

Л4.1. Упрощённое задание (не более 7 баллов за работу)

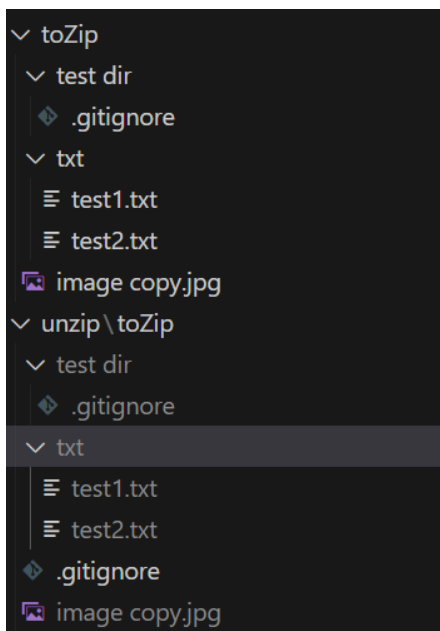
Разработайте кодер и декодер «наивного» RLE. Продемонстрируйте работу кодера и декодера.

Штраф —3 балла за текстовое представление архива или если кодер обрабатывает только файлы в текстовом представлении — см. раздел Л1.2.

Пример работы

Размеры файлов до и после кодирования

```
PS C:\Users\thetr\labs\otik> py .\xipA.py
encoding dir 'toZip'
+   file 'image copy.jpg' was encoded: 385894 -> 768350(in bytes)
encoding dir 'toZip\test dir'
+   file '.gitignore' was encoded: 73 -> 146(in bytes)
encoding dir 'toZip\txt'
+   file 'test1.txt' was encoded: 18477 -> 1580(in bytes)
+   file 'test2.txt' was encoded: 49 -> 98(in bytes)
object 'toZip' encoded
object 'toZip' decoded
```



Папка 'toZip' была архивирована в файл 'toZip.xip' и разархивирована в папку unzip

Содержание файла toZip.xip:

Полный дамп файла хранится в hex_dump.md:

	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	
00000000	11	11	01	00	02	00	00	00	24	C1	0B	00	00	00	00	00\$Á.....
00000010	35	3B	74	6F	5A	69	70	7B	31	34	3B	69	6D	61	67	65	5;toZip{14;image < ===== Начало файла image copy.jpg =====
00000020	20	63	6F	70	79	2E	6A	70	67	3B	37	36	38	33	35	30	copy.jpg;768350
00000030	3B	00	FF	00	D8	00	FF	00	E0	00	00	00	10	00	4A	00	;...0...à.....J.
00000040	46	00	49	00	46	00	00	02	01	00	00	00	60	00	00	00	F.I.F.....`...
00000050	60	01	00	00	FF	00	FE	00	00	00	3B	00	43	00	52	00	`.....p...;.C.R.
00000060	45	00	41	00	54	00	4F	00	52	00	3A	00	20	00	67	00	E.A.T.O.R... .g.
00000070	64	00	2D	00	6A	00	70	00	65	00	67	00	20	00	76	00	d...j.p.e.g. .v.
00000080	31	00	2E	00	30	00	20	00	28	00	75	00	73	00	69	00	1...0. .(.u.s.i.
00000090	6E	00	67	00	20	00	49	00	4A	00	47	00	20	00	4A	00	n.g. .I.J.G. .J.
000000A0	50	00	45	00	47	00	20	00	76	00	36	00	32	00	29	00	P.E.G. .v.6.2.).
000000B0	2C	00	20	00	71	00	75	00	61	00	6C	00	69	00	74	00	...g.u.a.l.i.t.

Формат данных: {Длина названия файла;Название файла;Длина файла;Файл}

Пример с повторами(test1.txt):

78 74 7B 39 3B 74 65 73	74 31 2E 74 78 74 3B 31	xt{9;test1.txt;1
35 38 30 3B 43 73 00 0D	00 0A 42 73 00 0D 00 0A	580;Cs....Bs....
43 73 00 0D 00 0A 43 73	00 0D 00 0A 43 73 00 0D	Cs....Cs....Cs..
00 0A 43 73 00 0D 00 0A	43 73 00 0D 00 0A 43 73	..Cs....Cs....Cs
00 0D 00 0A 43 73 00 0D	00 0A 43 73 00 0D 00 0ACs....Cs....
43 73 00 0D 00 0A 43 73	00 0D 00 0A 43 73 00 0D	Cs....Cs....Cs..
00 0A 43 73 00 0D 00 0A	43 73 00 0D 00 0A 43 73	..Cs....Cs....Cs
00 0D 00 0A 43 73 00 0D	00 0A 43 73 00 0D 00 0ACs....Cs....
43 73 00 0D 00 0A 43 73	00 0D 00 0A 43 73 00 0D	Cs....Cs....Cs..
00 0A 43 73 00 0D 00 0A	43 73 00 0D 00 0A 43 73	..Cs....Cs....Cs
00 0D 00 0A 43 73 00 0D	00 0A 43 73 00 0D 00 0ACs....Cs....
43 73 00 0D 00 0A 43 73	00 0D 00 0A 43 73 00 0D	Cs....Cs....Cs..
00 0A 43 73 00 0D 00 0A	43 73 00 0D 00 0A 43 73	..Cs....Cs....Cs
00 0D 00 0A 43 73 00 0D	00 0A 43 73 00 0D 00 0ACs....Cs....
43 73 00 0D 00 0A 43 73	00 0D 00 0A 43 73 00 0D	Cs....Cs....Cs..
00 0A 43 73 00 0D 00 0A	43 73 00 0D 00 0A 43 73	..Cs....Cs....Cs
00 0D 00 0A 43 73 00 0D	00 0A 43 73 00 0D 00 0ACs....Cs....
43 73 00 0D 00 0A 43 73	00 0D 00 0A 43 73 00 0D	Cs....Cs....Cs..
00 0A 43 73 00 0D 00 0A	43 73 00 0D 00 0A 43 73	..Cs....Cs....Cs
00 0D 00 0A 43 73 00 0D	00 0A 43 73 00 0D 00 0ACs....Cs....

Во второй строчке файла на 1 с меньше. Это видно в файле в архиве .xir:

43 73 (67 's') 00 0D (0 '\r') 00 0A (0 '\n') 42 73 Bs (66 's')...

Пример файла без повторений(test2.txt):

7C 39 3B 74 65 73 74 32	2E 74 78 74 3B 39 38 3B	9;test2.txt;98;
00 66 00 64 00 6B 00 76	00 6E 00 6B 00 64 00 66	.f.d.k.v.n.k.d.f
00 62 00 6A 00 66 00 67	00 6B 00 62 00 66 00 6B	.b.j.f.g.k.b.f.k
00 67 00 62 00 6B 00 D0	00 B0 00 D0 00 B2 00 D0	.g.b.k.⌌.⌌.⌌.⌌
00 BE 00 D0 00 BB 00 D0	00 BC 00 D1 00 82 00 D0	.⌌.⌌.⌌.⌌.⌌.⌌.⌌.⌌
00 B2 00 D0 00 B0 00 D0	00 BB 00 D0 00 BC 00 D0	.⌌.⌌.⌌.⌌.⌌.⌌.⌌.⌌
00 B8 00 D0 00 BE 00 D0	00 B2 00 D0 00 B0 00 D0	.⌌.⌌.⌌.⌌.⌌.⌌.⌌.⌌
00 BC 7D	+	.⌌.⌌.⌌.⌌.⌌.⌌.⌌.⌌

Видно, что символы идут без повторений, перед каждым символом 00.