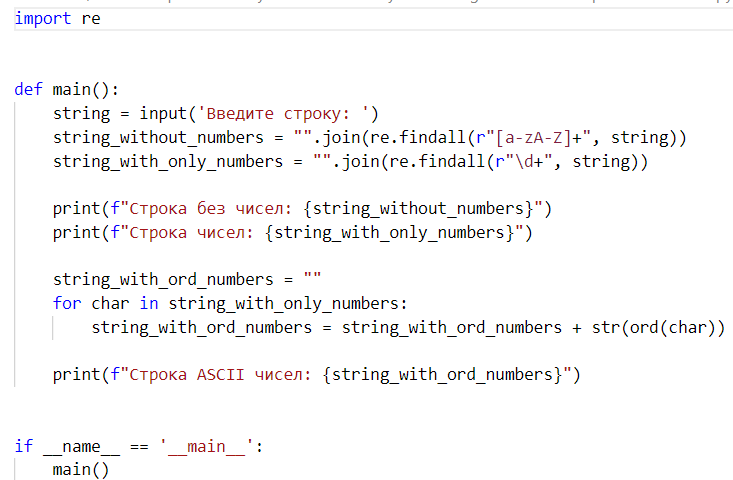
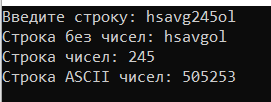
**Выполнение работы**

**Часть 1**

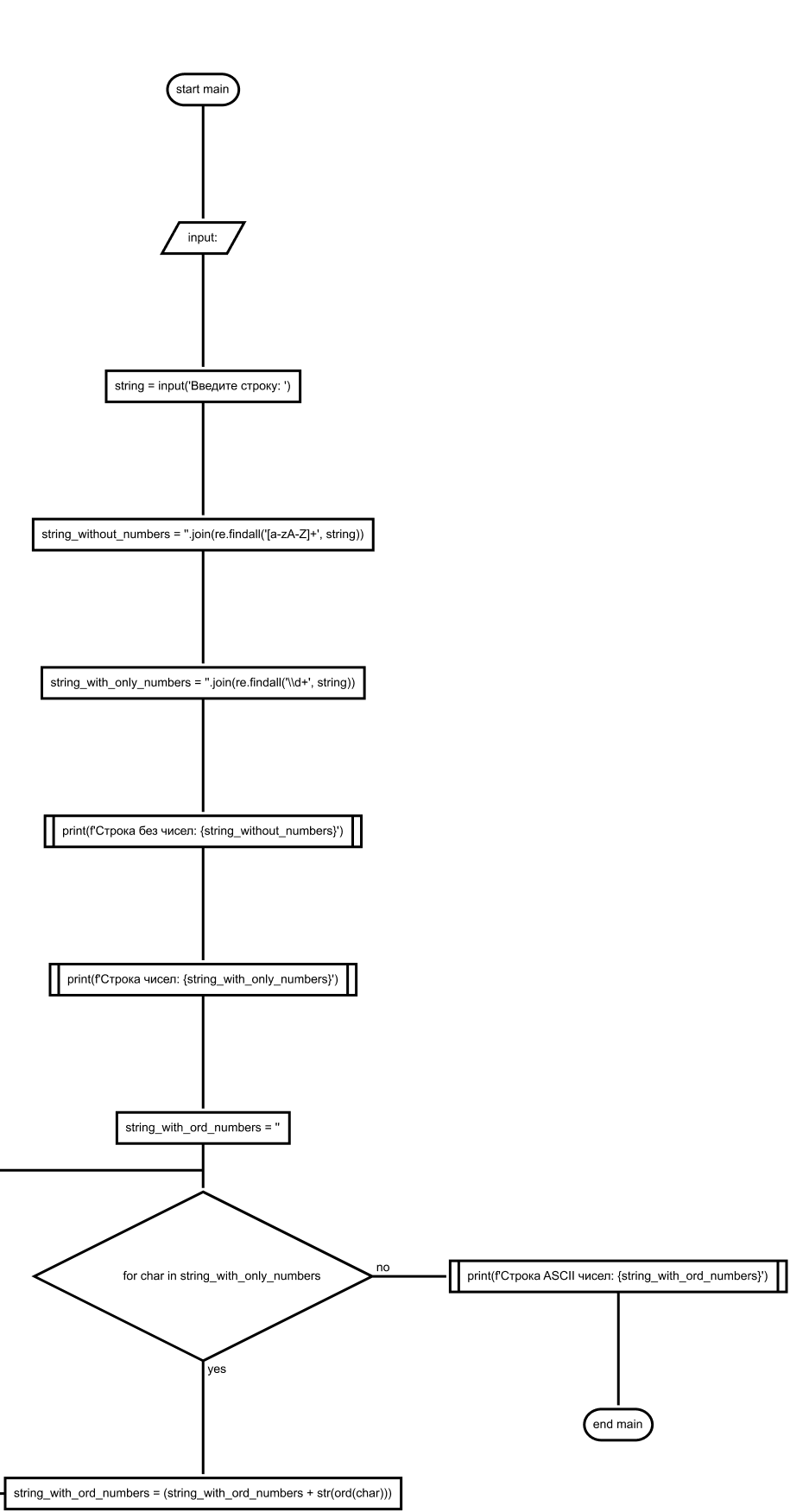
1. Вариант и условия задачи

Варант 8. Дана строка, состоящая из символов. Удалить из этой строки все цифры, записав в другую строку символы, соответствующие коды которых являются этими цифрами в таблице АSCII.

1. Код программы и результат ее работы

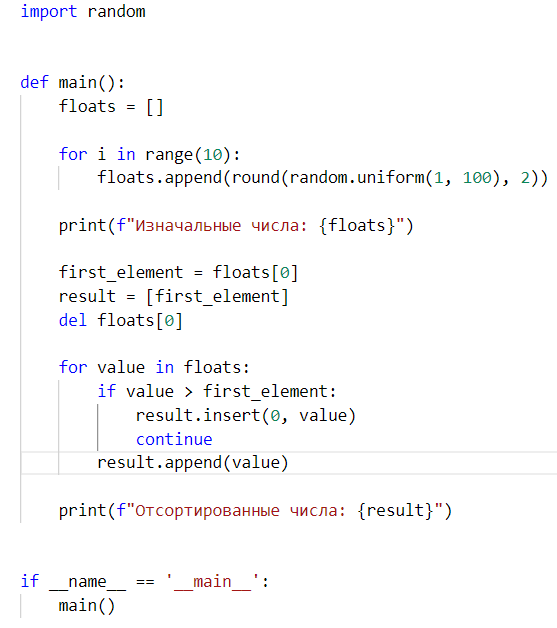
1. Блок схема



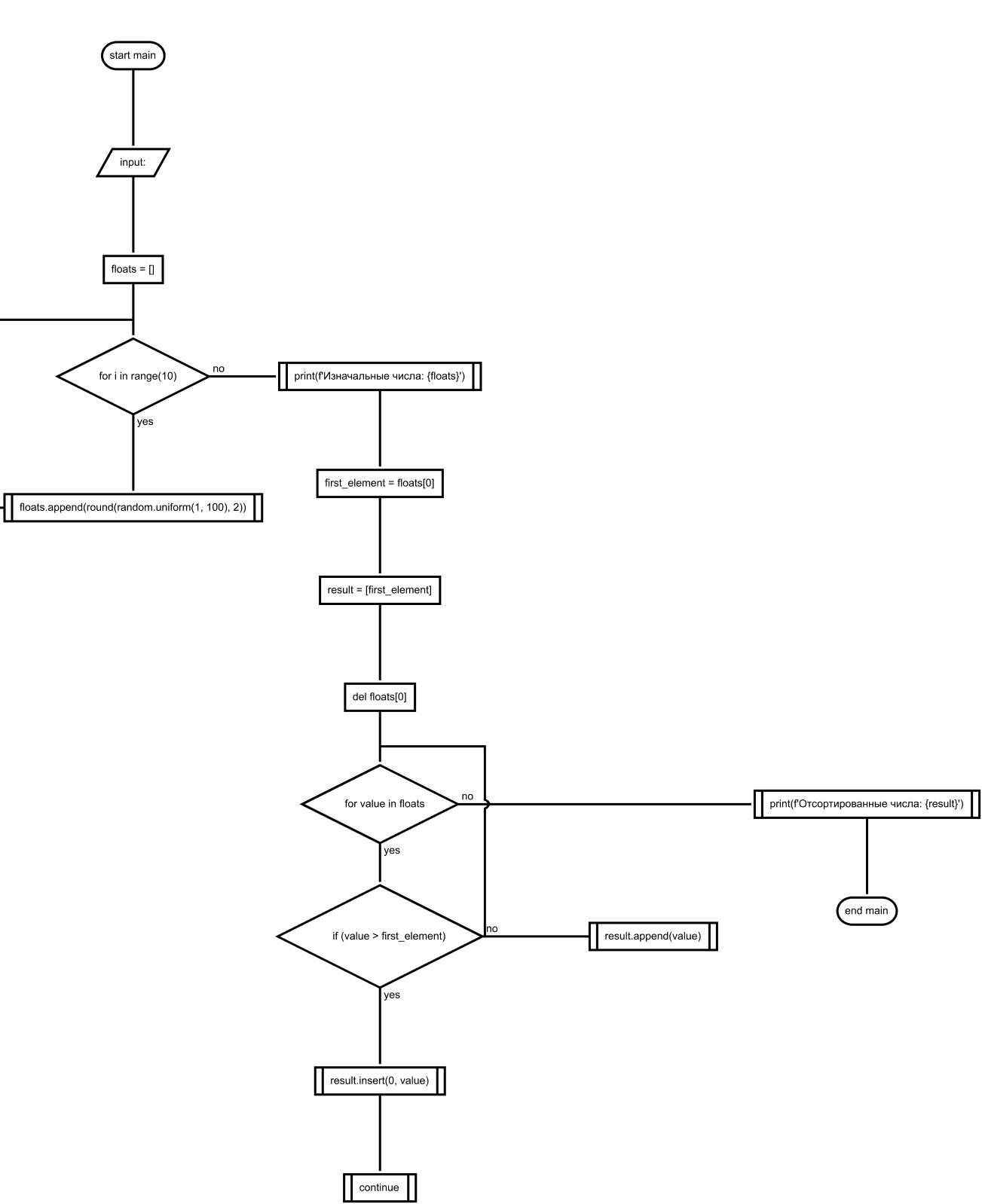
**Часть 2**

1. Вариант и условия задачи

Варант 8. Дан вещественный массив x размера n(n=10). Переставить элементы массива таким образом, чтобы вначале в массиве шла группа элементов больших первого элемента в исходном массиве, потом - группа элементов меньших или равных ему.Код программы и результат ее работы

1. Блок схема



**Контрольные вопросы**

1. Строка (str) - это упорядоченная неизменяемая последовательность символов Юникода.
2. Если в строке необходимо использовать специальные символы (например, перенос или одноименные кавычки), можно воспользоваться механизмом экранирования символов, для чего используется специальный символ "\"
3. Конкатенация, Равенство и сравнение, Индексация и срезы, chr(i), ord(c), upper()
4. Примеры операций над строками.
   * Конкатенация: s1 + s2

* Индексация и срезы: s[-1], s[0:2]: Срез включает первые 2 символа
* chr(i): Возвращает символ № i из таблицы Unicode.
* ord(c): Возвращает номер символа c из таблицы Unicode.
* upper(): Возвращает копию строки s в верхнем регистре.
* lower(): Возвращает копию строки s в нижнем регистре
* capitalize(): Возвращает копию строки с первым символом в верхнем регистре.
* title(): Возвращает копию строки, в которой первые символы каждого слова преобразованы в верхний регистр, а все остальные - в нижний регистр.
* count(t[, start[, end]]): Возвращает число вхождений строки t в строку s (или в срез s[start:end]).
* ind(t[, start[, end]]): Возвращает позицию самого первого (крайнего слева) вхождения подстроки t в строку s (или в срез s[start:end]); если подстрока t не найдена, возвращается -1.
* replace(old, new[, count]): Возвращает копию строки s, в которой каждое (но не более count, если этот аргумент определен) вхождение подстроки old замещается подстрокой new.
* split(sep=None, maxsplit=-1): Возвращает список строк, разбитых по строке sep.

1. Виды форматирования строк

* Конкатенация: print("a" + "b")
* %-форматирование: print("Hello %s" % (name))
* Template-строки: print(Template("$x $y").substitute(x=x, y=y))
* format(): print("Hello{}".format(name)
* f-строки: print(f"{x = }")

1. Смотреть пример вопроса 5
2. Список (list) — это упорядоченный набор элементов, каждый из которых имеет свой номер, или индекс, позволяющий быстро получить к нему доступ.
3. Особенности списка

* Списки это последовательности объектов произвольных типов.
* Обеспечение доступа к элементу списка по индексу.
* Переменная длина списков.
* Произвольное количество вложений списков.
* Списки принадлежат к категории изменяемых объектов.
* Возможность содержать массивы ссылок на другие объекты.

1. Примеры операций со списком

* len(l) — количество элементов в l;
* min(l) — наименьший элемент;
* max(l) — наибольший элемент;
* sum(l) — сумма чисел списка;
* for i in list() — перебирает элементы слева направо.
* index
* count
* append
* sort
* insert
* remove
* pop

Вывод: в ходе данной лабораторной работы получил практический опыт работы при использовании строк, а также одномерных и двумерных списков.