**Министерство образования и науки Российской Федерации**

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**АМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

**(ФГБОУ ВО «АмГУ»)**

Факультет математики и информатики

Кафедра информационных и управляющих систем

Направление подготовки 09.03.04 - Программная инженерия

Направленность (профиль) образовательной программы: Программная инженерия

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №5**

на тему: **Строки**

|  |  |  |
| --- | --- | --- |
| Исполнитель  студент группы 357-об | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись, дата) | Д.Е. Буханов |
| Проверил | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись, дата) | Е.В. Дегтярёв |

Благовещенск 2023

**1 ОБЩИЕ ЗАДАНИЯ**

**1 Задание.**

Извлеките из строки следующие срезы:

- первые восемь символов;

- четыре символа из центра строки;

- пять символов с конца строки.

def main():

    '''Функция main'''

    st = input("Введите строку (длина >= 8): ")

    print(f"Длина введеной строки: {len(st)}")

    if st.\_\_len\_\_() < 8:

        print("Введеная строка слишком коротка")

        return

    print(f"Первые восемь символов: '{st[:8]}'")

    print(f"Четыре символа из центра строки: '{st[int(len(st) / 2) - 2: int(len(st) / 2) + 2]}'")

    print(f"Пять символов с конца строки: '{st[-5:]}'")

if \_\_name\_\_ == "\_\_main\_\_":

    main()

Рисунок 1 – Листинг рабочей программы «file1.py»

Таблица 1

|  |  |
| --- | --- |
| Данные: | |
| Входные (ввод) | Выходные (вывод) |
| Зоревать у окна | Первые восемь символов: 'Зоревать'  Четыре символа из центра строки: 'ать '  Пять символов с конца строки: ' окна' |

**2 Задание.**

Дана строка длиной N. Вывести символы строки в обратном порядке. (Не использовать цикл)

def main():

    '''Функция main'''

    st = input("Введите строку: ")

    print(f"Ваша отзеркаленная строка: {st[::-1]}")

if \_\_name\_\_ == "\_\_main\_\_":

    main()

Рисунок 2 – Листинг рабочей программы «file2.py»

Таблица 2

|  |  |
| --- | --- |
| Данные: | |
| Входные (ввод) | Выходные (вывод) |
| Пора | Ваша отзеркаленная строка: ароП |

**3 Задание.**

Дана строка длиной N (N – четное число). Вывести символы с четными номерами в порядке возрастания их номеров:

def main():

    '''Функция main'''

    st = input("Введите строку: ")

    print(f"Ваши символы с четными номерами: {list(st[1::2])}")

if \_\_name\_\_ == "\_\_main\_\_":

    main()

Рисунок 3 – Листинг рабочей программы «file3.py»

Таблица 3

|  |  |
| --- | --- |
| Данные: | |
| Входные (ввод) | Выходные (вывод) |
| Роза | Ваши символы с четными номерами: ['о', 'а'] |

**4 Задание.**

Дана строка длиной N. Вывести сначала символы с четными номерами (в порядке возрастания номеров), а затем – символы с нечетными номерами (также в порядке возрастания номеров)

def main():

    '''Функция main'''

    st = input("Введите строку: ")

    print(f"Ваши символы с четными номерами: {list(st[1::2])}")

    print(f"Ваши символы с не четными номерами: {list(st[::2])}")

if \_\_name\_\_ == "\_\_main\_\_":

    main()

Рисунок 4 – Листинг рабочей программы «file4.py»

Таблица 4

|  |  |
| --- | --- |
| Данные: | |
| Входные (ввод) | Выходные (вывод) |
| Роза | Ваши символы с четными номерами: ['о', 'а']  Ваши символы с не четными номерами: ['Р', 'з'] |

**5 Задание.**

Преобразовать дату в «компьютерном» представлении (системную дату: 2018-03-26) в «российский» формат, т. е. день/месяц/год (например, 26/03/2018). Известно, что на год выделено всегда 4 цифры, а на день и месяц – всегда 2 цифры.

import datetime

def main():

    '''Функция main'''

    date\_now = str(datetime.date.today())

    date\_now = date\_now.split("-")

    print(f"Дата в формате дд.мм.гггг: {date\_now[-1]}.{date\_now[1]}.{date\_now[0]}")

if \_\_name\_\_ == "\_\_main\_\_":

    main()

Рисунок 5 – Листинг рабочей программы «file5.py»

Таблица 5

|  |  |
| --- | --- |
| Данные: | |
| Входные (ввод) | Выходные (вывод) |
| Дата с в системе | Дата в формате дд.мм.гггг: 06.11.2023 |

**6 Задание.**

Вывести адрес файла и «разобрать» его на части, разделенные знаком ‘/’. Каждую часть вывести в отдельной строке

def main():

    '''Функция main'''

    roat\_to\_file = input("Введите путь к файлу: ")

    if "\\" in roat\_to\_file:

        roat\_to\_file = roat\_to\_file.replace("\\", "/")

    roat\_to\_file = roat\_to\_file.split("/")

    for i in roat\_to\_file:

        print(i)

if \_\_name\_\_ == "\_\_main\_\_":

    main()

Рисунок 6 – Листинг рабочей программы «file6.py»

Таблица 6

|  |  |
| --- | --- |
| Данные: | |
| Входные (ввод) | Выходные (вывод) |
| C:\Windows\regedit.exe | C:  Windows  regedit.exe |

**7 Задание.**

Вывести строку, в которой записана сумма натуральных чисел, например, ‘1+25+3’. Вычислите это выражение. Использовать строковые функции языка.

def main():

    '''Функция main'''

    prim = input("Введите выражение: ")

    prim = [int(i) for i in prim.split("+")]

    print(f"Сумма выражения: {sum(prim)}")

if \_\_name\_\_ == "\_\_main\_\_":

    main()

Рисунок 7 – Листинг рабочей программы «file7.py»

Таблица 7

|  |  |
| --- | --- |
| Данные: | |
| Входные (ввод) | Выходные (вывод) |
| 3+3+4 | Сумма выражения: 10 |

**8 Задание.**

Определить, является ли введенная строка палиндромом («перевертышем») типа ABBA, kazak и пр.

def main():

    '''Функция main'''

    S = input("Введите строку: ")

    L = len(S)

    flag = True

    for i in range(1, L // 2):

        if S[i] == S[L - i + 1]:

            k = True

        else:

            k = False

        flag = flag \* k

    if flag:

        res = "Палиндром"

    else:

        res = "Не палиндром"

    print(res)

if \_\_name\_\_ == "\_\_main\_\_":

    main()

Рисунок 8 – Листинг рабочей программы «file8.py»

Таблица 8

|  |  |
| --- | --- |
| Данные: | |
| Входные (ввод) | Выходные (вывод) |
| арозаупаланалапуазора | Палиндром |
| 1134 | Не палиндром |
| 1001 | Палиндром |

**8 Задание.**

Допустим, есть какое-то объявление и несколько разных людей, которым нужно это объявление отправить. Для этого создается заготовка с содержанием объявления, внутри которого есть ряд изменяющихся параметров: имена людей и названия событий. Вывести один вариант итого объявления на экран. Для задания имен и названий использовать форматирование при помощи метода format.

MESSAGE\_CONST = '''Уважаемый (ая), {name\_person}!

Приглашаем Вас на {name\_quest}.

Дата события: {date\_quest} мая. С уважением, {name\_speaker}.'''

def main():

    '''Функция main'''

    name\_quest = input("Введите название мероприятия: ").strip()

    name\_speaker = input("Кто отправляет сообщение (имя): ").strip()

    names\_person = []

    for i in range(1, 6):

        names\_person.append(input(f"Введите имя {i} участника: ").strip())

    for i in range(len(names\_person)):

        print(MESSAGE\_CONST.format(name\_person = names\_person[i],

                                   name\_quest = name\_quest,

                                   date\_quest = i + 1,

                                   name\_speaker = name\_speaker))

        print()

if \_\_name\_\_ == "\_\_main\_\_":

    main()

Рисунок 9 – Листинг рабочей программы «file9.py»

Таблица 9

|  |  |
| --- | --- |
| Данные: | |
| Входные (ввод) | Выходные (вывод) |
| Свадьба  Гавриил Михайлович  Зубенко Михаил  Петрович Маузер  Олег Хоркавич  Ольга Бузова  Даниил Милохин | Уважаемый (ая), Зубенко Михаил!  Приглашаем Вас на Свадьба.  Дата события: 1 мая. С уважением, Гавриил Михайлович.  Уважаемый (ая), Петрович Маузер!  Приглашаем Вас на Свадьба.  Дата события: 2 мая. С уважением, Гавриил Михайлович.  Уважаемый (ая), Олег Хоркавич!  Приглашаем Вас на Свадьба.  Дата события: 3 мая. С уважением, Гавриил Михайлович.  Уважаемый (ая), Ольга Бузова!  Приглашаем Вас на Свадьба.  Дата события: 4 мая. С уважением, Гавриил Михайлович.  Уважаемый (ая), Даниил Милохин!  Приглашаем Вас на Свадьба.  Дата события: 5 мая. С уважением, Гавриил Михайлович. |

**2 ИНДИВИДУЛЬНОЕ ЗАДАНИЕ**

**2.1 Задание.**

Записать элементы массива X, удовлетворяющие условию 1<= x <= 2, подряд в массив Y. Поменять местами максимальный и минимальный элементы в массиве.

def main():

    '''Функция main'''

    x = []

    while True:

        number = input("Введите число (break - остановка): ").rstrip('\n')

        if number == "break":

            break

        x.append(float(number))

    print(f"Полученный список {x}")

    Y = []

    for i in x:

        if 1 <= i <= 2:

            Y.append(i)

    if len(Y) <= 1:

        print(f"Полученный список c элементами с из диапазона [1, 2]: {Y}")

        return

    in\_max = Y.index(max(Y))

    in\_min = Y.index(min(Y))

    Y[in\_min], Y[in\_max] = max(Y), min(Y)

    print(f"Полученный список c элементами с из диапазона [1, 2]: {Y}")

if \_\_name\_\_ == "\_\_main\_\_":

    main()

Рисунок 10 – Листинг рабочей программы «file10.py»

Таблица 10

|  |  |
| --- | --- |
| Данные: | |
| Входные (ввод) | Выходные (вывод) |
| 1  2  3  1.2  1.4  5  6  7  1.5  1.9  1.0  1.7  break | Полученный список [1.0, 2.0, 3.0, 1.2, 1.4, 5.0, 6.0, 7.0, 1.5, 1.9, 1.0, 1.7]  Полученный список c элементами с из диапазона [1, 2]: [2.0, 1.0, 1.2, 1.4, 1.5, 1.9, 1.0, 1.7] |



Блок схема 1

**2.2 Задание.**

Удалить из строки все запятые

def main():

    '''Функция main'''

    while True:

        st = input("Введите строку (break для завершения ввода): ")

        if st == "break":

            return

        print(f"Строка до преобразований: {st}")

        st = st.replace(",", " ")

        print(f"Строка после преобразований: {st}")

if \_\_name\_\_ == "\_\_main\_\_":

    main()

Рисунок 11 – Листинг рабочей программы «file11.py»

Таблица 11

|  |  |
| --- | --- |
| Данные: | |
| Входные (ввод) | Выходные (вывод) |
| ав, ав, ыв  break | Строка до преобразований: ав, ав, ыв  Строка после преобразований: ав ав ыв |



Блок схема 2