

Отчет

Лабораторная работа 3

Проектирование формата файлов

Операционная система ОС Windows, 64-разрядная ОС

Задание ЛЗ.1.

Выберите сигнатуру для собственного формата файлов, которая будет использоваться во всех дальнейших работах (6 байт). Разработайте программу-кодировщик, состоящую из двух частей — кодировщика и декодировщика.

Выбранная сигнатура - 52 41 52 4e 49 4b.

1. Кодировщик по заданному файлу  $Q$  создаёт архив  $R$  формата, описанного в таблице ЛЗ.1

Нулевая версия формата архива

Таблица ЛЗ.1

Смещение поля (байты)	Размер поля (байты)	Описание поля
0	6	Сигнатура формата (выбранное выше 6-байтовое значение)
6	2	Версия формата (беззнаковое 16-битное целое): для задания ЛЗ.№1 — 0
8	8	Исходная длина $n$ файла в байтах (беззнаковое 64-битное целое)
16	$n$	«Сырые» данные исходного файла (несжатый текст)

2. Декодировщик по заданному архиву  $R$ : — проверяет сигнатуру на соответствие выбранной в ЛЗ.№1 и версию формата; — при корректных сигнатуре и версии восстанавливает файл  $Q$  (который должен совпадать с исходным  $Q$ ) по данным архива  $R$ .

Код: <https://pastebin.com/VFytWef0>

```
import struct
```

```
# сигнатура формата (6 байт)
SIGNATURE = b'RARNIK'
VERSION = 0
```

```
# кодировщик: создаем архив R из файла Q
```

```

def encode(input_file, output_file):
    with open(input_file, 'rb') as f:
        file_data = f.read()

    file_size = len(file_data)

    # формируем архивный файл
    with open(output_file, 'wb') as f:
        # записываем сигнатуру (6 байт)
        f.write(SIGNATURE)
        # записываем версию формата (2 байта, беззнаковое короткое целое)
        f.write(struct.pack('H', VERSION)) # H означает unsigned short (2
байта)
        # записываем длину исходного файла (8 байт, беззнаковое длинное целое)
        f.write(struct.pack('Q', file_size)) # Q означает unsigned long long
(8 байт)
        # записываем сырые данные исходного файла
        f.write(file_data)
    print(f"Файл {input_file} успешно заархивирован в {output_file}")

# декодер: восстанавливаем файл Q из архива R
def decode(input_archive, output_file):
    with open(input_archive, 'rb') as f:
        # читаем и проверяем сигнатуру
        signature = f.read(6)
        if signature != SIGNATURE:
            print("Ошибка: неверная сигнатура файла!")
            return

        # читаем и проверяем версию формата
        version = struct.unpack('H', f.read(2))[0]
        if version != VERSION:
            print("Ошибка: неподдерживаемая версия формата!")
            return

        # читаем длину исходного файла
        file_size = struct.unpack('Q', f.read(8))[0]

        # читаем "сырые" данные исходного файла
        file_data = f.read(file_size)

        # восстанавливаем исходный файл
        with open(output_file, 'wb') as out_f:
            out_f.write(file_data)

    print(f"Архив {input_archive} успешно разархивирован в {output_file}")

encode("otik-labs.pdf", 'archived.team5')
decode('archived.team5', 'otik-labs-dearchived.pdf')

```

Проверьте при помощи шестнадцатеричного просмотрщика/редактора, что поля заголовка созданного архива соответствуют таблице ЛЗ.1

Исходный файл:

00000000	00	01	02	03	04	05	06	07	08	09	0a	0b	0c	0d	0e	0f
00000000	25	50	44	46	2d	31	2e	35	0a	25	d0	d4	c5	d8	0a	31
00000010	20	30	20	6f	62	6a	0a	3c	3c	0a	2f	4c	65	6e	67	74
00000020	68	20	38	31	37	20	20	20	20	20	20	20	0a	2f	46	69
00000030	6c	74	65	72	20	2f	46	6c	61	74	65	44	65	63	6f	64
00000040	65	0a	3e	3e	0a	73	74	72	65	61	6d	0a	78	da	6d	55
00000050	4d	6f	e2	30	10	bd	e7	57	78	0f	95	da	03	c5	76	3e
00000060	88	2b	84	64	27	20	f5	b0	db	aa	a0	d5	5e	69	62	ba
00000070	91	4a	82	02	1c	fa	ef	d7	6f	c6	94	ed	6a	0f	a0	e7
00000080	f1	f3	cc	9b	37	89	73	f3	ed	79	3d	b1	ed	f0	ea	27
00000090	e9	bd	14	2f	fe	38	9c	c7	c6	4f	aa	ef	db	43	72	73
000000a0	53	0f	cd	79	ef	fb	d3	0f	ef	5b	df	5e	76	8f	0f	e2
000000b0	79	1c	9a	b5	3f	89	db	ea	b1	7e	ec	bb	d3	5d	20	3f
000000c0	f6	cd	fb	b9	f5	17	d6	ff	49	ce	bf	75	fd	95	82	3a
000000d0	e2	76	e3	7f	4d	36	da	4e	24	18	9b	ee	f4	1e	76	ae
000000e0	41	11	90	08	48	d0	f6	4f	3f	1e	bb	a1	7f	10	ea	5e
000000f0	4a	19	02	cb	be	ad	86	3d	54	1e	93	69	ac	24	a6	97
00000100	da	bb	ae	6f	c7	58	4e	bc	a2	78	a2	b4	68	bb	e6	14

Архив:

00000000	00	01	02	03	04	05	06	07	08	09	0a	0b	0c	0d	0e	0f
00000000	52	41	52	4e	49	4b	00	00	57	31	08	00	00	00	00	00
00000010	25	50	44	46	2d	31	2e	35	0a	25	d0	d4	c5	d8	0a	31
00000020	20	30	20	6f	62	6a	0a	3c	3c	0a	2f	4c	65	6e	67	74
00000030	68	20	38	31	37	20	20	20	20	20	20	20	0a	2f	46	69
00000040	6c	74	65	72	20	2f	46	6c	61	74	65	44	65	63	6f	64
00000050	65	0a	3e	3e	0a	73	74	72	65	61	6d	0a	78	da	6d	55
00000060	4d	6f	e2	30	10	bd	e7	57	78	0f	95	da	03	c5	76	3e
00000070	88	2b	84	64	27	20	f5	b0	db	aa	a0	d5	5e	69	62	ba
00000080	91	4a	82	02	1c	fa	ef	d7	6f	c6	94	ed	6a	0f	a0	e7
00000090	f1	f3	cc	9b	37	89	73	f3	ed	79	3d	b1	ed	f0	ea	27
000000a0	e9	bd	14	2f	fe	38	9c	c7	c6	4f	aa	ef	db	43	72	73
000000b0	53	0f	cd	79	ef	fb	d3	0f	ef	5b	df	5e	76	8f	0f	e2
000000c0	79	1c	9a	b5	3f	89	db	ea	b1	7e	ec	bb	d3	5d	20	3f
000000d0	f6	cd	fb	b9	f5	17	d6	ff	49	ce	bf	75	fd	95	82	3a
000000e0	e2	76	e3	7f	4d	36	da	4e	24	18	9b	ee	f4	1e	76	ae
000000f0	41	11	90	08	48	d0	f6	4f	3f	1e	bb	a1	7f	10	ea	5e
00000100	4a	19	02	cb	be	ad	86	3d	54	1e	93	69	ac	24	a6	97
00000110	da	bb	ae	6f	c7	58	4e	bc	a2	78	a2	b4	68	bb	e6	14

00000000	00	01	02	03	04	05	06	07	08	09	0a	0b	0c	0d	0e	0f
00000000	52	41	52	4e	49	4b	00	00	57	31	08	00	00	00	00	00
00000010	25	50	44	46	2d	31	2e	35	0a	25	d0	d4	c5	d8	0a	31
00000020	20	30	20	6f	62	6a	0a	3c	3c	0a	2f	4c	65	6e	67	74
00000030	68	20	38	31	37	20	20	20	20	20	20	20	0a	2f	46	69
00000040	6c	74	65	72	20	2f	46	6c	61	74	65	44	65	63	6f	64
00000050	65	0a	3e	3e	0a	73	74	72	65	61	6d	0a	78	da	6d	55
00000060	4d	6f	e2	30	10	bd	e7	57	78	0f	95	da	03	c5	76	3e
00000070	88	2b	84	64	27	20	f5	b0	db	aa	a0	d5	5e	69	62	ba
00000080	91	4a	82	02	1c	fa	ef	d7	6f	c6	94	ed	6a	0f	a0	e7
00000090	f1	f3	cc	9b	37	89	73	f3	ed	79	3d	b1	ed	f0	ea	27
000000a0	e9	bd	14	2f	fe	38	9c	c7	c6	4f	aa	ef	db	43	72	73
000000b0	53	0f	cd	79	ef	fb	d3	0f	ef	5b	df	5e	76	8f	0f	e2
000000c0	79	1c	9a	b5	3f	89	db	ea	b1	7e	ec	bb	d3	5d	20	3f
000000d0	f6	cd	fb	b9	f5	17	d6	ff	49	ce	bf	75	fd	95	82	3a
000000e0	e2	76	e3	7f	4d	36	da	4e	24	18	9b	ee	f4	1e	76	ae
000000f0	41	11	90	08	48	d0	f6	4f	3f	1e	bb	a1	7f	10	ea	5e
00000100	4a	19	02	cb	be	ad	86	3d	54	1e	93	69	ac	24	a6	97
00000110	da	bb	ae	6f	c7	58	4e	bc	a2	78	a2	b4	68	bb	e6	14

- сигнатура
- код защиты от помех
- Версия формата(Мажорная/Минорна)
- количество файлов и папок
- код сжатия

## Разархивированный файл:

otik-labs-dearchived.pdf																
00000000	00	01	02	03	04	05	06	07	08	09	0a	0b	0c	0d	0e	0f
00000000	25	50	44	46	2d	31	2e	35	0a	25	d0	d4	c5	d8	0a	31
00000010	20	30	20	6f	62	6a	0a	3c	3c	0a	2f	4c	65	6e	67	74
00000020	68	20	38	31	37	20	20	20	20	20	20	20	0a	2f	46	69
00000030	6c	74	65	72	20	2f	46	6c	61	74	65	44	65	63	6f	64
00000040	65	0a	3e	3e	0a	73	74	72	65	61	6d	0a	78	da	6d	55
00000050	4d	6f	e2	30	10	bd	e7	57	78	0f	95	da	03	c5	76	3e
00000060	88	2b	84	64	27	20	f5	b0	db	aa	a0	d5	5e	69	62	ba
00000070	91	4a	82	02	1c	fa	ef	d7	6f	c6	94	ed	6a	0f	a0	e7
00000080	f1	f3	cc	9b	37	89	73	f3	ed	79	3d	b1	ed	f0	ea	27
00000090	e9	bd	14	2f	fe	38	9c	c7	c6	4f	aa	ef	db	43	72	73
000000a0	53	0f	cd	79	ef	fb	d3	0f	ef	5b	df	5e	76	8f	0f	e2
000000b0	79	1c	9a	b5	3f	89	db	ea	b1	7e	ec	bb	d3	5d	20	3f
000000c0	f6	cd	fb	b9	f5	17	d6	ff	49	ce	bf	75	fd	95	82	3a

## Сравнение файлов:

```
D:\pyprj\lab3>FC otik-labs.pdf otik-labs-dearchived.pdf
Сравнение файлов otik-labs.pdf и OTIK-LABS-DEARCHIVED.PDF
FC: различия не найдены
```

## Задание Л3.2.

Разработайте и опишите формат файла-архива для дальнейших работ. Архив обязательно содержит заголовок и закодированные данные. Заголовок обязательно включает:

- сигнатуру;
- ненулевую версию формата;
- коды использованных алгоритмов сжатия и защиты от помех;
- исходную длину файла в символах кодирования (то есть байтах);

а также любые необходимые, по мнению автора, поля. Формат должен предусматривать возможность добавления служебных данных для различных алгоритмов сжатия и защиты от помех (например, массив частот для методов сжатия без учёта контекста) без кардинальной его переработки. В качестве кодов алгоритмов, соответствующих отсутствию сжатия и защиты от помех, используйте значение 0.

Смещение поля (байты)	Размер поля (байты)	Описание поля
0	6	Сигнатура формата 52 41 52 4e 49 4b
6	1	Версия формата - мажорная
7	1	Версия формата - минорная
8	1	Код сжатия

9	1	Код защиты от помех
10	4	Количество файлов и папок

Описывается для каждого файла/папки в архиве

	2	Длина пути для каждого файла и папки
	$x_i$	Путь файла или папки
	1	Тип (файл или папка)

Описывается для каждого файла в архиве

	4	Защита от помех
	8	Размер сжатых данных
	$l_i$	Сжатые данные

### Задание Л3.3.

**Разработайте программу-кодировщик, создающую архив формата, разработанного в задании Л3.2.**

Код: <https://pastebin.com/esapMqzn>

```
import os
import struct

# сигнатура формата (6 байт)
SIGNATURE = b'RARNIK'

# константы для формата сжатия и защиты от помех
NO_COMPRESSION = 0
NO_PROTECTION = 0

# версия формата
FORMAT_VERSION = (1, 1) # версия 1.1

# функция для получения всех файлов и папок
def get_files_and_folders(root_dir):
    file_list = []
    if os.path.isfile(root_dir):
        file_size = os.path.getsize(root_dir)
        # добавление файла
        file_list.append((os.path.basename(root_dir), 0, file_size))
    else:
        for root, dirs, files in os.walk(root_dir):
            # папка, тип = 1, размер = 0
            for directory in dirs:
                dir_path = os.path.relpath(os.path.join(root, directory), root_dir)
                file_list.append((dir_path, 1, 0))
            # файл, тип = 0, размер = размеру файла
            for file in files:
                file_path = os.path.relpath(os.path.join(root, file), root_dir)
                file_size = os.path.getsize(os.path.join(root_dir, file_path))
                file_list.append((file_path, 0, file_size))
    return file_list

# функция сжатия данных
def compress_data(data, compression):
```

```

    return data

# функция декомпрессии данных
def decompress_data(data, compression):
    return data

# функция защиты от помех
def calculate_checksum(data, protection):
    return 0

# кодер: архивирует директории или файлы
def encode(input_path, output_file, compression=NO_COMPRESSION, protection=NO_PROTECTION):
    files_and_folders = get_files_and_folders(input_path)

    with open(output_file, 'wb') as f:
        # записываем сигнатуру (6 байт)
        f.write(SIGNATURE)
        # записываем версию формата (1 байт для мажорной версии и 1 байт для минорной)
        f.write(struct.pack('B', FORMAT_VERSION[0])) # B = unsigned char (1 байт)
        f.write(struct.pack('B', FORMAT_VERSION[1]))
        # записываем коды сжатия и защиты от помех (по 1 байту на каждый)
        f.write(struct.pack('B', compression))
        f.write(struct.pack('B', protection))
        # записываем количество файлов и папок (4 байта, беззнаковое целое)
        f.write(struct.pack('I', len(files_and_folders))) # I = unsigned int (4 байта)

        # записываем метаданные для каждого файла/папки
        for file_path, file_type, _ in files_and_folders:
            # записываем длину пути и сам путь (длина + строка)
            encoded_path = file_path.encode('utf-8')
            f.write(struct.pack('H', len(encoded_path))) # H = unsigned short (2 байта)
            f.write(encoded_path)
            # записываем тип (1 байт: 0 для файла, 1 для папки)
            f.write(struct.pack('B', file_type))

        # записываем данные файлов и их сжатые размеры
        # если на входе файл
        if len(files_and_folders) == 1 and files_and_folders[0][0] == input_path:
            with open(file_path, 'rb') as infile:
                raw_data = infile.read()
                compressed_data = compress_data(raw_data, compression)
                checksum = calculate_checksum(raw_data, protection)

                # записываем контрольную сумму (4 байта)
                f.write(struct.pack('I', checksum)) # CRC32 (если нет защиты, будет 0)

                # записываем размер сжатых данных (8 байт)
                f.write(struct.pack('Q', len(compressed_data))) # Размер сжатых данных

                # записываем сжатые данные
                f.write(compressed_data)
            # если на входе директория
        else:
            for file_path, file_type, _ in files_and_folders:
                if file_type == 0: # Если это файл
                    with open(os.path.join(input_path, file_path), 'rb') as infile:
                        raw_data = infile.read()
                        compressed_data = compress_data(raw_data, compression)
                        checksum = calculate_checksum(raw_data, protection)

                        # записываем контрольную сумму (4 байта)
                        f.write(struct.pack('I', checksum)) # CRC32 (если нет защиты, будет 0)

                        # записываем размер сжатых данных (8 байт)
                        f.write(struct.pack('Q', len(compressed_data))) # Размер сжатых данных

                        # записываем сжатые данные
                        f.write(compressed_data)
    print(f"{input_path} успешно заархивировано в {output_file}")

```

```

# декодер: восстанавливает файлы и папки из архива
def decode(input_archive, output_dir):
    with open(input_archive, 'rb') as f:
        # читаем и проверяем сигнатуру
        signature = f.read(6)
        if signature != SIGNATURE:
            print("Ошибка: неверная сигнатура файла!")
            return

        # читаем версию формата
        major_version = struct.unpack('B', f.read(1))[0]
        minor_version = struct.unpack('B', f.read(1))[0]
        print(f"Версия формата: {major_version}.{minor_version}")
        if major_version != FORMAT_VERSION[0] or minor_version != FORMAT_VERSION[1]:
            print("Ошибка: неверная версия архива!")
            return

        # читаем коды сжатия и защиты от помех
        compression = struct.unpack('B', f.read(1))[0]
        protection = struct.unpack('B', f.read(1))[0]
        print(f"Алгоритм сжатия: {compression}, Защита от помех: {protection}")

        # читаем количество файлов/папок
        num_files = struct.unpack('I', f.read(4))[0]
        print(f"Количество файлов/папок: {num_files}")

        file_info = []

        # Читаем метаданные для каждого файла/папки
        for _ in range(num_files):
            # читаем длину пути
            path_length = struct.unpack('H', f.read(2))[0]
            # читаем путь
            file_path = f.read(path_length).decode('utf-8')
            # читаем тип (1 байт)
            file_type = struct.unpack('B', f.read(1))[0]
            file_info.append((file_path, file_type))

        # восстанавливаем файлы и папки
        for file_path, file_type in file_info:
            output_path = os.path.join(output_dir, file_path)
            # это папка
            if file_type == 1:
                os.makedirs(output_path, exist_ok=True)
            # это файл
            else:
                os.makedirs(os.path.dirname(output_path), exist_ok=True)
                # читаем контрольную сумму
                checksum = struct.unpack('I', f.read(4))[0]
                # читаем размер сжатых данных
                compressed_size = struct.unpack('Q', f.read(8))[0]
                # читаем сжатые данные
                compressed_data = f.read(compressed_size)
                raw_data = decompress_data(compressed_data, compression)

                # проверка контрольной суммы
                if calculate_checksum(raw_data, protection) != checksum:
                    print(f"Ошибка: контрольная сумма не совпадает для файла {file_path}!")
                    return

                # записываем данные в файл
                with open(output_path, 'wb') as outfile:
                    outfile.write(raw_data)

        print(f"Архив {input_archive} успешно разархивирован в директорию {output_dir}")

encode('otik-labs.pdf', 'archived.team5', compression=NO_COMPRESSION, protection=NO_PROTECTION)
decode('archived.team5', 'otik-labs-dearchived')

```



Заархивируем файл:

Исходный файл:

otik-labs.pdf																	
00000000	00	01	02	03	04	05	06	07	08	09	0a	0b	0c	0d	0e	0f	
00000000	25	50	44	46	2d	31	2e	35	0a	25	d0	d4	c5	d8	0a	31	PDF-1.5.%PDFEW.1
00000010	20	30	20	6f	62	6a	0a	3c	0a	2f	4c	65	6e	67	74	0	obj.<<./Lengt
00000020	68	20	38	31	37	20	20	20	20	20	20	0a	2f	46	69	h	817 ./.Fi
00000030	6c	74	65	72	20	2f	46	6c	61	74	65	44	65	63	6f	64	lter /FlateDecod
00000040	65	0a	3e	3e	0a	73	74	72	65	61	6d	0a	78	da	6d	55	e.>>.stream.xbmU
00000050	4d	6f	e2	30	10	bd	e7	57	78	0f	95	da	03	c5	76	3e	Mov0.S3Wx.·b.Ev>
00000060	88	2b	84	64	27	20	f5	b0	db	aa	a0	d5	5e	69	62	ba	€+„d' x'ЫE X^ibe
00000070	91	4a	82	02	1c	fa	ef	d7	6f	c6	94	ed	6a	0f	a0	e7	'J,..ьпЧoЖ"нј. з
00000080	f1	f3	cc	9b	37	89	73	f3	ed	79	3d	b1	ed	f0	ea	27	cyM>7%syHy=±нрк'
00000090	e9	bd	14	2f	fe	38	9c	c7	c6	4f	aa	ef	db	43	72	73	йS./ю8н3X0EnbCrs
000000a0	53	0f	cd	79	ef	fb	d3	0f	ef	5b	df	5e	76	8f	0f	e2	S. HyпыУ. п[Я^vU.8
000000b0	79	1c	9a	b5	3f	89	db	ea	b1	7e	ec	bb	d3	5d	20	3f	у.лп7%Ык±-м>У] ?
000000c0	f6	cd	fb	b9	f5	17	d6	ff	49	ce	bf	75	fd	95	82	3a	цЫлх.ЦяIOiuэ+.,?
000000d0	e2	76	e3	7f	4d	36	da	4e	24	18	9b	ee	f4	1e	76	ae	евг.М6bNS.>оф.у®
000000e0	41	11	90	08	48	d0	f6	4f	3f	1e	bb	a1	7f	10	ea	5e	A.ђ.НРц07.>»Y..к^
000000f0	4a	19	02	cb	be	ad	86	3d	54	1e	93	69	ac	24	a6	97	J..лs.т=T."i-}\$ -
00000100	da	bb	ae	6f	c7	58	4e	bc	a2	78	a2	b4	68	bb	e6	14	b=®o3XNjyxъrh»ж.

Архив:

archived.team5																	
00000000	00	01	02	03	04	05	06	07	08	09	0a	0b	0c	0d	0e	0f	
00000000	52	41	52	4e	49	4b	01	01	00	00	01	00	00	00	0d	00	ARNIK.....
00000010	6f	74	69	6b	2d	6c	61	62	73	2e	70	64	66	00	00	00	otik-labs.pdf...
00000020	00	00	57	31	08	00	00	00	00	00	25	50	44	46	2d	31	..w1.....%PDF-1
00000030	2e	35	0a	25	d0	d4	c5	d8	0a	31	20	30	20	6f	62	6a	.5.%PDFEШ.1 0 obj
00000040	0a	3c	3c	0a	2f	4c	65	6e	67	74	68	20	38	31	37	20	<<./Length 817
00000050	20	20	20	20	20	20	0a	2f	46	69	6c	74	65	72	20	2f	./Filter /
00000060	46	6c	61	74	65	44	65	63	6f	64	65	0a	3e	3e	0a	73	FlateDecode.>>.s
00000070	74	72	65	61	6d	0a	78	da	6d	55	4d	6f	e2	30	10	bd	tream.xbmUMov0.S
00000080	e7	57	78	0f	95	da	03	c5	76	3e	88	2b	84	64	27	20	3Wx.·b.Ev>€+„d'
00000090	f5	b0	db	aa	a0	d5	5e	69	62	ba	91	4a	82	02	1c	fa	x'ЫE X^ibe'J,..b

archived.team5																	
00000000	00	01	02	03	04	05	06	07	08	09	0a	0b	0c	0d	0e	0f	
00000000	52	41	52	4e	49	4b	01	01	00	00	01	00	00	00	0d	00	ARNIK.....
00000010	6f	74	69	6b	2d	6c	61	62	73	2e	70	64	66	00	00	00	otik-labs.pdf...
00000020	00	00	57	31	08	00	00	00	00	00	25	50	44	46	2d	31	..w1.....%PDF-1
00000030	2e	35	0a	25	d0	d4	c5	d8	0a	31	20	30	20	6f	62	6a	.5.%PDFEШ.1 0 obj
00000040	0a	3c	3c	0a	2f	4c	65	6e	67	74	68	20	38	31	37	20	<<./Length 817
00000050	20	20	20	20	20	20	0a	2f	46	69	6c	74	65	72	20	2f	./Filter /
00000060	46	6c	61	74	65	44	65	63	6f	64	65	0a	3e	3e	0a	73	FlateDecode.>>.s
00000070	74	72	65	61	6d	0a	78	da	6d	55	4d	6f	e2	30	10	bd	tream.xbmUMov0.S
00000080	e7	57	78	0f	95	da	03	c5	76	3e	88	2b	84	64	27	20	3Wx.·b.Ev>€+„d'
00000090	f5	b0	db	aa	a0	d5	5e	69	62	ba	91	4a	82	02	1c	fa	x'ЫE X^ibe'J,..b

- сигнатура
- код защиты от помех
- Версия формата(Мажорная/Минорна)
- количество файлов и папок
- код сжатия

Разархивированный файл:

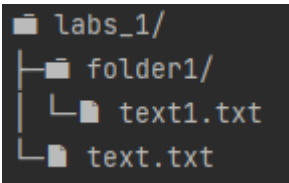
otik-labs.pdf																	
00000000	00	01	02	03	04	05	06	07	08	09	0a	0b	0c	0d	0e	0f	
00000000	25	50	44	46	2d	31	2e	35	0a	25	d0	d4	c5	d8	0a	31	%PDF-1.5.%PDFEШ.1
00000010	20	30	20	6f	62	6a	0a	3c	3c	0a	2f	4c	65	6e	67	74	0 obj.<<./Lengt
00000020	68	20	38	31	37	20	20	20	20	20	20	20	0a	2f	46	69	h 817 ./.Fi
00000030	6c	74	65	72	20	2f	46	6c	61	74	65	44	65	63	6f	64	lter /FlateDecod
00000040	65	0a	3e	3e	0a	73	74	72	65	61	6d	0a	78	da	6d	55	e.>>.stream.xbmU
00000050	4d	6f	e2	30	10	bd	e7	57	78	0f	95	da	03	c5	76	3e	Mov0.S3Wx.·b.Ev>
00000060	88	2b	84	64	27	20	f5	b0	db	aa	a0	d5	5e	69	62	ba	€+„d' x'ЫE X^ibe
00000070	91	4a	82	02	1c	fa	ef	d7	6f	c6	94	ed	6a	0f	a0	e7	'J,..ьпЧoЖ"нј. з
00000080	f1	f3	cc	9b	37	89	73	f3	ed	79	3d	b1	ed	f0	ea	27	cyM>7%syHy=±нрк'

Сравнение файлов:

```
D:\pyprj\lab3>FC otik-labs.pdf otik-labs-dearchived/otik-labs.pdf
Сравнение файлов otik-labs.pdf и OTIK-LABS-DEARCHIVED/OTIK-LABS.PDF
FC: различия не найдены
```



Заархивируем папку:



Архив:

archived.team5																	
000000ac	00	01	02	03	04	05	06	07	08	09	0a	0b	0c	0d	0e	0f	
00000000	52	41	52	4e	49	4b	01	01	00	00	03	00	00	00	07	00	RARNIK.....
00000010	66	6f	6c	64	65	72	31	01	08	00	74	65	78	74	2e	74	folder1...text.t
00000020	78	74	00	11	00	66	6f	6c	64	65	72	31	5c	74	65	78	xt...folder1\tex
00000030	74	31	2e	74	78	74	00	00	00	00	00	10	00	00	00	00	t1.txt.....
00000040	00	00	00	74	65	78	74	20	69	6e	20	74	65	78	74	2e	...text in text.
00000050	74	78	74	00	00	00	00	11	00	00	00	00	00	00	00	74	txt.....t
00000060	65	78	74	20	69	6e	20	74	65	78	74	31	2e	74	78	74	ext in text1.txt
00000070	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	

archived.team5																	
000000ac	00	01	02	03	04	05	06	07	08	09	0a	0b	0c	0d	0e	0f	
00000000	52	41	52	4e	49	4b	01	01	00	00	03	00	00	00	07	00	RARNIK.....
00000010	66	6f	6c	64	65	72	31	01	08	00	74	65	78	74	2e	74	folder1...text.t
00000020	78	74	00	11	00	66	6f	6c	64	65	72	31	5c	74	65	78	xt...folder1\tex
00000030	74	31	2e	74	78	74	00	00	00	00	00	10	00	00	00	00	t1.txt.....
00000040	00	00	00	74	65	78	74	20	69	6e	20	74	65	78	74	2e	...text in text.
00000050	74	78	74	00	00	00	00	11	00	00	00	00	00	00	00	74	txt.....t
00000060	65	78	74	20	69	6e	20	74	65	78	74	31	2e	74	78	74	ext in text1.txt
00000070	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	

- сигнатура
- Версия формата(Мажорная/Минорна)
- код сжатия
- код защиты от помех
- количество файлов и папок

archived.team5

×

000000ac	00	01	02	03	04	05	06	07	08	09	0a	0b	0c	0d	0e	0f	
00000000	52	41	52	4e	49	4b	01	01	00	00	03	00	00	00	07	00	RARNIK.....
00000010	66	6f	6c	64	65	72	31	01	08	00	74	65	78	74	2e	74	folder1...text.t
00000020	78	74	00	11	00	66	6f	6c	64	65	72	31	5c	74	65	78	xt...folder1\tex
00000030	74	31	2e	74	78	74	00	00	00	00	00	10	00	00	00	00	t1.txt.....
00000040	00	00	00	74	65	78	74	20	69	6e	20	74	65	78	74	2e	...text in text.
00000050	74	78	74	00	00	00	00	11	00	00	00	00	00	00	00	74	txt.....t
00000060	65	78	74	20	69	6e	20	74	65	78	74	31	2e	74	78	74	ext in text1.txt
00000070	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	

Защита от помех

Размер сжатых данных

Сжатые данные

сигнатура

код защиты от помех

Версия формата(Мажорная/Минорна)

количество файлов и папок

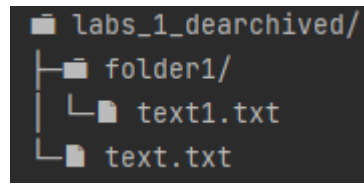
код сжатия

Путь файла/папки

Длина пути для каждого файла/папки

Тип: 00 - файл, 01 - папка

Разархивированная папка:

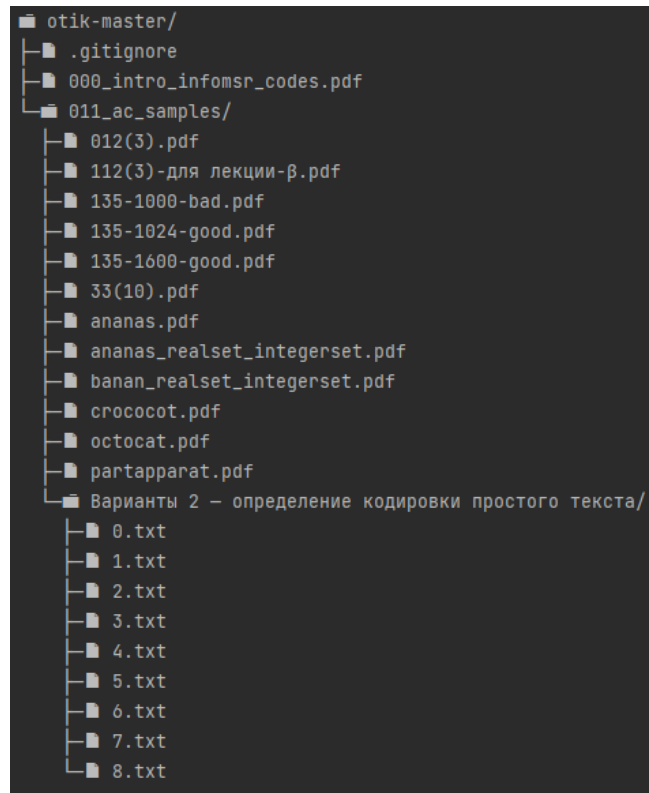


Сравнение файлов:

```
D:\pyprj\lab3>FC labs_1\folder1\text1.txt labs_1_dearchived\folder1\text1.txt
Сравнение файлов LABS_1\FOLDER1\text1.txt и LABS_1_DEARCHIVED\FOLDER1\TEXT1.TXT
FC: различия не найдены

D:\pyprj\lab3>FC labs_1\text.txt labs_1_dearchived\text.txt
Сравнение файлов LABS_1\text.txt и LABS_1_DEARCHIVED\TEXT.TXT
FC: различия не найдены
```

Заархивируем более сложную папку:



Архив:

archived.team5															
00000000	00	01	02	03	04	05	06	07	08	09	0a	0b	0c	0d	0e 0f
00000000	52	41	52	4e	49	4b	01	01	00	00	19	00	00	00	RARNIK.....
00000010	30	31	31	5f	61	63	5f	73	61	6d	70	6c	65	73	011_ac_samples..
00000020	00	2e	67	69	74	69	67	6e	6f	72	65	00	1b	00	..gitignore...00
00000030	30	5f	69	6e	74	72	6f	5f	69	6e	66	6f	6d	73	0_intro_infomsr_
00000040	63	6f	64	65	73	2e	70	64	66	00	6d	00	30	31	codes.pdf.m.011_
00000050	61	63	5f	73	61	6d	70	6c	65	73	5c	d0	92	d0	ac_samples\P'P'C
00000060	80	d0	b8	d0	b0	d0	bd	d1	82	d1	8b	20	32	20	бPèP'PSC,C< 2 бб
00000070	94	20	d0	be	d0	bf	d1	80	d0	b5	d0	b4	d0	b5	" PsPiCбPµPrPµP»
00000080	d0	b5	d0	bd	d0	b8	d0	b5	20	d0	ba	d0	be	d0	PµPSPèPµ PePsPrP
00000090	b8	d1	80	d0	be	d0	b2	d0	ba	d0	b8	20	d0	bf	ёCбPsPIPePè PiCб
000000a0	d0	be	d1	81	d1	82	d0	be	d0	b3	d0	be	20	d1	PsCfC,PsPiPs C,P
000000b0	b5	d0	ba	d1	81	d1	82	d0	b0	01	19	00	30	31	µPeCfC,P'...011_
000000c0	61	63	5f	73	61	6d	70	6c	65	73	5c	30	31	32	ac_samples\012(3
000000d0	29	2e	70	64	66	00	30	00	30	31	31	5f	61	63	).pdf.0.011_ac_s
000000e0	61	6d	70	6c	65	73	5c	31	31	32	28	33	29	2d	amples\112(3)-Pr
000000f0	d0	bb	d1	8f	20	d0	bb	d0	b5	d0	ba	d1	86	d0	P»Cµ P»PµPeC†PèP
00000100	b8	2d	ce	b2	2e	70	64	66	00	1f	00	30	31	31	è-OI.pdf...011_a
00000110	63	5f	73	61	6d	70	6c	65	73	5c	31	33	35	2d	c_samples\135-10
00000120	30	30	2d	62	61	64	2e	70	64	66	00	20	00	30	00-bad.pdf. .011
00000130	5f	61	63	5f	73	61	6d	70	6c	65	73	5c	31	33	_ac_samples\135-
00000140	31	30	32	34	2d	67	6f	6f	64	2e	70	64	66	00	1024-good.pdf. .
00000150	30	31	31	5f	61	63	5f	73	61	6d	70	6c	65	73	011_ac_samples\1
00000160	33	35	2d	31	36	30	30	2d	67	6f	6f	64	2e	70	35-1600-good.pdf
00000170	00	19	00	30	31	31	5f	61	63	5f	73	61	6d	70	...011_ac_sample
00000180	73	5c	33	33	28	31	30	29	2e	70	64	66	00	19	s\33(10).pdf...0
00000190	31	31	5f	61	63	5f	73	61	6d	70	6c	65	73	5c	11_ac_samples\an
000001a0	61	6e	61	73	2e	70	64	66	00	2c	00	30	31	31	anas.pdf. ,.011_a
000001b0	63	5f	73	61	6d	70	6c	65	73	5c	61	6e	61	6e	c_samples\ananas
000001c0	5f	72	65	61	6c	73	65	74	5f	69	6e	74	65	67	_realset_integer
000001d0	73	65	74	2e	70	64	66	00	2b	00	30	31	31	5f	set.pdf.+011_ac
000001e0	5f	73	61	6d	70	6c	65	73	5c	62	61	6e	61	6e	_samples\banan_r
000001f0	65	61	6c	73	65	74	5f	69	6e	74	65	67	65	72	ealset_integerse
00000200	74	2e	70	64	66	00	1b	00	30	31	31	5f	61	63	t.pdf...011_ac_s
00000210	61	6d	70	6c	65	73	5c	63	72	6f	63	6f	63	6f	amples\crococot.
00000220	70	64	66	00	1a	00	30	31	31	5f	61	63	5f	73	pdf...011_ac_sam
00000230	70	6c	65	73	5c	6f	63	74	6f	63	61	74	2e	70	ples\octocat.pdf
00000240	00	1e	00	30	31	31	5f	61	63	5f	73	61	6d	70	...011_ac_sample

00000000	00	01	02	03	04	05	06	07	08	09	0a	0b	0c	0d	0e	0f	
00000000	52	41	52	4e	49	4b	01	01	00	00	19	00	00	00	0e	00	BARNIK.....
00000010	30	31	31	5f	61	63	5f	73	61	6d	70	6c	65	73	01	0a	011_ac_samples..
00000020	00	2e	67	69	74	69	67	6e	6f	72	65	00	1b	00	30	30	..gitignore...00
00000030	30	5f	69	6e	74	72	6f	5f	69	6e	66	6f	6d	73	72	5f	0_intro_infomsr_
00000040	63	6f	64	65	73	2e	70	64	66	00	6d	00	30	31	31	5f	codes.pdf.m.011_
00000050	61	63	5f	73	61	6d	70	6c	65	73	5c	d0	92	d0	b0	d1	ac_samples\P'P'C
00000060	80	d0	b8	d0	b0	d0	bd	d1	82	d1	8b	20	32	20	e2	80	ЪРeP'PSC,C< 2 eЪ
00000070	94	20	d0	be	d0	bf	d1	80	d0	b5	d0	b4	d0	b5	d0	bb	" PsPіCЪPμPrPμP»
00000080	d0	b5	d0	bd	d0	b8	d0	b5	20	d0	ba	d0	be	d0	b4	d0	PμPSPePμ PеPSPrP
00000090	b8	d1	80	d0	be	d0	b2	d0	ba	d0	b8	20	d0	bf	d1	80	еCЪPSPіPеPе PіCЪ
000000a0	d0	be	d1	81	d1	82	d0	be	d0	b3	d0	be	20	d1	82	d0	PSCГC,PSPіPс C,P
000000b0	b5	d0	ba	d1	81	d1	82	d0	b0	01	19	00	30	31	31	5f	μPеCГC,P'...011_
000000c0	61	63	5f	73	61	6d	70	6c	65	73	5c	30	31	32	28	33	ac_samples\012(3
000000d0	29	2e	70	64	66	00	30	00	30	31	31	5f	61	63	5f	73	).pdf.0.011_ac_s
000000e0	61	6d	70	6c	65	73	5c	31	31	32	28	33	29	2d	d0	b4	amples\112(3)-Pr

	сигнатура		код защиты от помех
	Версия формата(Мажорная/Минорна)		количество файлов и папок
	код сжатия		

Разархивированная папка:

```

otik-master_dearchived/
├── .gitignore
├── 000_intro_infomsr_codes.pdf
├── 011_ac_samples/
│   ├── 012(3).pdf
│   ├── 112(3)-для лекции-β.pdf
│   ├── 135-1000-bad.pdf
│   ├── 135-1024-good.pdf
│   ├── 135-1600-good.pdf
│   ├── 33(10).pdf
│   ├── ananas.pdf
│   ├── ananas_realset_integerset.pdf
│   ├── banan_realset_integerset.pdf
│   ├── crococot.pdf
│   ├── octocat.pdf
│   ├── partappararat.pdf
│   └── Варианты 2 – определение кодировки простого текста/
│       ├── 0.txt
│       ├── 1.txt
│       ├── 2.txt
│       ├── 3.txt
│       ├── 4.txt
│       ├── 5.txt
│       ├── 6.txt
│       ├── 7.txt
│       └── 8.txt

```

## Сравнение произвольно взятых файлов:

```
D:\pyprj\lab3>FC otik-master\000_intro_infomsr_codes.pdf otik-master_dearchived\000_intro_infomsr_codes.pdf
Сравнение файлов OTIK-MASTER\000_intro_infomsr_codes.pdf и OTIK-MASTER_DEARCHIVED\000_INTRO_INFOMSR_CODES.PDF
FC: различия не найдены
```

```
D:\pyprj\lab3>FC "otik-master\011_ac_samples\Варианты 2 - определение кодировки простого текста\5.txt" "otik-master\011_ac_samples\Варианты 2 - определение кодировки простого текста\5.txt"
Сравнение файлов OTIK-MASTER\011_AC_SAMPLES\ВАРИАНТЫ 2 - ОПРЕДЕЛЕНИЕ КОДИРОВКИ ПРОСТОГО ТЕКСТА\5.txt и OTIK-MASTER\011_AC_SAMPLES\ВАРИАНТЫ 2 - ОПРЕДЕЛЕНИЕ КОДИРОВКИ ПРОСТОГО ТЕКСТА\5.TXT
FC: различия не найдены
```