

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**«САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»**
Факультет безопасности информационных технологий

Дисциплина:

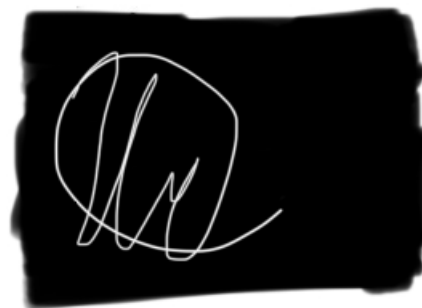
«Программирование»

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №5

Выполнил:

Студент группы N3149

Ильчук Денис.



Проверила:

Горлина А.В.

№ варианта	Формат данных в строке
8	Дата и время

№ варианта	Формат файла
5	<div><div><div>Начало файла</div><div><div>cnt</div><div>i₀</div><div>o₀</div><div>...</div><div>i_{N-1}</div><div>o_{N-1}</div><div>data</div><div>p₀</div><div>s₀</div><div>...</div><div>data</div><div>p_{N-1}</div><div>s_{N-1}</div><div>Конец файла</div></div></div><p>Содержимое файла:</p><p>Поле cnt (целое без знака [4 байта]). Количество пар полей индексов и смещений строк, расположенных сразу после этого поля.</p><p>Индексы i₀, ..., i_{N-1} (целые без знака [2 байта]). Индексы соответствующих строк s₀, ..., s_{N-1}, определяющие порядок строк в списке.</p><p>Смещения o₀, ..., o_{N-1} (целые без знака [4 байта]). Смещения полей длин соответствующих строк в файле. Каждый элемент массива o_i содержит смещение в байтах от начала файла поля p_i, за которым следует строка s_i.</p><p>Размеры строк p₀, ..., p_{N-1} (целые без знака [4 байта]). Размеры строк в байтах.</p><p>Строки s₀, ..., s_{N-1}. Строки в кодировке UTF-8 без нулевого символа в конце. Размер каждой строки в байтах задается полем p_i.</p><p>Области произвольных данных data. Области произвольного размера (возможно, нулевого).</p></div>

Примеры работы программы на различных исходных данных (скриншоты)

```
> python3 lab5dviN3149.py -n 5 2.bin
```

Строка [0], по адресу: (0x00000000) = 13.02.6728

Строка [1], по адресу: (0x0000000e) = 30.02.7968

Строка [2], по адресу: (0x0000001c) = 14.00.32

Строка [3], по адресу: (0x00000028) = 13.25.59

Строка [4], по адресу: (0x00000034) = 13.12.1721

```
> python3 lab5dviN3149.py 2.bin
```

Строка [0], по адресу: (0x00000000) = 16.34.15

Строка [1], по адресу: (0x0000000c) = 09.17.18

Строка [2], по адресу: (0x00000018) = 05.45.53

Строка [3], по адресу: (0x00000024) = 07.07.2562

Строка [4], по адресу: (0x00000032) = 11.08.30

Строка [5], по адресу: (0x0000003e) = 07.14.43

Строка [6], по адресу: (0x0000004a) = 14.08.2345

Строка [7], по адресу: (0x00000058) = 23.31.02

Строка [8], по адресу: (0x00000064) = 03.10.1782

Строка [9], по адресу: (0x00000072) = 10.08.6231

Строка [10], по адресу: (0x00000080) = 01.58.10

Строка [11], по адресу: (0x0000008c) = 04.41.08

Строка [12], по адресу: (0x00000098) = 27.06.4310

Строка [13], по адресу: (0x000000a6) = 26.05.8339

Строка [14], по адресу: (0x000000b4) = 04.42.31

Строка [15], по адресу: (0x000000c0) = 18.34.22

Строка [16], по адресу: (0x000000cc) = 06.08.59

Строка [17], по адресу: (0x000000d8) = 09.51.37

Строка [18], по адресу: (0x000000e4) = 20.37.12

Строка [19], по адресу: (0x000000f0) = 11.26.17

Строка [20], по адресу: (0x000000fc) = 16.36.08

Строка [21], по адресу: (0x00000108) = 01.10.7887

Строка [22], по адресу: (0x00000116) = 23.04.7219

Строка [23], по адресу: (0x00000124) = 12.02.44

Строка [24], по адресу: (0x00000130) = 20.09.7063

Строка [25], по адресу: (0x0000013e) = 13.12.6465

Строка [26], по адресу: (0x0000014c) = 25.05.4594

Строка [27], по адресу: (0x0000015a) = 21.08.3198

Строка [28], по адресу: (0x00000168) = 27.10.3430

Строка [29], по адресу: (0x00000176) = 05.27.17

Строка [335], по адресу: (0x00001108) = 01.05.5231
Строка [336], по адресу: (0x00001116) = 06.02.5758
Строка [337], по адресу: (0x00001124) = 23.10.3312
Строка [338], по адресу: (0x00001132) = 27.06.6172
Строка [339], по адресу: (0x00001140) = 31.02.1464
Строка [340], по адресу: (0x0000114e) = 20.17.11
Строка [341], по адресу: (0x0000115a) = 00.51.03
Строка [342], по адресу: (0x00001166) = 07.55.01
Строка [343], по адресу: (0x00001172) = 03.44.00

```

import sys
import os
import random
import struct
from datetime import datetime

# Функция для генерации случайной строки формата даты или времени
def generate_random_date_time():
    if random.choice([True, False]):
        # Генерация времени в формате чч.мм.сс
        return "{:02d}.{:02d}.{:02d}".format(random.randint(0, 23),
random.randint(0, 59), random.randint(0, 59))
    else:
        # Генерация даты в формате дд.мм.гггг
        return "{:02d}.{:02d}.{:04d}".format(random.randint(1, 31),
random.randint(1, 12), random.randint(1000, 9999))

# Функция для записи данных в бинарный файл
def write_to_file(filename, n, io_array, ls_array):
    with open(filename, "wb") as file:
        # Запись n
        file.write(struct.pack("i", n))

        # Запись области i, o
        for io in io_array:
            file.write(struct.pack("ii", io[0], io[1]))

        # Запись пустой области в 100 байт
        file.write(b'\x00' * 100)

        # Запись области k, s
        for ls in ls_array:
            file.write(struct.pack("i", ls[0]))
            file.write(ls[1].encode())

def print_data(n, io_array, ls_array):
    for i in range(n):
        index, offset = io_array[i]
        length, string = ls_array[i]
        print(f"Строка [{index}], по адресу: (0x{offset:08x}) = {string}")

# Основная функция программы
def main():
    name = sys.argv[0]
    args = sys.argv[1:]

    if not args:
        print(f"Используйте: {name} [-n <количество_строк>] <имя_файла> || {name} -v")
        sys.exit(1)

    if args[0] == "-v":
        print("\tДенис Ильчук Витальевич, гр. N3149\n\tВариант: 8-5\n")
        sys.exit(0)

```



```

if args[0] == "-n":
    if len(args) != 3:
        print(f"Используйте: {name} -n <количество_строк> <имя_файла>")
        sys.exit(1)
    try:
        n = int(args[1])
    except ValueError:
        print("Количество строк, должно быть целым числом")
        sys.exit(1)
    if n <= 0:
        print("Количество строк, должно быть натуральным числом")
        sys.exit(1)
    filename = args[2]
else:
    if len(args) != 1:
        print(f"Используйте: {name} <имя_файла>")
        sys.exit(1)
    filename = args[0]
    n = random.randint(10, 1000)

io_array = []
ls_array = []

current_offset = 0

for i in range(n):
    random_string = generate_random_date_time()
    string_length = len(random_string)

    ls_array.append((string_length, random_string))
    io_array.append((i, current_offset))

    current_offset += struct.calcsize("i") + string_length
print_data(n, io_array, ls_array)
write_to_file(filename, n, io_array, ls_array)

if __name__ == "__main__":
    main()

```