

Группа К20

Вариант №1

Задача 3.1. Реализовать разбор двоичного формата данных (в духе формата WAD игры Doom или графического формата PNG). Данные начинаются с сигнатуры 0x51 0x4b 0x42, за которой следует структура А. Порядок байт: от младшего к старшему. Адреса указаны в виде смещений от начала данных. В решении разрешено использовать модуль struct.

Структура А:	1	Структура В
	2	int64

Структура В:	1	float
	2	Массив структур С, размер 6
	3	uint8
	4	uint64
	5	uint64
	6	Структура D

Структура С:	1	int16
	2	int8

Структура D:	1	Структура E
	2	uint64

Структура E:	1	Размер (uint32) и адрес (uint16) массива int8
	2	float
	3	int32
	4	Размер (uint16) и адрес (uint32) массива uint16
	5	Массив float, размер 5

Примеры разбора двоичного формата с помощью функции f31:

1. Двоичные данные:

```
(b'QKBq!\xa3\xbeg\xfeP\x16g\x92\xa3\x86Y\xb9\x9a\xd8\x180\x1eq\x98\x1d\xa2_\xd9'
b'z:x\xf0\x96u\xcd\t\xbe|\xbfg\xa6/\x02\x00\x00\x00b\x00\xe4 7?+\xc6\xb6\x92'
b'\x04\x00d\x00\x00\x00\xd2(\x17>\xb7\xcf\x02?\xd5\x9e\xe7=3\x8c\x0f?)\xf2'
b'\xce=\x12E\xc3\xa4\xf1*\tde\xc3\x88\xa7E\xfc\x08\xe7\x988wq:\x9a\x98K\xca|')
```

Результат разбора:

```
{ 'A1': { 'B1': -0.31861451268196106,
          'B2': [{ 'C1': -409, 'C2': 80},
                  { 'C1': 26390, 'C2': -110},
                  { 'C1': -31069, 'C2': 89},
```

```

        {'C1': -25927, 'C2': -40},
        {'C1': 12312, 'C2': 30},
        {'C1': -26511, 'C2': 29}],
'B3': 162,
'B4': 8473224148112038239,
'B5': 3433510653673736653,
'B6': {'D1': {'E1': [-104, 56],
               'E2': 0.7153456211090088,
               'E3': -1833515477,
               'E4': [29047, 39482, 19352, 31946],
               'E5': [0.1476166546344757,
                      0.5109819769859314,
                      0.11309591680765152,
                      0.5607330203056335,
                      0.1010478213429451]}},
'D2': 7208339895922935058}},
'A2': -1798910675040812187}

```

2. Двоичные данные:

```

(b'QKB9\xcc\x0f>u\xc4?\xe2_D7#\xee4{\x8e\xd6\x12sz\x8d\x8d\xc7\xbc\xb5'
 b'p\x9f+\xf6\xde+\xe8\xc8\xe1\x13\x916\xfc\x89\x05\x00\x00\x00b\x00'
 b'\xea\xeb3\xbfP\xac1\xb1\x03\x00g\x00\x00\x00Wo\t?N\x03F>zTZ\xbe\xca\x17'
 b's\xbf0L?\x86\xc3\xfb\x17\xf8\xd0\xc6\xb9\x16\xbd\x97D\x02\x10a\xd6\x14'
 b'qA\x167i\x96\xac\xd2l')

```

Результат разбора:

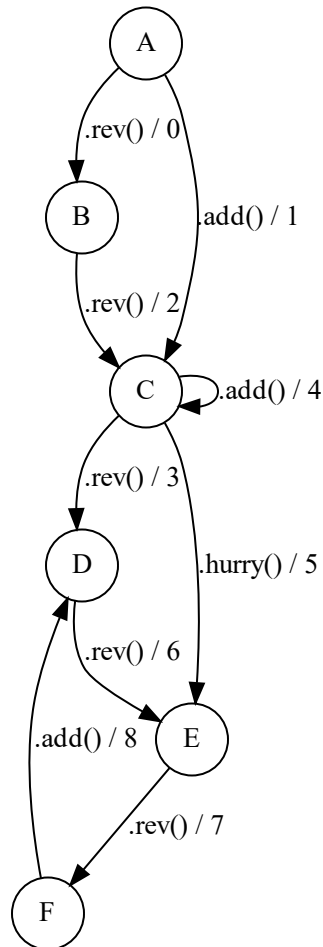
```

{'A1': {'B1': 0.4703080952167511,
        'B2': [{'C1': -15243, 'C2': 63},
                 {'C1': 24546, 'C2': 68},
                 {'C1': 9015, 'C2': -18},
                 {'C1': 31540, 'C2': -114},
                 {'C1': 4822, 'C2': 115},
                 {'C1': -29318, 'C2': -115}],
        'B3': 199,
        'B4': 3161234655679657404,
        'B5': 9942882074058934504,
        'B6': {'D1': {'E1': [-42, 20, 113, 65, 22],
                       'E2': -0.7028185129165649,
                       'E3': -1322144688,
                       'E4': [26935, 44182, 27858],
                       'E5': [0.536855161190033,
                              0.19337198138237,
                              -0.21321287751197815,
                              -0.9495817422866821,

```

```
0.7980863451957703]],  
'D2': 8702873093912904582}},  
'A2': 6994092714933163705}
```

Задача 3.2. Реализовать конечный автомат Мили в виде класса C32. Начальным состоянием автомата является A. Методы возвращают числовые значения. Если вызываемый метод не реализован для некоторого состояния, необходимо вызвать исключение RuntimeError.



1. Пример использования класса C32:

<code>o = C32()</code>	
<code>o.rev()</code>	0
<code>o.add()</code>	RuntimeError
<code>o.rev()</code>	2
<code>o.add()</code>	4
<code>o.rev()</code>	3
<code>o.rev()</code>	6
<code>o.rev()</code>	7
<code>o.add()</code>	8
<code>o.rev()</code>	6
<code>o.rev()</code>	7
<code>o.add()</code>	8

2. Пример использования класса C32:

<code>o = C32()</code>	
<code>o.rev()</code>	0
<code>o.rev()</code>	2
<code>o.add()</code>	4
<code>o.rev()</code>	3
<code>o.rev()</code>	6
<code>o.rev()</code>	7
<code>o.add()</code>	8
<code>o.rev()</code>	6
<code>o.rev()</code>	7
<code>o.add()</code>	8
<code>o.rev()</code>	6
<code>o.rev()</code>	7

Вариант №2

Задача 3.1. Реализовать разбор двоичного формата данных (в духе формата WAD игры Doom или графического формата PNG). Данные начинаются с сигнатуры 0x4f 0x5a 0x4d, за которой следует структура A. Порядок байт: от младшего к старшему. Адреса указаны в виде смещений от начала данных. В решении разрешено использовать модуль struct.

Структура A:	1	int8
	2	float
	3	Структура B
	4	uint8
	5	Адрес (uint32) структуры D
	6	uint64

Структура B:	1	Размер (uint32) и адрес (uint32) массива адресов (uint16) структур C
	2	uint8
	3	uint16

Структура C:	1	int16
	2	double
	3	int64
	4	uint16
	5	int64
	6	Массив uint64, размер 2
	7	Массив int16, размер 6

Структура D:	1	int16
	2	Массив uint32, размер 4

Примеры разбора двоичного формата с помощью функции f31:

1. Двоичные данные:

```
(b'0ZM\x8d\xb5\xee\x01\xbf\x02\x00\x00\x00\x90\x00\x00\x00\xdc\x15\x14k'
b'\x94\x00\x00\x00\xe6.\x98 \xae\xaf\t\x81\x8cS\x80\x1e\xc5)\x160'
b"\xe7\xbf\xa2\xe4\xeb5\xd4\xe2Q\xe4'\x91\n\xbf\xf2\xfc\x8b\xf4\xcc\xf7"
b'H\xa2J\xe2\xa5\xc8z\xbb4\x08R\xcc\x18\x963a\x9b\xcev_\x02\xca\xd2\xc0'
b'\xa5\xe3\xe8\x05\xc5B@u\xe7\xeb\xdc\xb8\xe8?\x0c\xe8\xa6)\x15\x97H\x91\x80N'
b'\xb9\xc05\xc2\xea\x87\t\xc9\x0b\xa8\x0b\xbaq\xf7\xf1f{\xf5\xe6\xa1j/\xc4B'
b'\x86\xa2\x0e\x1d\xc0\xa5\x14\xd29\x8dDk \x00X\x00,U7\xcd\x89"\rc'
b'\x9c\x8b\xdc\xa9\xb8\xdfZ\x01\xbek')
```

Результат разбора:

```
{'A1': -115,
'A2': -0.5075486302375793,
```

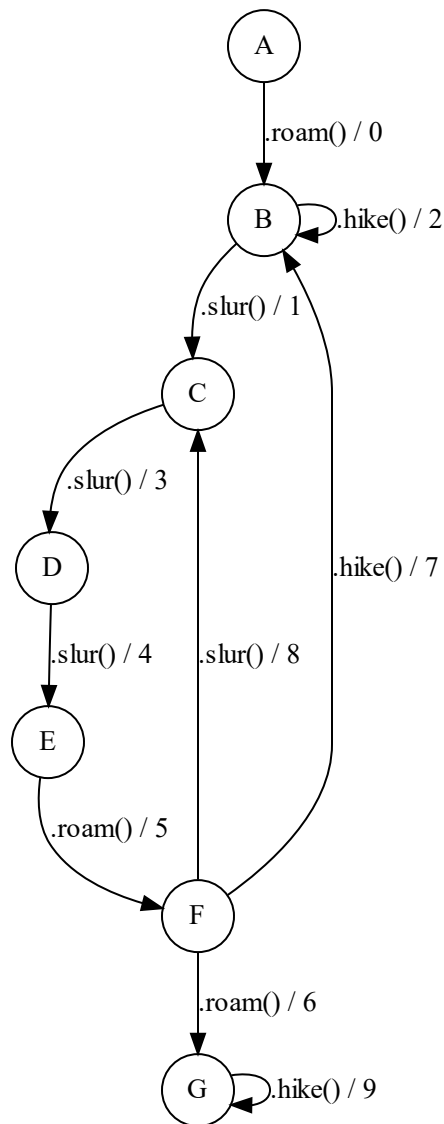


```

'C5': -4179243348295844661,
'C6': [12650642359273389969, 16281969506547888637],
'C7': [24329, -20233, 30707, -25732, 21551, -31194]}},
'B2': 176,
'B3': 49986},
'A4': 25,
'A5': {'D1': 2108, 'D2': [3855498968, 3971273585, 3842919187, 1647636699]},
'A6': 5382975413623921711}

```

Задача 3.2. Реализовать конечный автомат Мили в виде класса C32. Начальным состоянием автомата является A. Методы возвращают числовые значения. Если вызываемый метод не реализован для некоторого состояния, необходимо вызвать исключение RuntimeError.



1. Пример использования класса C32:

o = C32()	
o.roam()	0
o.hike()	2
o.slur()	1
o.slur()	3
o.slur()	4
o.hike()	RuntimeError
o.roam()	5
o.hike()	7
o.roam()	RuntimeError
o.slur()	1
o.slur()	3
o.slur()	4
o.hike()	RuntimeError
o.roam()	5
o.roam()	6
o.hike()	9
o.roam()	RuntimeError
o.hike()	9

2. Пример использования класса C32:

o = C32()	
o.roam()	0
o.hike()	2
o.slur()	1
o.slur()	3
o.slur()	4
o.roam()	5
o.slur()	8
o.slur()	3
o.roam()	RuntimeError
o.slur()	4
o.roam()	5
o.roam()	6
o.hike()	9
o.hike()	9

Вариант №3

Задача 3.1. Реализовать разбор двоичного формата данных (в духе формата WAD игры Doom или графического формата PNG). Данные начинаются с сигнатуры 0x42 0x52 0x56 0xe, за которой следует структура A. Порядок байт: от младшего к старшему. Адреса указаны в виде смещений от начала данных. В решении разрешено использовать модуль struct.

Структура A:	1	int8
	2	Адрес (uint16) структуры B
	3	int32
	4	uint64
	5	Размер (uint16) и адрес (uint16) массива адресов (uint16) структур C
	6	Адрес (uint16) структуры D
	7	uint16
	8	int64

Структура B:	1	Массив uint64, размер 2
	2	float
	3	Массив char, размер 6

Структура C:	1	uint8
	2	uint8
	3	float

Структура D:	1	Размер (uint32) и адрес (uint16) массива double
	2	int16
	3	Размер (uint32) и адрес (uint16) массива int32
	4	int8
	5	uint64

Примеры разбора двоичного формата с помощью функции f31:

1. Двоичные данные:

```
(b'BRV\x0e\r#\x00\xf0k\x8a\xf9t#\x17A7\xc3\xbc\xf5\x02\x00I\x00y\x00\xf6aN'
b'\x9c1\x15\xcdF\x80\xe2 \x08\xda\xd6\xf67\xc0\xa5Xn\x85\xf6\xd1\x80\xcc\xc6y'
b'SK\xbfqgsjyf\x7f\x14@C\xdb=\xe9\x8f\x1c\xd3\x80>=\x00C\x00\xc4\x9d\xf2'
b'\x11*5\xe0\xbf*z\xd5\xfcE~\xef\xbf\xea\xbe\xef\xd5\x9diB\xacc^j\xe9>\xb4\x17'
b'\x85~\x9b\r\xeb\x17fA\xe8\xe5\xc1e\xc8\x02\x00\x00\x00M\x00>\xe6\x07\x00\x00'
b'\x00J\x00\x86y\x19yC+\xf7\x0fN')
```

Результат разбора:

```
{'A1': 13,
 'A2': {'B1': [11943607745092651040, 14324966154031165016],
        'B2': -0.7942424416542053,
```

```

'B3': 'gqsjyf'},
'A3': -108368912,
'A4': 17707242476996600692,
'A5': [{'C1': 127, 'C2': 20, 'C3': 0.10706186294555664},
        {'C1': 233, 'C2': 143, 'C3': 0.2516106367111206}],
'A6': {'D1': [-0.5064897871533778, -0.9841642320787447],
        'D2': -6594,
        'D3': [-705708310,
                -1404933731,
                -378904989,
                -2062044098,
                -351429762,
                -398367209,
                -932855323],
        'D4': -122,
        'D5': 5624986224796703097},
'A7': 25078,
'A8': -2125621177477194674}

```

2. Двоичные данные:

```

(b'BRV\x0e\xe1#\x00V\xca\x81\xb7\x08\xa9=\xe0\xed\xb4\x0c\xe8\x03\x000\x00m'
b'\x00\xe3\xa9\xc6YKH\x89W\xd8\xb9ENS\xadF\x9b\x996GT\xa1`\xc8\xec\xf3\x94\xfe'
b'\xb3^?gafoxn\x1aX\xa5\x02\x8a>\xbasG\xb1\xab>\xa7\xc6\x07#\x08?=\x00C\x00I'
b'\x00\xd0\x83\xa8\xa2\x1b\xd0\xbf8I\x9e{\xf0\xe1\xc3\xbf\xRDs\xa2\x88w'
b'\x1b\x02\x00\x00\x00U\x00+\xce\x02\x00\x00\x00e\x00\x1e,=<\x89'
b'\xa8\xc5\x05\x98')

```

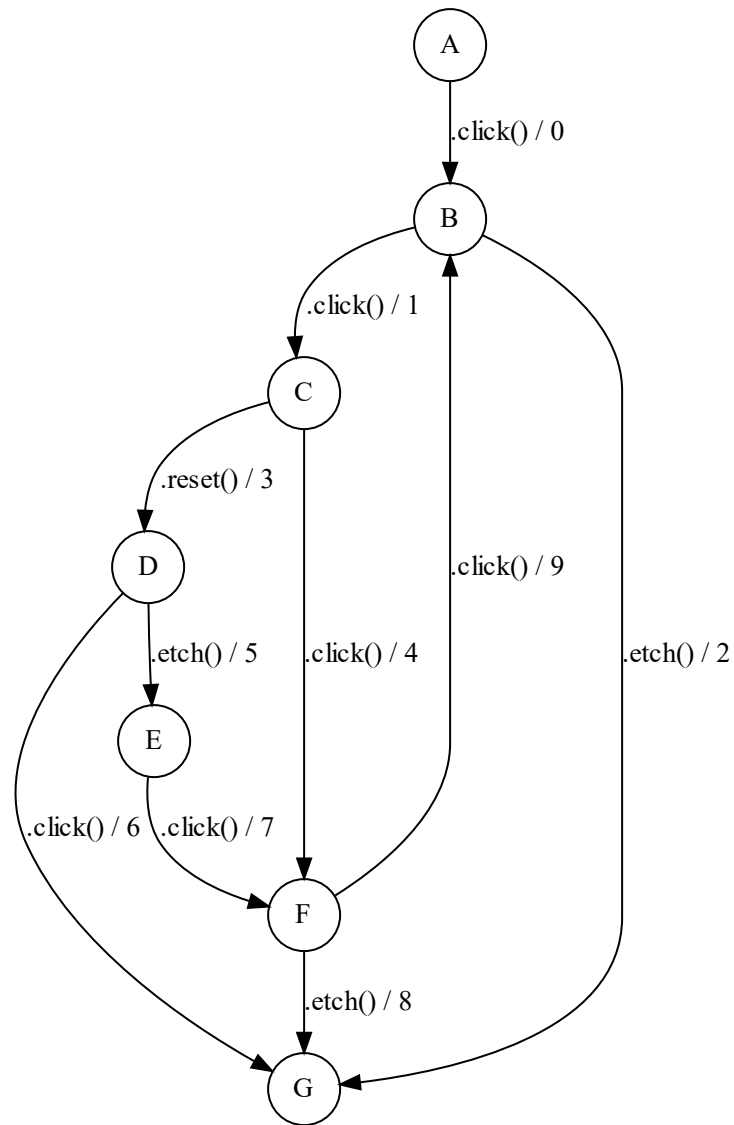
Результат разбора:

```

{'A1': -31,
'A2': {'B1': [3934346477342772805, 10733182682312823879],
        'B2': 0.8699339628219604,
        'B3': 'gafoxn'},
'A3': -1216230826,
'A4': 16720938450282195208,
'A5': [{'C1': 26, 'C2': 88, 'C3': 0.26955142617225647},
        {'C1': 186, 'C2': 115, 'C3': 0.3353368937969208},
        {'C1': 167, 'C2': 198, 'C3': 0.5317844748497009}],
'A6': {'D1': [-0.25168672997029473, -0.1553326228718761],
        'D2': -12757,
        'D3': [1933857400, 460818594],
        'D4': 30,
        'D5': 10954378996296203564},
'A7': 43491,
'A8': -5055194334588347962}

```

Задача 3.2. Реализовать конечный автомат Мили в виде класса C32. Начальным состоянием автомата является A. Методы возвращают числовые значения. Если вызываемый метод не реализован для некоторого состояния, необходимо вызвать исключение RuntimeError.



1. Пример использования класса C32:

<code>o = C32()</code>	
<code>o.reset()</code>	<code>RuntimeError</code>
<code>o.click()</code>	<code>0</code>
<code>o.reset()</code>	<code>RuntimeError</code>
<code>o.click()</code>	<code>1</code>
<code>o.reset()</code>	<code>3</code>
<code>o.reset()</code>	<code>RuntimeError</code>
<code>o.etch()</code>	<code>5</code>
<code>o.click()</code>	<code>7</code>
<code>o.click()</code>	<code>9</code>
<code>o.click()</code>	<code>1</code>
<code>o.click()</code>	<code>4</code>
<code>o.etch()</code>	<code>8</code>

2. Пример использования класса C32:

<code>o = C32()</code>	
<code>o.click()</code>	<code>0</code>
<code>o.click()</code>	<code>1</code>
<code>o.reset()</code>	<code>3</code>
<code>o.etch()</code>	<code>5</code>
<code>o.click()</code>	<code>7</code>
<code>o.click()</code>	<code>9</code>
<code>o.click()</code>	<code>1</code>
<code>o.click()</code>	<code>4</code>
<code>o.click()</code>	<code>9</code>
<code>o.etch()</code>	<code>2</code>

Вариант №4

Задача 3.1. Реализовать разбор двоичного формата данных (в духе формата WAD игры Doom или графического формата PNG). Данные начинаются с сигнатуры 0x4e 0x55 0x54 0x4e, за которой следует структура А. Порядок байт: от младшего к старшему. Адреса указаны в виде смещений от начала данных. В решении разрешено использовать модуль struct.

Структура А:	1	uint64
	2	Размер (uint32) и адрес (uint16) массива char
	3	Массив адресов (uint16) структур В, размер 2
	4	int64

Структура В:	1	Адрес (uint32) структуры С
	2	Структура D

Структура С:	1	double
	2	int16
	3	Массив uint16, размер 2
	4	Массив int32, размер 8
	5	int16
	6	int32
	7	uint8
	8	uint8

Структура D:	1	uint64
	2	int32

Примеры разбора двоичного формата с помощью функции f31:

1. Двоичные данные:

```
(b'NUTNqd\x01\xbbJ\xf3pZ\x03\x00\x00\x00\x1e\x00W\x00\x9d\x00\x06\x07'
b'\x02\x04\xb0\x8e\x7fumcf\xeaYv\x99\xfdI\xe0\xbf0\xaa\x0b\xff\xad\x12\x11'
b'\xf9\xfe\x90\x15\xedc\xaa\x9e\x92\xd8\xa9\x02\x16-\xbf\x0c\x88\x1aA\xf9'
b'HP\x8b\x91\xc0H\xc0\x1d\x7fH\xb0\xa9\xa5%\x86\xca2\x9cZ!\x00\x00\x00m'
b'\xd6\x06\x8d8\xa7\xdc\xf9\x87\xb8\xe9\x1b\x88Y\x19\xf6K\xe6\xc5?\x08'
b'u\xdf\xe5\xc8\x14\xd3(\xc6W\x90>\xa4\xb4\xces_\xda\xad\xbd\xbb\xdb\xf93\xc5'
b'\xbc\x9b\x18\xa2V\xbc\xbd\x13E\xb8CE\xb4\xbf\xee~\x07\x1d\xae\x8epg\x00\x00'
b'\x00\x1f\xbb0\xb8s\xa12\xcd0\x90\xd1\x0c')
```

Результат разбора:

```
{ 'A1': 6516976163095667825,
  'A2': 'mcf',
  'A3': [ { 'B1': { 'C1': -0.5090320584593517,
                  'C2': -21937,
```

```

'C3': [65291, 4781],
'C4': [-1862338287,
-1436291819,
-1445424482,
-1087564286,
1092257804,
-1957672711,
-1068973935,
-1337426147],
'C5': -23127,
'C6': 852133413,
'C7': 156,
'C8': 90},
'B2': {'D1': 18004449271646443117, 'D2': 468301959}},
{'B1': {'C1': 0.17109059826425388,
'C2': 29960,
'C3': [58847, 5320],
'C4': [1472604371,
-1264304496,
-631278642,
-608453203,
-1127926791,
1453463707,
1158921660,
-1270529096],
'C5': -4417,
'C6': -1373829250,
'C7': 142,
'C8': 112},
'B2': {'D1': 14786058044994272031, 'D2': 215060591}}}],
'A4': 8466642711112451846}

```

2. Двоичные данные:

```

(b'NUTN9\x935\x05e\xed|c\x02\x00\x00\x00\x1e\x00V\x00\x9c\x00?,\xdb#\x14( '
b'\xa1\x90tj\xac6\x19\x86\x07\xe7\xd1?\x7f\x9e,\xbePn\xbc\x7f%\xac\x9c\xe4'
b'\x88\xb3`\xbe~\xd3\xefY;\x92\xef\xa3v{\xd0\xb9\x883\x15\x93\xb3\x89\xef$'
b'\xdb\xd4\x95t\x92\x17I\xa7\xadT \x00\x00\x00\xe0\xa1\x8e\xda ju'
b'\xb6\xf0\x00\x80\xb5 @\xe2\x1e:\xe5\xd2\xd3?q\x13R\xab\x91f\xb9o\x1e4'
b'\xfde\x04K\xe5\xc7\xcb\xe9\xa3t\x92\xfaJ\xc2\xc7b\x92]a:\xd3\t\x9e`05\x92j'
b'\x88@\xee\xc1\xd3,\xf2;f\x00\x00\x00\xb2f\x08\xfa\xc9\xf4\x0f0@)4\x16')

```

Результат разбора:

```

{'A1': 7168865725001863993,
'A2': 'tj',

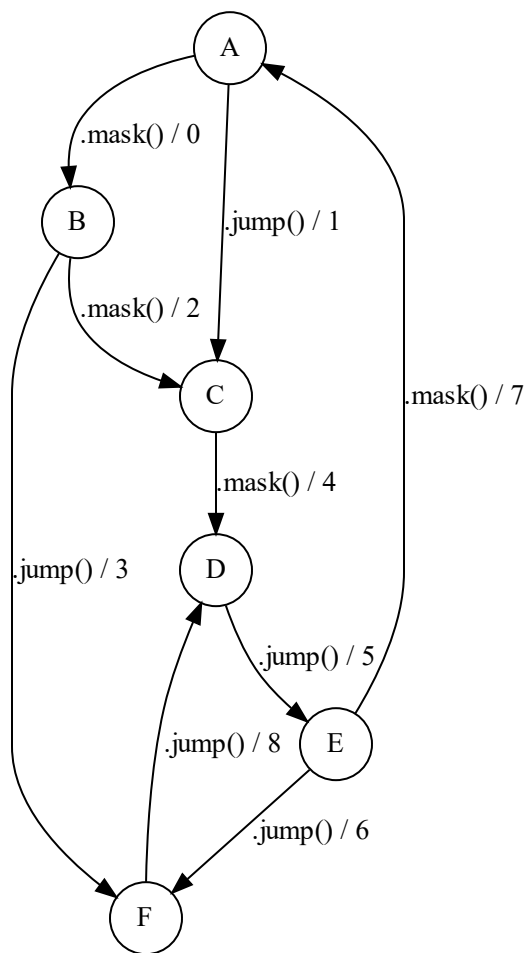
```

```

'A3': [{ 'B1': { 'C1': 0.2797259149127551,
                'C2': -24961,
                'C3': [48684, 28240],
                'C4': [-1406828612,
                      -1282874212,
                      -746668448,
                      -1841604113,
                      2071372783,
                      864598480,
                      -1984720107,
                      -723835665],
                'C5': 29845,
                'C6': -1488382062,
                'C7': 173,
                'C8': 84},
        'B2': { 'D1': 17345180116657807840, 'D2': 548765696}},
  { 'B1': { 'C1': 0.30974703479617816,
            'C2': 4977,
            'C3': [43858, 26257],
            'C4': [874409913,
                  1258579453,
                  -372520987,
                  -91065181,
                  1657258570,
                  979459474,
                  1620969939,
                  1787966768],
            'C5': 16520,
            'C6': 752075246,
            'C7': 242,
            'C8': 59},
    'B2': { 'D1': 3463255786791659186, 'D2': 372517184}}}],
'A4': -8025088994031489985}

```

Задача 3.2. Реализовать конечный автомат Мили в виде класса C32. Начальным состоянием автомата является А. Методы возвращают числовые значения. Если вызываемый метод не реализован для некоторого состояния, необходимо вызвать исключение `RuntimeError`.



1. Пример использования класса C32:

```

o = C32()
o.mask()           0
o.jump()           3
o.jump()           8
o.mask()           RuntimeError
o.jump()           5
o.mask()           7
o.jump()           1
o.mask()           4
o.jump()           5
o.jump()           6
o.jump()           8
o.jump()           5
o.mask()           7

```

2. Пример использования класса C32:

<code>o = C32()</code>	
<code>o.mask()</code>	0
<code>o.mask()</code>	2
<code>o.mask()</code>	4
<code>o.mask()</code>	RuntimeError
<code>o.jump()</code>	5
<code>o.jump()</code>	6
<code>o.jump()</code>	8
<code>o.jump()</code>	5
<code>o.jump()</code>	6
<code>o.mask()</code>	RuntimeError
<code>o.jump()</code>	8
<code>o.jump()</code>	5
<code>o.mask()</code>	7
<code>o.jump()</code>	1

Вариант №5

Задача 3.1. Реализовать разбор двоичного формата данных (в духе формата WAD игры Doom или графического формата PNG). Данные начинаются с сигнатуры 0x43 0x55 0x4f 0x51, за которой следует структура A. Порядок байт: от младшего к старшему. Адреса указаны в виде смещений от начала данных. В решении разрешено использовать модуль struct.

Структура A:	1	Адрес (uint16) структуры B
	2	uint16
	3	Структура C
	4	Массив float, размер 6
	5	int16

Структура B:	1	uint64
	2	uint64

Структура C:	1	Размер (uint16) и адрес (uint16) массива char
	2	int8
	3	uint64
	4	Размер (uint32) и адрес (uint32) массива адресов (uint16) структур D
	5	Размер (uint16) и адрес (uint32) массива uint32

Структура D:	1	uint16
	2	int64
	3	uint32
	4	uint64
	5	uint8
	6	int8

Примеры разбора двоичного формата с помощью функции f31:

1. Двоичные данные:

```
(b'CUOQ=\x006\x8f\x02\x00M\x00k\xce\xde(\xc82LE\xd8\x02\x00\x00\x00\x7f\x00\x00'
b'\x00\x06\x00\x83\x00\x00\x00\xa9b8?\xba\x9a/?\xf9\xe9\x95>\xf6\x86c\xbe4'
b'\xa1T\xbf\x98\xd1\x01?E\xcf|/\x053\x9e\xa5\xbe\xbf\xb1t\xb3Y\xec\xd6\xbd'
b'\x11qp\x91 \xacw\x8cJ%\x03\x0c\x8f\x0cE-N\xd48\xbfG\xc6Et.IoG\xfeHd\x1c'
b'\xa7\xf3z\x1e\t\x19\xdc\xb4\xa94J\xd0\x93\x15\xf5%Lk`0\x00g\x00\xf1'
b'\xd8\xba0\xb0W\x1a\x95\x1e\x896\xdcc\x6a6\xfd%\x1e\xea j1\x86\x08/')
```

Результат разбора:

```
{ 'A1': { 'B1': 13816662805698654076, 'B2': 1278414179848647857 },
  'A2': 36662,
  'A3': { 'C1': 'qp',
          'C2': 107,
```

```
'C3': 15583945866575666894,
'C4': [{ 'D1': 8337,
          'D2': -8139126967547168852,
          'D3': 1316898060,
          'D4': 3347377140952676564,
          'D5': 73,
          'D6': 111},
        { 'D1': 65095,
          'D2': 657097782541968456,
          'D3': 2847202329,
          'D4': 5487061194043836980,
          'D5': 107,
          'D6': 96}],
'C5': [1874516209,
        2501531568,
        3694561566,
        4255540067,
        1793728037,
        789087793]},
'A4': [0.720255434513092,
        0.6859546899795532,
        0.2928006947040558,
        -0.22219452261924744,
        -0.8305847644805908,
        0.5071043968200684],
'A5': -12475}
```

2. Двоичные данные:

```
(b'CUOQ=\x00[z\x02\x00M\x00\xce\nty\xe7a\x9c\xd4\xaaF\x02\x00\x00'
b'\x00\x7f\x00\x00\x00\x03\x00\x83\x00\x00\x00\xea\xc3\x9e>\xdaSa\xb9{'
b'\x98\x9f;\x93(^?\xffx6>g\x1d-\xbfM\x0f\xaa\xb5\x933\xcd\xe4\x81\xbc\xcf\\v'
b'9@\x05?\x87erLoI8\xe2\x00}\xe1\xe50\xcb|\x8d\x17=\xd4\x11-MzcV[\xd0D$;[\xfb'
b'\xb9\xdd\x19RK\\ \x13\xcd\x1p\xa6)\x1b\t\xefM@\x99\xc10\x00g\x00\x03\xfb<v|'
b'G\x1b\x16U\x8c\t\t')
```

Результат разбора:

```
{
  'A1': {
    'B1': 13583389521110873514,
    'B2': 9745513892053146831,
    'A2': 31323,
    'A3': {
      'C1': 'er',
      'C2': -50,
      'C3': 5092116096818313482,
      'C4': [
        'D1': 28492,
        'D2': 3523470210489268297,
        'D3': 395148491,

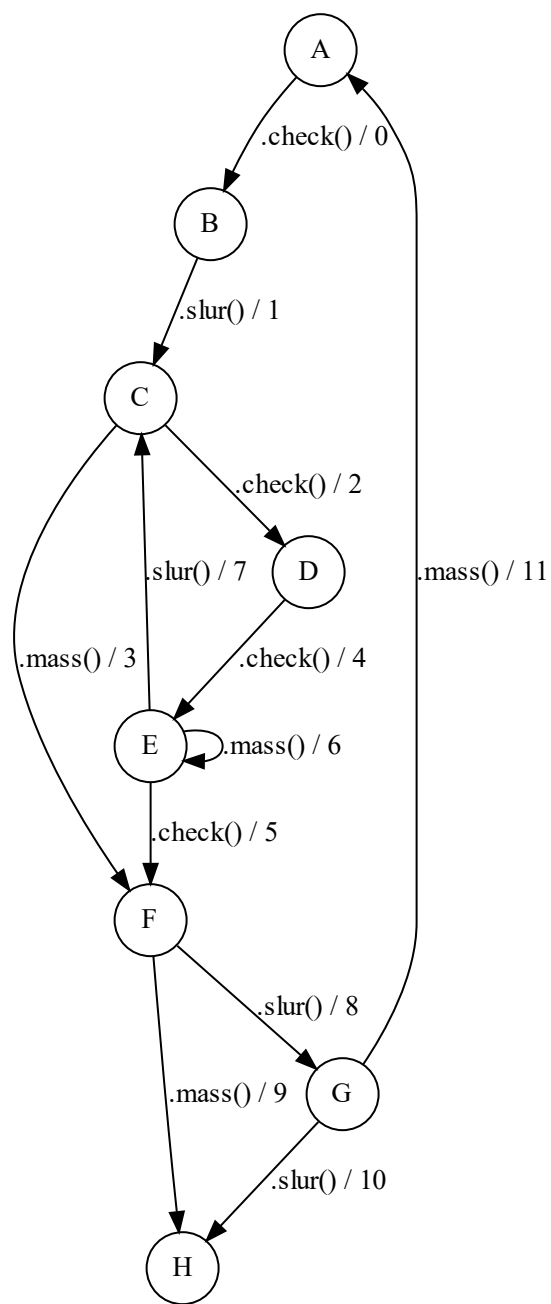
```

```

        'D4': 6224953581843371069,
        'D5': 91,
        'D6': -48},
    {'D1': 9284,
     'D2': 5427428941033593659,
     'D3': 3519877980,
     'D4': 4633622414023566960,
     'D5': 153,
     'D6': -63}],
    'C5': [1983705347, 370886524, 151620693]}},
    'A4': [0.31008845567703247,
           -0.00021488909260369837,
           0.004870472010225058,
           0.867806613445282,
           0.17819593846797943,
           -0.6762298941612244],
    'A5': 3917}

```

Задача 3.2. Реализовать конечный автомат Мили в виде класса C32. Начальным состоянием автомата является А. Методы возвращают числовые значения. Если вызываемый метод не реализован для некоторого состояния, необходимо вызвать исключение RuntimeError.



1. Пример использования класса C32:

o = C32()	
o.check()	0
o.slur()	1
o.check()	2
o.check()	4
o.mass()	6
o.check()	5
o.slur()	8
o.mass()	11
o.check()	0
o.slur()	1
o.check()	2
o.check()	4
o.slur()	7
o.mass()	3
o.slur()	8
o.slur()	10

2. Пример использования класса C32:

o = C32()	
o.check()	0
o.mass()	RuntimeError
o.slur()	1
o.mass()	3
o.slur()	8
o.mass()	11
o.mass()	RuntimeError
o.check()	0
o.slur()	1
o.check()	2
o.check()	4
o.mass()	6
o.check()	5
o.mass()	9

Вариант №6

Задача 3.1. Реализовать разбор двоичного формата данных (в духе формата WAD игры Doom или графического формата PNG). Данные начинаются с сигнатуры 0x46 0x52 0x4c 0x47 0xaa, за которой следует структура A. Порядок байт: от старшего к младшему. Адреса указаны в виде смещений от начала данных. В решении разрешено использовать модуль struct.

Структура A:	1	Размер (uint16) и адрес (uint16) массива char
	2	uint8
	3	uint16
	4	int16
	5	int64
	6	uint8
	7	Структура B
	8	Адрес (uint16) структуры C

Структура B:	1	int64
	2	double
	3	float

Структура C:	1	uint32
	2	int16
	3	uint8
	4	Размер (uint32) и адрес (uint16) массива структур D
	5	Массив int16, размер 8
	6	Массив int32, размер 2

Структура D:	1	int16
	2	int32
	3	Размер (uint32) и адрес (uint16) массива uint8

Примеры разбора двоичного формата с помощью функции f31:

1. Двоичные данные:

```
(b'FRLG\xaa\x00\x05\x00-\xa8\xef\xf5\x81\x97.\xff\x0c\x94\x88\r\xde\x16f|'
b'b\xbbM\xac\x94u}\xbf\xea\x0e\x90\x8f\xc2\xd3D\xbf\x14\x82\x0c\x00bbtqbc\x01'
b'7!d'\xacd\x842\x9ah<\xcf\xfdN\x1c\x00\x00\x00\x03\x002\xdf\x84\x0f\xdb4j'
b'\x00\x00\x00\x04\x005\xcb\x8d\xe9\xaf\x83h\x00\x00\x00\x05\x009\x01\x98'
b'\x1b\xc0\xc5\xb6F\x00\x00\x00\x03\x00>J$\xad\xf6\xfb\x9cd+\xca\x12\x92<\x9f'
b'D\xd5\xe0\xe1=s\xf6Z\xcf\xe4c')
```

Результат разбора:

```
{ 'A1': 'btqbc',
  'A2': 168,
```

```

'A3': 61429,
'A4': -32361,
'A5': 3386439276883205654,
'A6': 102,
'A7': {'B1': 8962932150702994813,
      'B2': -0.8142779167015779,
      'B3': -0.5801093578338623},
'A8': {'C1': 26745792,
      'C2': -14922,
      'C3': 70,
      'C4': [{'D1': 26684, 'D2': -805482980, 'D3': [98, 1, 55]},
              {'D1': -8316, 'D2': 266024042, 'D3': [33, 100, 96, 120]},
              {'D1': -13427,
               'D2': -374373528,
               'D3': [172, 100, 132, 50, 154]}]},
'C5': [23844, -21002, -1124, 25643, -13806, -28100, -24764, -10784],
'C6': [-516066314, 1523573859]}

```

2. Двоичные данные:

```

(b'FRLG\xaa\x00\x03\x00-\xdf\x9b\x97\xadr!`zJ\x01J\xbb\xc7\x11\xc8#\xa'
b'\xc4\xf6\xe1?\xe8\r\x0b\xacL\xa8\x96>\xcf$\xe3\x00Mzfm\x7f\xbc\xa0\xee'
b'\x95m\xa0\xd5\xa6\xaeq\x00\x00\x00\x03\x000\xcbD\x08\xea~\xf2\x00'
b'\x00\x00\x02\x003H9VbF\x80\n\x00\x00\x00\x02\x005\xba\x85,H[\xb0T\xfa\xa7*'
b'\x07\xefr\xe3\rg\xe4h\xadr\x0f\x9e5J')

```

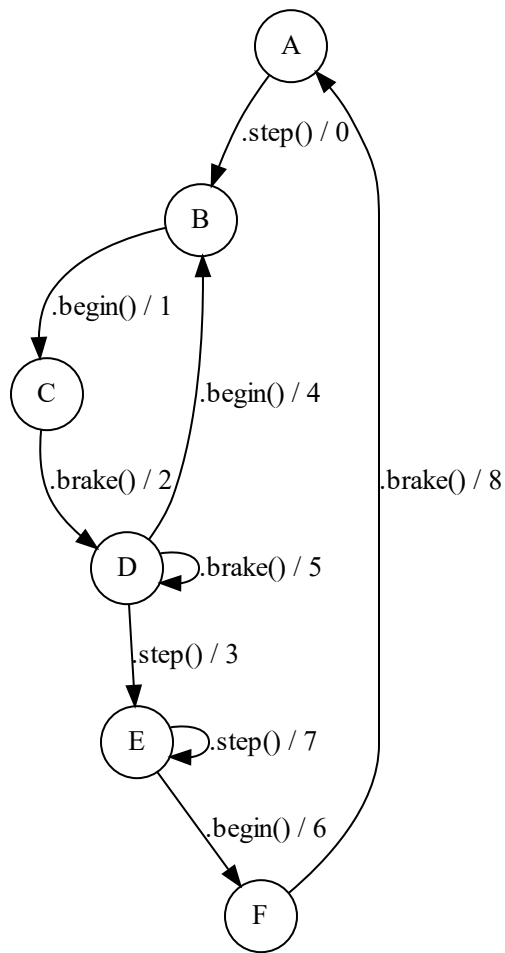
Результат разбора:

```

{'A1': 'zfm',
 'A2': 223,
 'A3': 39783,
 'A4': -26707,
 'A5': 8223960472899111611,
 'A6': 199,
 'A7': {'B1': 1281313107503019745,
      'B2': 0.7515924802031175,
      'B3': 0.404578298330307},
 'A8': {'C1': 1211717218,
      'C2': 18048,
      'C3': 10,
      'C4': [{'D1': 28064, 'D2': -710496655, 'D3': [127, 188, 160]},
              {'D1': -13500, 'D2': 149585650, 'D3': [238, 149]}]},
 'C5': [-17787, 11336, 23472, 21754, -22742, 2031, 29411, 3431],
 'C6': [-462901902, 262026570]}

```

Задача 3.2. Реализовать конечный автомат Мили в виде класса C32. Начальным состоянием автомата является А. Методы возвращают числовые значения. Если вызываемый метод не реализован для некоторого состояния, необходимо вызвать исключение RuntimeError.



1. Пример использования класса C32:

```

o = C32()
o.step()           0
o.begin()          1
o.brake()          2
o.begin()          4
o.begin()          1
o.begin()          RuntimeError
o.brake()          2
o.brake()          5
o.step()           3
o.brake()          RuntimeError
o.step()           7
o.begin()          6
o.begin()          RuntimeError
o.brake()          8
o.step()           0
  
```

2. Пример использования класса C32:

o = C32()	
o.step()	0
o.begin()	1
o.brake()	2
o.begin()	4
o.begin()	1
o.brake()	2
o.brake()	5
o.brake()	5
o.step()	3
o.step()	7
o.begin()	6
o.brake()	8

Вариант №7

Задача 3.1. Реализовать разбор двоичного формата данных (в духе формата WAD игры Doom или графического формата PNG). Данные начинаются с сигнатуры 0x55 0x57 0x4a 0x47 0xc8, за которой следует структура A. Порядок байт: от старшего к младшему. Адреса указаны в виде смещений от начала данных. В решении разрешено использовать модуль struct.

Структура A:	1	uint32
	2	Размер (uint16) и адрес (uint16) массива char
	3	uint64
	4	uint8
	5	Структура B

Структура B:	1	uint16
	2	Размер (uint16) и адрес (uint16) массива адресов (uint32) структур C
	3	Массив float, размер 5
	4	Массив uint8, размер 8
	5	Адрес (uint16) структуры D

Структура C:	1	Размер (uint32) и адрес (uint16) массива int8
	2	float
	3	int32

Структура D:	1	double
	2	uint64
	3	int64

Примеры разбора двоичного формата с помощью функции f31:

1. Двоичные данные:

```
(b'UWJG\xc8\xbf\xd9\xb9\x13\x00\x06\x00:\x9e>V\xaeK\xe1\xcd\x13T\xd8\xfc'  
b'\x00\x02\x00a?=\xd5P\xbd9\x9f/\xbf`N\xea\xbf\x149\x96=2\x13\xd7\xc3iFG'  
b'\xd2A\xe2\xa3\x00ifsidxm>\x14\x96\x00\x00\x00\x03\x00@\xba1U\xe7\x17\xbf7'  
b'\xc953\x00\x00\x00\x02\x00Q?{v\x88Y\x12\xf6\xed\x00\x00\x00C\x00\x00\x00'  
b"S?\xde\xfc\x1d\x81\xb1\x1aXM,tq\xfb\xf4\x0c\xdd\xe50\xb4G'o=.")
```

Результат разбора:

```
{'A1': 3218716947,  
'A2': 'fsidxm',  
'A3': 11402646613146062099,  
'A4': 84,  
'A5': {'B1': 55548,  
       'B2': [{'C1': [62, 20, -106],  
               'C2': -0.0006764814606867731,
```

```

        'C3': 398407625},
        {'C1': [53, 51], 'C2': 0.9822773933410645, 'C3': 1494415085}],
'B3': [0.7415361404418945,
        -0.04531782492995262,
        -0.8762041330337524,
        -0.5790036916732788,
        0.04347595199942589],
'B4': [195, 105, 70, 71, 210, 65, 226, 163],
'B5': {'D1': 0.4841378942987027,
        'D2': 5560947672802921693,
        'D3': -1931846022444663506}}}}

```

2. Двоичные данные:

```

(b'UWJG\xc8x8\x06\xf0\x00\x04\x00:\x19P\x10J\xb6\x17\x0f\nv\x05\x87'
 b'\x00\x03\x00n?\x1a\x19=\xbe\xb2\xefh\xbfiv\x03?3\x81\xb9\xbf\n\xc3='
 b'\xdfM\xff\xde\xa6R\xad\xf5\x00zpsn\xbbo\x00\x00\x02\x00>\xbfSz6\x05\xfc'
 b'$p\xa9f\x00\x00\x00\x02\x00N\xbf)D\xbf\xf7\x0f\xdf19\x97\x00\x00\x00\x02'
 b'\x00^? \x8b\xbc\xaf\x19\xe3\x8e\x00\x00\x00@\x00\x00\x00P\x00\x00'
 b'\x00'\xbf\xe7\r5\xfe\xec\xcfV@E\xa2\xc9\xa9\x94w\xac!\x9c0\xe6\xf5\x98'
 b'\x99\xfb')

```

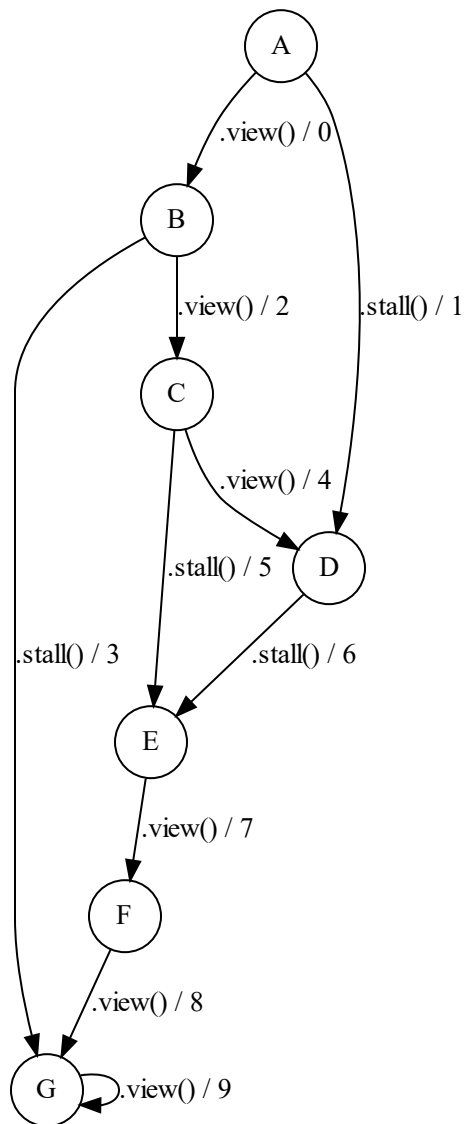
Результат разбора:

```

{'A1': 2016937712,
 'A2': 'ppsn',
 'A3': 1823975762153639690,
 'A4': 118,
 'A5': {'B1': 1415,
        'B2': [{'C1': [-69, 111], 'C2': -0.8260835409164429, 'C3': 100410480},
                {'C1': [-87, 102], 'C2': -0.6612052321434021, 'C3': -149954767},
                {'C1': [57, -105], 'C2': 0.6271321773529053, 'C3': -1357257842}],
        'B3': [0.6019476056098938,
                -0.34948277473449707,
                -0.9119569659233093,
                0.701198160648346,
                -0.5420415997505188],
        'B4': [223, 77, 255, 222, 166, 82, 173, 245],
        'B5': {'D1': -0.7203626612667715,
                'D2': 4631286778837628844,
                'D3': 2421898552999975419}}}}

```

Задача 3.2. Реализовать конечный автомат Мили в виде класса C32. Начальным состоянием автомата является А. Методы возвращают числовые значения. Если вызываемый метод не реализован для некоторого состояния, необходимо вызвать исключение RuntimeError.



1. Пример использования класса C32:

```

o = C32()
o.view()           0
o.view()           2
o.view()           4
o.stall()          6
o.view()           7
o.view()           8
o.view()           9

```

2. Пример использования класса C32:

o = C32()	
o.view()	0
o.view()	2
o.view()	4
o.stall()	6
o.stall()	RuntimeError
o.view()	7
o.view()	8
o.stall()	RuntimeError
o.view()	9
o.stall()	RuntimeError
o.view()	9
o.view()	9
o.view()	9
o.view()	9
o.view()	9
o.stall()	RuntimeError
o.view()	9
o.view()	9

Вариант №8

Задача 3.1. Реализовать разбор двоичного формата данных (в духе формата WAD игры Doom или графического формата PNG). Данные начинаются с сигнатуры 0xa 0x4b 0x4d 0x45, за которой следует структура A. Порядок байт: от младшего к старшему. Адреса указаны в виде смещений от начала данных. В решении разрешено использовать модуль struct.

Структура A:	1	int64
	2	int16
	3	int32
	4	Массив структур B, размер 5
	5	Массив char, размер 4
	6	Адрес (uint32) структуры C

Структура B:	1	int16
	2	uint16
	3	uint32
	4	int8

Структура C:	1	Адрес (uint32) структуры D
	2	Размер (uint16) и адрес (uint16) массива uint8
	3	int64

Структура D:	1	Размер (uint32) и адрес (uint16) массива float
	2	uint64

Примеры разбора двоичного формата с помощью функции f31:

1. Двоичные данные:

```
(b"\nKME\xfa'Ds0\x9a\x86Y=\xe5r\xc0\xc2\x04_\xdc\x8f\x1b\x8d\xbf"
b"\xcc\x88\xbc\xc96\x17e\xd0\xba\xb5\x97HdCN\x12 '\xce\xbc\xa1\xcb\x8a~"
b'\xd1Q\x07\xb3T\xa9\xc5\x04IG!)\x15\xe4\x01ljymc\x00\x00\x00\x1a8\xaf=\x88'
b"\\\\\\?\x86\x06\xbf\x03\x00\x00\x00G\x00\x06\x9a\xc6\xeaia\r\xf6\x0c\x17S"
b'\x00\x00\x00\x02\x00a\x00\xe8\xf4\x1f\x19\xa7\xfbj_')
```

Результат разбора:

```
{'A1': 6451013182281754618,
'A2': -6851,
'A3': 79872114,
'A4': [{'B1': -9121, 'B2': 7055, 'B3': 2295119757, 'B4': -68},
{'B1': 14025, 'B2': 25879, 'B3': 2545269456, 'B4': 72},
{'B1': 17252, 'B2': 4686, 'B3': 3167627040, 'B4': -95},
{'B1': -30005, 'B2': 53630, 'B3': 1421018961, 'B4': -87},
{'B1': 1221, 'B2': 18249, 'B3': 3826592033, 'B4': 1}]}
```

```
'A5': 'ljym',
'A6': {'C1': {'D1': [0.08555622398853302,
                    0.8607869148254395,
                    -0.5254845023155212],
          'D2': 17729934415565855238},
       'C2': [12, 23],
       'C3': 6875584476234118376}}
```

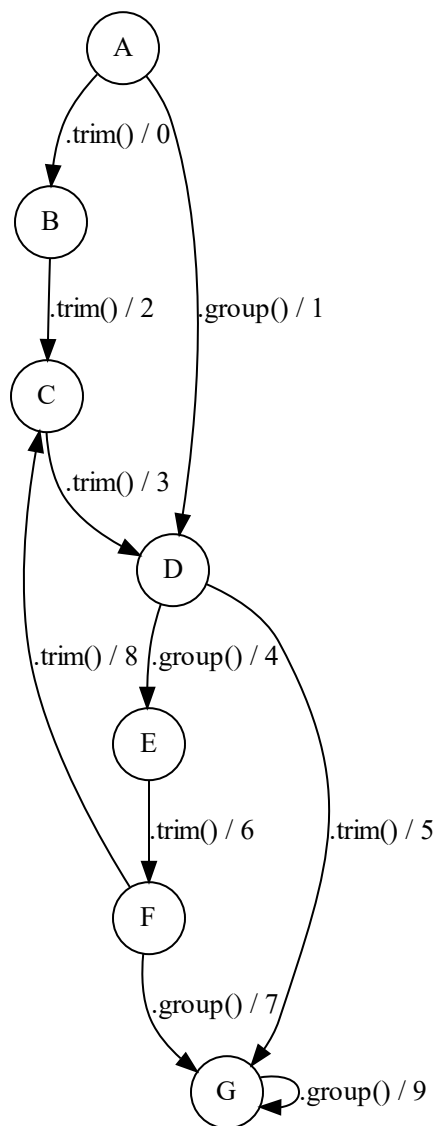
2. Двоичные данные:

```
(b'\nKME\x06\xc8\x05\x979\xbc\xe8\x1aZ\xb5\xfeN\xee\xd5\x87Lm:\xd8\x89'
b'\xb6\x9f\x8b\xe3\xbai]\xea\x91A\xe5\xdb\xf4:\xe9\xd3\xc6\xa4\x9d\x15'
b'\xbbj\xcf\xc0\x95W?\xa0B\xabL\x94Fm\xf4\x90;O!wgfs0\x00\x00\x00\xe7'
b'\xc0\x9d>\xe1\xfd\xd5\xbc\x86z\xa1\xbe\x87\nr>\x8f\xcb\x14>G\x9b\xbf\x06'
b'\x00\x00\x00G\x000\xf8\xe2\xbd\x9dMm\x10\xcb`_\x00\x00\x00\x02\x00m\x00\x9e'
b'\x89\xfe\xe0\xa8\x89\x03\xb6')
```

Результат разбора:

```
{'A1': 1939006595115894790,
 'A2': -19110,
 'A3': -705802498,
 'A4': [{'B1': 19591, 'B2': 14957, 'B3': 2679540184, 'B4': -117},
        {'B1': -17693, 'B2': 23913, 'B3': 3846279658, 'B4': -37},
        {'B1': 15092, 'B2': 54249, 'B3': 362652870, 'B4': -69},
        {'B1': -12438, 'B2': 38336, 'B3': 1117798231, 'B4': -85},
        {'B1': -27572, 'B2': 27974, 'B3': 1329303796, 'B4': 33}],
 'A5': 'wgfs',
 'A6': {'C1': {'D1': [0.3082267940044403,
                    -0.026122035458683968,
                    -0.31538790464401245,
                    0.23636828362941742,
                    0.1453077644109726,
                    -0.9711193442344666],
          'D2': 1183687616959281200},
       'C2': [203, 96],
       'C3': -5331266175454246498}}
```

Задача 3.2. Реализовать конечный автомат Мили в виде класса C32. Начальным состоянием автомата является А. Методы возвращают числовые значения. Если вызываемый метод не реализован для некоторого состояния, необходимо вызвать исключение RuntimeError.



1. Пример использования класса C32:

```

o = C32()
o.trim()           0
o.trim()           2
o.group()          RuntimeError
o.trim()           3
o.group()          4
o.group()          RuntimeError
o.trim()           6
o.trim()           8
o.trim()           3
o.trim()           5
o.group()          9
  
```

2. Пример использования класса C32:

<code>o = C32()</code>	
<code>o.trim()</code>	0
<code>o.trim()</code>	2
<code>o.trim()</code>	3
<code>o.group()</code>	4
<code>o.group()</code>	RuntimeError
<code>o.trim()</code>	6
<code>o.trim()</code>	8
<code>o.trim()</code>	3
<code>o.trim()</code>	5
<code>o.group()</code>	9
<code>o.group()</code>	9
<code>o.group()</code>	9
<code>o.trim()</code>	RuntimeError
<code>o.group()</code>	9
<code>o.group()</code>	9

Вариант №9

Задача 3.1. Реализовать разбор двоичного формата данных (в духе формата WAD игры Doom или графического формата PNG). Данные начинаются с сигнатуры 0x43 0x5a 0x5a 0x55 0x53, за которой следует структура A. Порядок байт: от старшего к младшему. Адреса указаны в виде смещений от начала данных. В решении разрешено использовать модуль struct.

Структура A:	1	uint32
	2	Размер (uint16) и адрес (uint32) массива char
	3	uint16
	4	int32
	5	uint8
	6	int32
	7	Структура B
	8	Массив int16, размер 5

Структура B:	1	Размер (uint32) и адрес (uint16) массива структур C
	2	double
	3	int8
	4	double
	5	Структура D
	6	float

Структура C:	1	int16
	2	uint16
	3	Массив int8, размер 4
	4	int8
	5	int16

Структура D:	1	Размер (uint16) и адрес (uint32) массива int32
	2	uint16
	3	double
	4	int16
	5	float

Примеры разбора двоичного формата с помощью функции f31:

1. Двоичные данные:

```
(b' CZZUS\xb0c<\x92\x00\x02\x00\x00\x00U6\xe3\xb5iq\xb9\x94z\x1c\x8fa\x00\x00'
 b'\x00\x03\x00W\xbf\xc4\x8d\xa2\xbb\xe9\x83\x18K\xbf\xeb\xc3\x05d\x18:'
 b'r\x00\x03\x00\x00\x00\x00\x9eS\xbf\xd7\x17\xa5\x101\xcbD\xf5h>\xa5\xf7\x8a\xbd'
 b'\xfd`K;_Q\xe2\x05h|\x1bo\xeagw\x84\xaf$\xab\x96\xe5\x0b\xe6\x1a\x1f!\xfa\xa8'
 b'\x8d\xd1m\xe7+J\xc5,\xc7\x94\xbf\xf9\xf2\xd6\xbb\x18h\xc9\\J\x8a\xec\xd9\xdf'
 b'9\xf15@TW\xa6\xca')
```

Результат разбора:

```
{'A1': 2959293586,
'A2': 'gw',
'A3': 14051,
'A4': -1251380807,
'A5': 148,
'A6': 2048692065,
'A7': {'B1': [{'C1': -31569,
                'C2': 9387,
                'C3': [-106, -27, 11, -26],
                'C4': 26,
                'C5': 7969},
             {'C1': -1368,
                'C2': 36305,
                'C3': [109, -25, 43, 74],
                'C4': -59,
                'C5': 11463},
             {'C1': -27457,
                'C2': 63986,
                'C3': [-42, -69, 24, 104],
                'C4': -55,
                'C5': 23626}],
'B2': -0.16057237792394896,
'B3': 75,
'B4': -0.86755628156409,
'B5': {'D1': [-1964189217, 972109120, 1415030474],
        'D2': 40531,
        'D3': -0.3608181627416125,
        'D4': -2712,
        'D5': 0.32415419816970825},
'B6': -0.12371882051229477},
'A8': [15199, 20962, 1384, 31771, 28650]}
```

2. Двоичные данные:

```
(b'CZZUS\xe5\xd6\xb6c\x00\x03\x00\x00\x00U\xfad\x95J\x19\xaf\xcb]\xaa'
b's\x17\x00\x00\x00\x03\x00X?\xcf\xbdK\x96\xe0\x85h\x0b\xbf\x9aw\xb8s\x07\xac'
b"@\x00\x04\x00\x00\x00y\x1fv\xbf\xe0\xa6m\x9d\xb9\xd1\xae\xbbC?'\xeb\x14\xbf"
b'\x1b*\xf3\xf7\x94\x11v\xd2\xd5\x04_\x8e\x91bax\xf3\xa3\xfe @\xebsi\x93\xde"V'
b'\xba\xf6\xbe\xa0\xc09\x9f:|Y\xc1E\x87\x85\xc7\x1b\x85yN\xb3\xf4\\;\xe6'
b'\xc9(\xf5\x8c\xc7\x97\x1e\xdc\xbdw\xf6L\x85')
```

Результат разбора:

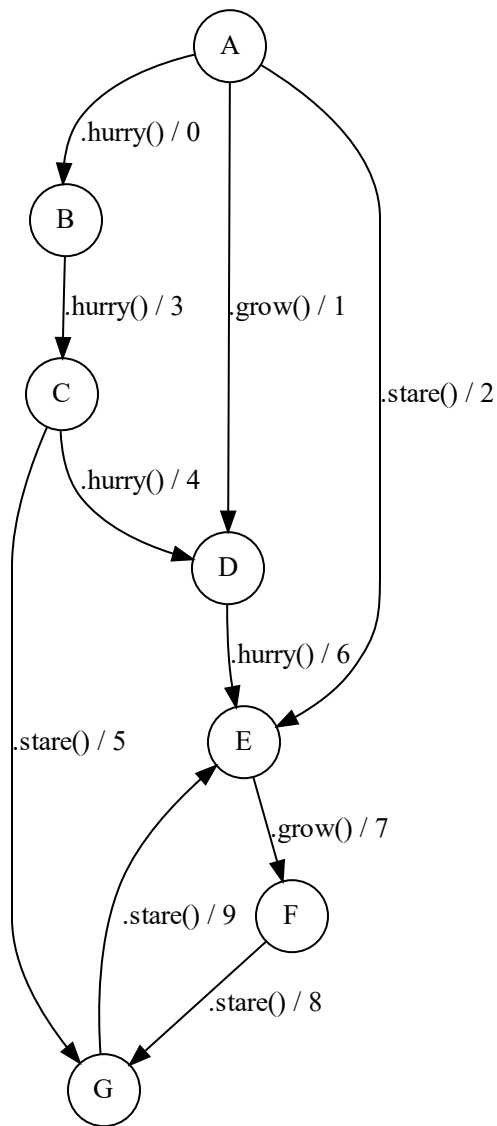
```
{'A1': 3856053859,
'A2': 'bax',
'A3': 64100,
```

```

'A4': -1790305873,
'A5': 203,
'A6': 1571451671,
'A7': {'B1': [{'C1': -3165,
                'C2': 65056,
                'C3': [64, -21, 115, 105],
                'C4': -109,
                'C5': -8670},
              {'C1': 22202,
                'C2': 63166,
                'C3': [-96, -64, 57, -97],
                'C4': 58,
                'C5': 31833},
              {'C1': -16059,
                'C2': 34693,
                'C3': [-57, 27, -123, 121],
                'C4': 78,
                'C5': -19468}],
      'B2': 0.24796433321993594,
      'B3': 11,
      'B4': -0.025847322483894652,
      'B5': {'D1': [1547429577, 687180999, -1759585091, 2012630149],
              'D2': 8054,
              'D3': -0.5203159409125442,
              'D4': -17597,
              'D5': 0.655930757522583},
      'B6': -0.6061241030693054},
'A8': [-2156, 4470, -11563, 1119, -29039]}

```

Задача 3.2. Реализовать конечный автомат Мили в виде класса C32. Начальным состоянием автомата является A. Методы возвращают числовые значения. Если вызываемый метод не реализован для некоторого состояния, необходимо вызвать исключение RuntimeError.



1. Пример использования класса C32:

<code>o = C32()</code>	
<code>o.hurry()</code>	0
<code>o.hurry()</code>	3
<code>o.grow()</code>	RuntimeError
<code>o.hurry()</code>	4
<code>o.hurry()</code>	6
<code>o.grow()</code>	7
<code>o.hurry()</code>	RuntimeError
<code>o.stare()</code>	8
<code>o.hurry()</code>	RuntimeError
<code>o.stare()</code>	9
<code>o.grow()</code>	7
<code>o.stare()</code>	8
<code>o.stare()</code>	9

2. Пример использования класса C32:

<code>o = C32()</code>	
<code>o.hurry()</code>	0
<code>o.hurry()</code>	3
<code>o.hurry()</code>	4
<code>o.hurry()</code>	6
<code>o.grow()</code>	7
<code>o.stare()</code>	8
<code>o.stare()</code>	9
<code>o.stare()</code>	RuntimeError
<code>o.grow()</code>	7
<code>o.stare()</code>	8
<code>o.stare()</code>	9
<code>o.grow()</code>	7
<code>o.stare()</code>	8

Вариант №10

Задача 3.1. Реализовать разбор двоичного формата данных (в духе формата WAD игры Doom или графического формата PNG). Данные начинаются с сигнатуры 0x59 0x56 0x54 0x48 0x46, за которой следует структура A. Порядок байт: от младшего к старшему. Адреса указаны в виде смещений от начала данных. В решении разрешено использовать модуль struct.

Структура A:	1	int64
	2	float
	3	Структура B
	4	Массив float, размер 4

Структура B:	1	int32
	2	Адрес (uint32) структуры C
	3	uint16
	4	Массив адресов (uint16) структур D, размер 7
	5	uint16
	6	Массив int16, размер 6
	7	Размер (uint32) и адрес (uint32) массива uint64

Структура C:	1	int16
	2	uint16
	3	uint64
	4	int32
	5	Размер (uint16) и адрес (uint32) массива uint16
	6	int8
	7	uint32

Структура D:	1	int32
	2	int8

Примеры разбора двоичного формата с помощью функции f31:

1. Двоичные данные:

```
(b'YVTHF\x83\xc6\\<\xec\x82b\x87\x91\x93\x9c<\x80\x1e8}5\x00\x00\x00\xf5\xfdn'
b'\x00s\x00x\x00}\x00\x82\x00\x87\x00\x8c\x00\xaf\x94\xccai@\x17\xa1\x9c\xd3C'
b']\x05V\x02\x00\x00\x00\x91\x00\x00\x00?\xec\x03?\x8f\x1f\x84>{\x8aX\xbf9'
b'H6\xbf@b\xa4\xe5\xc84\x0f\xbf\x11IIqE\x00\x1a\xf7%F\xad\x7f\x02\x000\x00\x00'
b'\x00f+\xe2\xedQ\x18\xf8\xf874\xab\xe0\xaf\xe7\x97w\xd9\xa4"\xd6\xc6\x19\xe5'
b'\x851|\xd2\xa1H\xe2\xc0L\x02\xbcX\xf3y\xdf\xbb\xc0)\xe6\x85\x88\xf8\x01\xb1'
b'\xf13\x8c\xc38\xe6\x02w\x96')
```

Результат разбора:

```
{'A1': -8691240379735030141,
```



```

'A2': 0.019113333895802498,
'A3': {'B1': 2100829824,
      'B2': {'C1': 13512,
            'C2': 48911,
            'C3': 17805544375016048913,
            'C4': 2142062117,
            'C5': [25152, 58788],
            'C6': 102,
            'C7': 1374544427},
      'B3': 65013,
      'B4': [{'D1': 939063320, 'D2': 52},
            {'D1': -407904085, 'D2': -105},
            {'D1': 581228919, 'D2': -42},
            {'D1': -2048583226, 'D2': 108},
            {'D1': 1218564732, 'D2': -30},
            {'D1': -1140699968, 'D2': 88},
            {'D1': -1142982157, 'D2': -64}],
      'B5': 38063,
      'B6': [25036, 16489, -24297, -11364, 23875, 22021],
      'B7': [17415703400972412457, 10842137815735831603]},
'A4': [0.5153235793113708,
      0.2580532729625702,
      -0.8458630442619324,
      -0.7120395302772522]}

```

2. Двоичные данные:

```

(b'YVTHF.\xa6+\xa0\x846U5\xf9CY\xbfq-ZkU\x00\x00\x00\x80\xe2p\x00u\x00z'
 b'\x00\x7f\x00\x84\x00\x89\x00\x8e\x00\xfc\x89Z\x98F\x99\x07\x9bb\xef\x8e'
 b'\x1d[\xe6\x03\x00\x00\x00\x93\x00\x00\x00\x060\x81>\x12\x12{?\x0f'
 b'[\x92\xbc\xae\x91\x10=\x10o\x92\x17[1((\xb8\xe9Aad\xe9\xf8\x92\xb7\xfaZo$'
 b'_\x03\x000\x00\x00\x00\xbf\xc6\xd4\xa1\xc9\xdb\x0b\xfb\xfc\x13\xbd\xd4'
 b'_2\xd3\x1fq\x14VY\xcc\xd1FMF\x085\x17\x9f\xdb\xa8\xee\xed\xdb\xd0\xd3'
 b'\xe2\x1cE\xdf\x17u\x6h[.\x05PG[\x92\xd9\x99\x12\xd4\x0c\x15\xfc\x03'
 b'D\xd9\xc9')

```

Результат разбора:

```

{'A1': 3843037800281384494,
 'A2': -0.8486934304237366,
 'A3': {'B1': 1801071985,
      'B2': {'C1': 10280,
            'C2': 59832,
            'C3': 18066070027985248577,
            'C4': 1596223354,
            'C5': [28432, 6034, 27739],

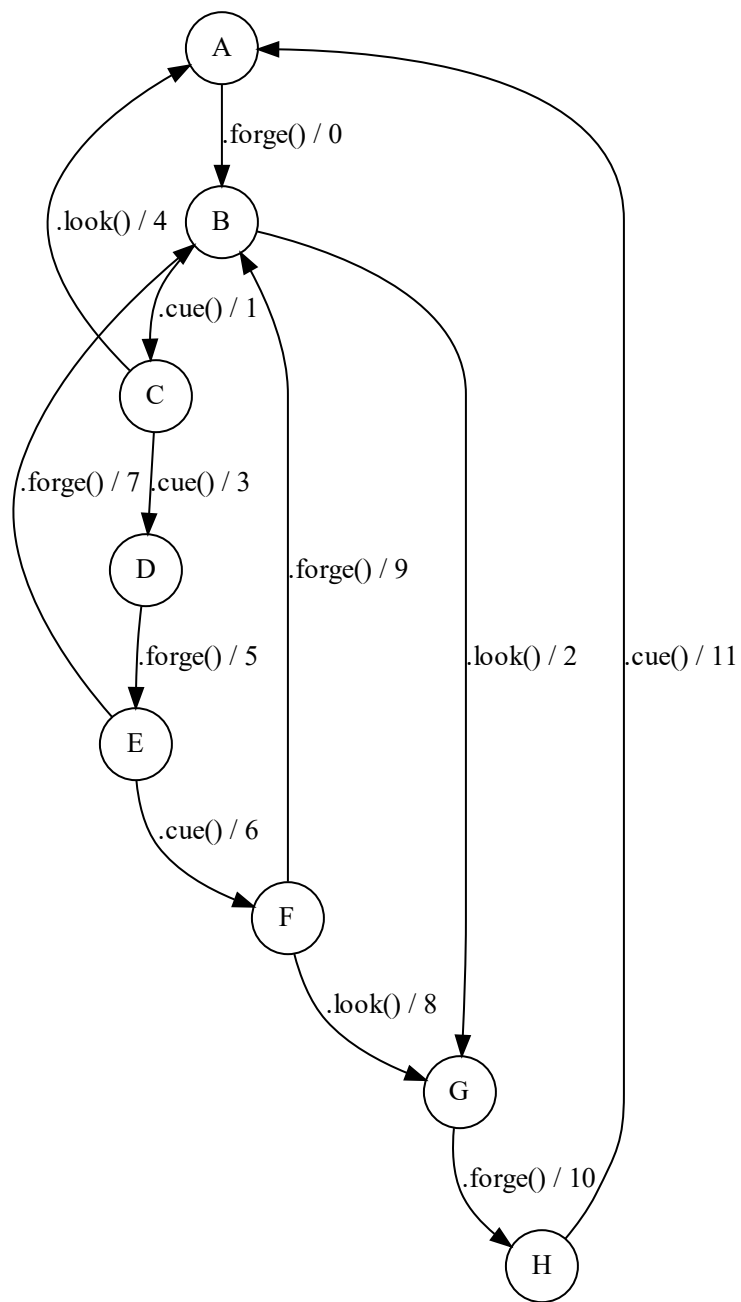
```

```

        'C6': -65,
        'C7': 1822086342},
    'B3': 57984,
    'B4': [{ 'D1': -83108919, 'D2': -4},
            { 'D1': 1607777555, 'D2': 50},
            { 'D1': 342958035, 'D2': 86},
            { 'D1': 1188154457, 'D2': 77},
            { 'D1': 389351494, 'D2': -97},
            { 'D1': -303126309, 'D2': -37},
            { 'D1': 484627408, 'D2': 69}],
    'B5': 35324,
    'B6': [-26534, -26298, -25849, -4254, 7566, -6565],
    'B7': [3340378956361066207, 1340341586739286021, 14544731255474556116]},
    'A4': [0.2525560259819031,
            0.980744481086731,
            -0.017865685746073723,
            0.03529518097639084]}

```

Задача 3.2. Реализовать конечный автомат Мили в виде класса C32. Начальным состоянием автомата является А. Методы возвращают числовые значения. Если вызываемый метод не реализован для некоторого состояния, необходимо вызвать исключение RuntimeError.



1. Пример использования класса C32:

```

o = C32()
o.forge()          0
o.look()           2
o.forge()          10
o.cue()            11
o.cue()            RuntimeError
o.forge()          0
o.look()           2
o.forge()          10
o.cue()            11
o.forge()          0
o.cue()            1
o.cue()            3
o.forge()          5
o.cue()            6
o.forge()          9
o.cue()            1
o.look()           4

```

2. Пример использования класса C32:

```

o = C32()
o.forge()          0
o.look()           2
o.forge()          10
o.cue()            11
o.forge()          0
o.cue()            1
o.look()           4
o.forge()          0
o.cue()            1
o.cue()            3
o.look()           RuntimeError
o.forge()          5
o.look()           RuntimeError
o.cue()            6
o.look()           8
o.forge()          10
o.cue()            11

```

Вариант №11

Задача 3.1. Реализовать разбор двоичного формата данных (в духе формата WAD игры Doom или графического формата PNG). Данные начинаются с сигнатуры 0x8f 0x50 0x58 0x52 0x47, за которой следует структура A. Порядок байт: от старшего к младшему. Адреса указаны в виде смещений от начала данных. В решении разрешено использовать модуль struct.

Структура A:	1	int32
	2	Массив адресов (uint16) структур B, размер 5
	3	float
	4	Адрес (uint32) структуры C
	5	Адрес (uint16) структуры D
	6	int8
	7	int8

Структура B:	1	uint8
	2	int64

Структура C:	1	int16
	2	int16

Структура D:	1	Массив char, размер 4
	2	Адрес (uint32) структуры E
	3	Размер (uint16) и адрес (uint32) массива int8
	4	Адрес (uint32) структуры F

Структура E:	1	Размер (uint32) и адрес (uint32) массива int32
	2	uint32

Структура F:	1	int32
	2	uint16
	3	int16

Примеры разбора двоичного формата с помощью функции f31:

1. Двоичные данные:

```
(b'\x8fPXRg<\x04\x06X\x00\x1f\x00(\x001\x00:\x00C?\x0f\xa8\xa0\x00\x00L\x00'
 b'r\x93\xebh\x13\xca\\xcfc\xcf\x03 (\x93\xb1\x93Qs\xdc\x93\xdb[\x11MJ'
 b'\x01UZ\xfc\xe5D\xa2\x14\x0b\x94\xd4{\x8d\x92\xf5\xc4\t\xa0\x18{\xec<\xc8\x8b'
 b'\xe0wSI\xaeG^\x08\x9f\nhs\xfbH\x8c\xd7\x00\x00\x00\x03\x00\x00\x00P'
 b'\x1b\x12\\\xa4\x11\xc0\x1e\xf2\x91q\x07\x08\xd7\xb4oqlv\x00\x00'
 b'\x00\\\x00\x02\x00\x00\x00h\x00\x00\x00j')
```

Результат разбора:

```
{'A1': 1006896728,
'A2': [{'B1': 104, 'B2': 1426054279617257512},
{'B1': 147, 'B2': -5651083549356074149},
{'B1': 17, 'B2': 5569265355307410756},
{'B1': 162, 'B2': 1444411745829229301},
{'B1': 196, 'B2': 693581263138506891}],
'A3': 0.5611667633056641,
'A4': {'C1': -8073, 'C2': 21321},
'A5': {'D1': 'oqlv',
'D2': {'E1': [-1371054584, -1626707853, -79131433], 'E2': 454188196},
'D3': [17, -64],
'D4': {'F1': 519213425, 'F2': 1800, 'F3': -10316}},
'A6': -109,
'A7': -21}
```

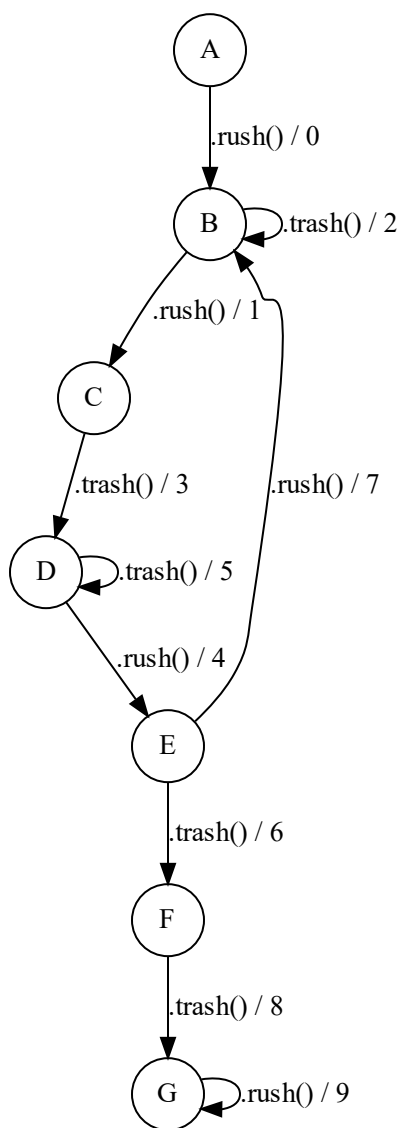
2. Двоичные данные:

```
(b'\x8fPXRgK\x04Z#\x00\x1f\x00(\x001\x00:\x00C>\x98\xda\xc5\x00\x00L\x00'
b'o\xa6e\t^\xc2\xf0+\xc9\x03\xba\x08"7\x81\x8f\xcaB\x1f8b\xdf\x14\xff'
b'\x1f\xf4\x9d\xd3?.\xf1\x04z\x9d\xb7A\xf7\xfdHQ>\xc5k\xdd\x86\xdf\xbb\xd6'
b'\xfb\xaa\xdf4\x8a"\xae\xbea\xed\x87\xa6\x00\x00\x00\x02\x00\x00\x00P'
b'\xb1\x8b{7\xe5\x93\x1f\xfc\xed\xab!\x97\x85Yzrb\x00\x00\x00X\x00'
b'\x03\x00\x00\x00d\x00\x00\x00g')
```

Результат разбора:

```
{'A1': 1258576419,
'A2': [{'B1': 9, 'B2': 6828284055893817864},
{'B1': 34, 'B2': 3999636042937219170},
{'B1': 223, 'B2': 1512963135300124462},
{'B1': 241, 'B2': 322743733721759048},
{'B1': 81, 'B2': 4523139999958285270}],
'A3': 0.2985440790653229,
'A4': {'C1': -1110, 'C2': -8396},
'A5': {'D1': 'yzrb',
'D2': {'E1': [-1977438530, 1642956710], 'E2': 2978708279},
'D3': [-27, -109, 31],
'D4': {'F1': -60691029, 'F2': 8599, 'F3': -31399}},
'A6': -90,
'A7': 101}
```

Задача 3.2. Реализовать конечный автомат Мили в виде класса C32. Начальным состоянием автомата является А. Методы возвращают числовые значения. Если вызываемый метод не реализован для некоторого состояния, необходимо вызвать исключение RuntimeError.



1. Пример использования класса C32:

o = C32()	
o.rush()	0
o.rush()	1
o.trash()	3
o.rush()	4
o.rush()	7
o.trash()	2
o.rush()	1
o.trash()	3
o.trash()	5
o.rush()	4
o.trash()	6
o.trash()	8
o.rush()	9
o.rush()	9

2. Пример использования класса C32:

o = C32()	
o.trash()	RuntimeError
o.rush()	0
o.rush()	1
o.rush()	RuntimeError
o.trash()	3
o.rush()	4
o.rush()	7
o.trash()	2
o.trash()	2
o.rush()	1
o.trash()	3
o.trash()	5
o.rush()	4
o.trash()	6
o.trash()	8
o.rush()	9

Вариант №12

Задача 3.1. Реализовать разбор двоичного формата данных (в духе формата WAD игры Doom или графического формата PNG). Данные начинаются с сигнатуры 0x58 0x54 0x52 0x7a, за которой следует структура A. Порядок байт: от старшего к младшему. Адреса указаны в виде смещений от начала данных. В решении разрешено использовать модуль struct.

Структура A:	1	int8
	2	uint8
	3	Адрес (uint16) структуры B
	4	Массив float, размер 3
	5	int8

Структура B:	1	float
	2	Структура C
	3	Массив структур G, размер 2
	4	int64
	5	Размер (uint32) и адрес (uint32) массива uint32

Структура C:	1	double
	2	int16
	3	Адрес (uint32) структуры D
	4	Структура F

Структура D:	1	Структура E
	2	uint8
	3	uint16

Структура E:	1	Массив int64, размер 4
	2	uint16

Структура F:	1	uint32
	2	float
	3	int8

Структура G:	1	int64
	2	double
	3	uint32
	4	int32

Примеры разбора двоичного формата с помощью функции f31:

1. Двоичные данные:

```
(b'XTRzi\xa3\x00F\xbe\xc4e\xdb=\xfc\xc5\x18\xbe\xff\xf0\x1a\xae~s\xc5'  
b'\x9f\xc7\xb8\xac1iV\xd4\xf6\xdd;\xbdG1\xc8\xe3\xd01\xb4\xfb#\xed\xb9\x18'
```

```

b'\x02\xbc(\xcc\xfc=j\xc2\x981-\x0f\xb1+\xa2\xac\x91\xf2\xbc\xabu>\xbe\x0f'
b'\xdd9\xbf\xe5[\x9b6\xaf\xfc\x1d\x00\x00\x00\x15\xc8%\x11\x8d=\xe14+'
b'\x18\x06\x15\x9cy\xc1\xa6\x1e\x0b?\xe1\xa7\x14\x87\xc2\x0c\xbc\x0b.'
b'\xb2\xa7\xd0\xa3\x12\xef\xd6[$\xdap\x84\x19?\xe46\xfe\xcd8\x90\xaaZ\x85/'
b'7\xa9J\xba R_\xe2\x99\x89\xa5\xac\x19\x00\x00\x00\x03\x00\x00\x00:')

```

Результат разбора:

```

{'A1': 105,
 'A2': 163,
 'A3': {'B1': -0.14049233496189117,
        'B2': {'C1': -0.6674324100600983,
                 'C2': 25885,
                 'C3': {'D1': {'E1': [9111843761141886001,
                                       7590488378718272839,
                                       3587367587517102883,
                                       -1316995016003367690],
                                'E2': 15722},
                        'D2': 194,
                        'D3': 38961},
                 'C4': {'F1': 3357872525, 'F2': 0.10996278375387192, 'F3': 24}},
        'B3': [{'G1': 438428585492356619,
                  'G2': 0.5516455317996569,
                  'G3': 1762340530,
                  'G4': -1479498990},
                 {'G1': -1164643239786937319,
                  'G2': 0.6317132957670044,
                  'G3': 1518677815,
                  'G4': -1454720480}],
        'B4': 5935711982964812825,
        'B5': [756003115, 2729218546, 3165353278]},
 'A4': [-0.38358959555625916, 0.12342280149459839, -0.4998787045478821],
 'A5': -82}

```

2. Двоичные данные:

```

(b'XTRz|\xe1\x00F\xbe)\x9cE\xbe\xc4\x1b\xa3\xbf\x14_\xcb\xdb\xa9=\xe4'
b'\x9c\xee\x8e\xc8\xc1\x8f\xd8\x9b\x16X>P\xed\x8a5Y\xc0/\x04\xb4X0\x07\xc5'
b'\xd3\x05?\x9bG\xee\xe0DG\xb0B[\xfd\x02\x03\ro\xbd)\xf5\xab\x91\xbf\x1d'
b"\xca'? \xef\xb9\x97\xab \x12\xe8\xce\x02\x00\x00\x00\x15\xe9\xd6\xf2\xa8"
b'\xbfc:\xc8\xcf\xb9/\xb8\x18\xa6A\xf8\xc0?\xe9\xe8\xf2\x87V\xa8\xb9\xaaC'
b'G1\xe0\xa8a\x9c\xec\xd9\xd28\x95\xbb\xa9?\xe4W\x9f\x04\xc9Y\xf4]\x9f\n'
b'\xb9{ecZ\xc1\x9f]\xc6\t\xb6g5\x00\x00\x00\x03\x00\x00\x00:')

```

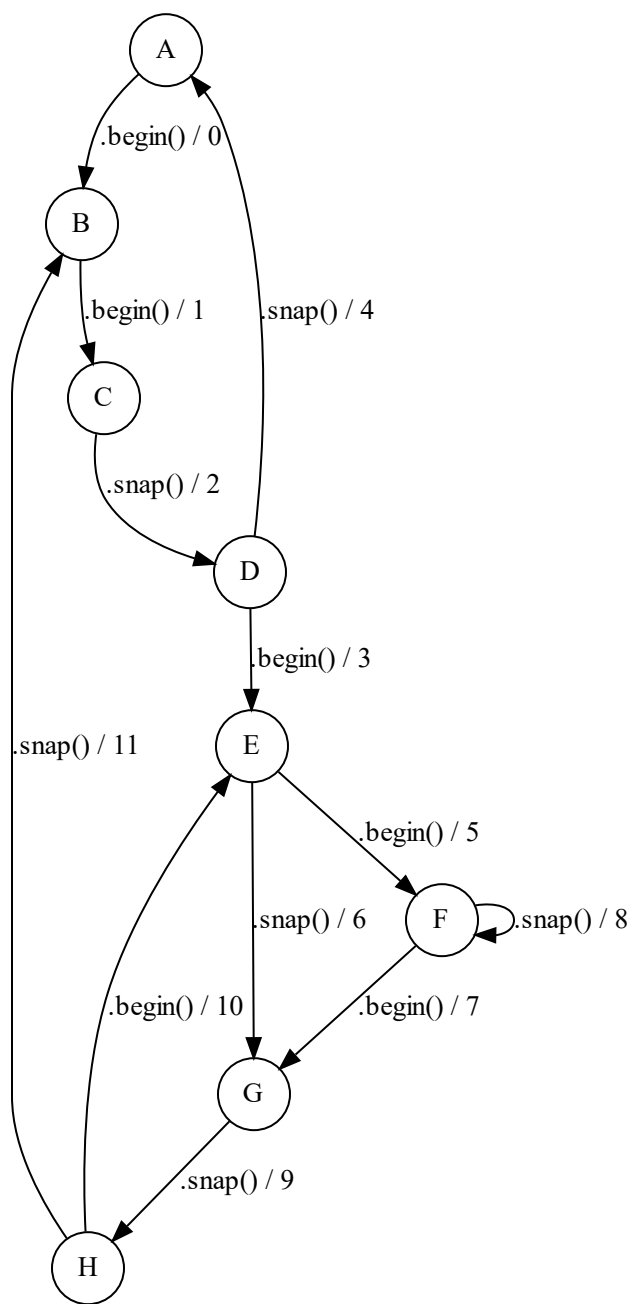
Результат разбора:

```

{'A1': 124,
 'A2': 225,
 'A3': {'B1': -0.6163658499717712,
        'B2': {'C1': 0.9914053289690772,
                 'C2': -12798,
                 'C3': {'D1': {'E1': [-6251589345052014399,
                                     -8081539011044290323,
                                     -8487779240752401320,
                                     5694737763950107463],
                                'E2': 61152},
                        'D2': 68,
                        'D3': 18352},
                 'C4': {'F1': 3923178122, 'F2': -0.7626156806945801, 'F3': -49}},
        'B3': [{'G1': -5102657436779415360,
                  'G2': 0.8095974671257027,
                  'G3': 3114943303,
                  'G4': 836806753},
                 {'G1': -7139091812373382231,
                  'G2': 0.635695943211259,
                  'G3': 1570704057,
                  'G4': 2070242138}],
        'B4': -4494770797944608971,
        'B5': [1113324802, 51212221, 703966097]},
 'A4': [-0.16563518345355988, -0.38302335143089294, -0.5795866847038269],
 'A5': -37}

```

Задача 3.2. Реализовать конечный автомат Мили в виде класса C32. Начальным состоянием автомата является А. Методы возвращают числовые значения. Если вызываемый метод не реализован для некоторого состояния, необходимо вызвать исключение `RuntimeError`.



1. Пример использования класса C32:

o = C32()	
o.begin()	0
o.begin()	1
o.begin()	RuntimeError
o.snap()	2
o.begin()	3
o.begin()	5
o.snap()	8
o.begin()	7
o.snap()	9
o.begin()	10
o.snap()	6
o.snap()	9
o.snap()	11
o.begin()	1
o.snap()	2
o.snap()	4
o.begin()	0

2. Пример использования класса C32:

o = C32()	
o.begin()	0
o.begin()	1
o.snap()	2
o.begin()	3
o.snap()	6
o.snap()	9
o.begin()	10
o.begin()	5
o.snap()	8
o.begin()	7
o.snap()	9
o.snap()	11
o.begin()	1
o.snap()	2
o.snap()	4
o.begin()	0

Вариант №13

Задача 3.1. Реализовать разбор двоичного формата данных (в духе формата WAD игры Doom или графического формата PNG). Данные начинаются с сигнатуры 0x55 0x56 0x42 0x41, за которой следует структура А. Порядок байт: от младшего к старшему. Адреса указаны в виде смещений от начала данных. В решении разрешено использовать модуль struct.

Структура А:	1	Адрес (uint16) структуры В
	2	Структура С
	3	int32
	4	Размер (uint16) и адрес (uint16) массива int64
	5	uint16
	6	Массив float, размер 2

Структура В:	1	uint16
	2	uint16

Структура С:	1	Массив char, размер 2
	2	Массив структур D, размер 7
	3	float
	4	int16
	5	Массив uint32, размер 4

Структура D:	1	float
	2	uint16

Примеры разбора двоичного формата с помощью функции f31:

1. Двоичные данные:

```
(b'UVBAZ\x00ua\x13\x1b\xbd\xbd\xaf5N\x16\x89\xbe\x91\xee\xd1b\xcb>\x804\x9b\x15'
b'\x96\xbe\x02\xe3\xea\xbe\xbb>\x8b\xcb\xb0\x87U\xbf\xf4\xa0\xe2\xa7\x12\xbf'
b'vY\xa7q,>\xce\x9c).w\x0c\x967`p\x9bf\x15T\x08\xa2R\xae\x00\r\xb4\xba'
b'\x02\x00^\x00Y\xee\xea\xa67?A}B?\xe9\x14\xccQ\xddj}\x0b\xd6\xbd'
b'\x1f\xfd\n\x1d6\xe3\x00\x97\xe6\xf5')
```

Результат разбора:

```
{ 'A1': { 'B1': 5353, 'B2': 20940 },
  'A2': { 'C1': 'ua',
    'C2': [ { 'D1': -0.09233679622411728, 'D2': 13743 },
      { 'D1': -0.26774829626083374, 'D2': 61073 },
      { 'D1': 0.39723828434944153, 'D2': 13440 },
      { 'D1': -0.29313358664512634, 'D2': 58114 },
      { 'D1': 0.3666909337043762, 'D2': 52107 },
      { 'D1': -0.834101676940918, 'D2': 41204 },
```

```

        {'D1': -0.5728741884231567, 'D2': 22902}},
    'C3': 0.168402299284935,
    'C4': -25394,
    'C5': [209137193, 1885353878, 1410688667, 2924651016]},
    'A3': -1162605312,
    'A4': [-207238330822337059, -727728259705987830],
    'A5': 61017,
    'A6': [0.7173906564712524, 0.7597237229347229]}

```

2. Двоичные данные:

```

(b'UVBAZ\x00ft\xc5\xb6>U\\N\xa7\x1a?AI\xe1\xd7s\xbe8|\xec]?>\x10\x06'
b'\xf9M\xca=\xb8\xe8\x15\x92\xe9>\xe5\x08\xd0\x96?\xd4\xe4\xfe8??\xd4\xa0'
b'\xf8\x1ay\x91/\xcf]\x13\xfc\x8a\x93"\x84I\nl\x83\xc9M\xa5\x02\x00^\x00'
b'\xf9\xb1\xa6\xde|?g\x16\xe?\x89=\xce\xab7\xbf=y\x8c\x85!?\x9d\xd3x*\xa3\xf4'
b'\xdbg')

```

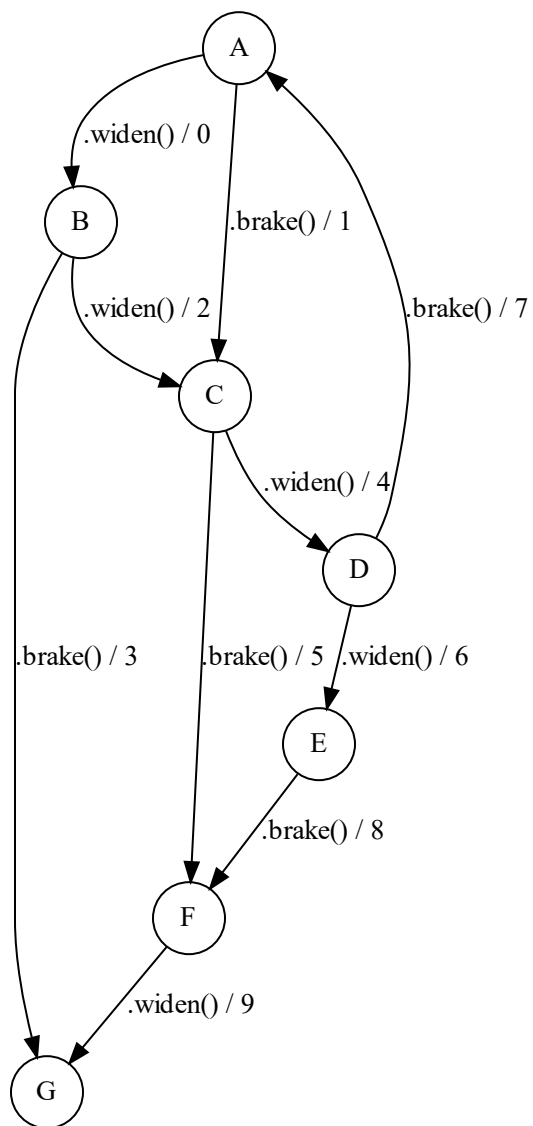
Результат разбора:

```

{'A1': {'B1': 15753, 'B2': 43982},
 'A2': {'C1': 'ft',
        'C2': [{'D1': 0.3565122187137604, 'D2': 23637},
                 {'D1': 0.60411536693573, 'D2': 18753},
                 {'D1': -0.238128200173378, 'D2': 31800},
                 {'D1': 0.18688172101974487, 'D2': 1552},
                 {'D1': 0.0987815335392952, 'D2': 59531},
                 {'D1': 0.30970826745033264, 'D2': 2277},
                 {'D1': 0.7132844924926758, 'D2': 58580}],
        'C3': 0.746963381767273,
        'C4': -24364,
        'C5': [2440633080, 324914991, 580094716, 1812613508]},
    'A3': -1521628797,
    'A4': [4549063936996917047, 7483844187435619229],
    'A5': 45561,
    'A6': [0.9877723455429077, 0.5550293326377869]}

```

Задача 3.2. Реализовать конечный автомат Мили в виде класса C32. Начальным состоянием автомата является А. Методы возвращают числовые значения. Если вызываемый метод не реализован для некоторого состояния, необходимо вызвать исключение RuntimeError.



1. Пример использования класса C32:

```

o = C32()
o.widen()           0
o.widen()           2
o.widen()           4
o.brake()           7
o.brake()           1
o.widen()           4
o.widen()           6
o.brake()           8
o.widen()           9
  
```

2. Пример использования класса C32:

o = C32()	
o.widen()	0
o.widen()	2
o.widen()	4
o.brake()	7
o.brake()	1
o.widen()	4
o.brake()	7
o.widen()	0
o.widen()	2
o.widen()	4
o.widen()	6
o.brake()	8
o.widen()	9

Вариант №14

Задача 3.1. Реализовать разбор двоичного формата данных (в духе формата WAD игры Doom или графического формата PNG). Данные начинаются с сигнатуры 0xа0 0х53 0х55 0х56, за которой следует структура А. Порядок байт: от младшего к старшему. Адреса указаны в виде смещений от начала данных. В решении разрешено использовать модуль struct.

Структура А:	1	double
	2	float
	3	Размер (uint16) и адрес (uint32) массива char
	4	Адрес (uint32) структуры В
	5	int64
	6	Адрес (uint32) структуры D
	7	Массив структур Е, размер 2
	8	uint32

Структура В:	1	uint16
	2	Адрес (uint16) структуры С
	3	uint8

Структура С:	1	uint16
	2	Массив int64, размер 7
	3	int16

Структура D:	1	uint32
	2	float

Структура Е:	1	float
	2	Размер (uint16) и адрес (uint32) массива int32

Примеры разбора двоичного формата с помощью функции f31:

1. Двоичные данные:

```
(b'\xa0SUV\xa8\x99bg\x07_\xd4\xbf,\xaY\xbe\x03\x00>\x00\x00\x00}\x00\x00\x00iV'
b'\x81\xc6\xf3\x1e\x89b\x82\x00\x00\x00\x9d;\xeb>\x04\x00\x8a\x00\x00\x00VD\n?'
b'\x05\x00\x9a\x00\x00\x00\xcc\x1e\x17dk\x1f4\x8f\x19\xacBf\x12\xf5\xdfI\xe6'
b'\x0f\xf8\xf7\x83^\x03\xac}#\xbeL \n^\x0c@\xb2\xa7\x05\xde!B\\\x8bD\xda[\x93'
b'\x19\x12r\xab\x16\x8a\xf7\xd5\xb9\x99\xa7\xcc\x01\xf1\x7f\xfb\x9e\x6r/'
b'\xf9\xe4\xf5A\x00\xfd\x8e\x18NZ\xb4f\x18?\xf7\x0cJ\x1a\xab\x7f\x0c\x194\x9b'
b'\x8d\xc1}U\xa0\xa7?";D\x19\x85*\xa3S[\xc6\xc1\xc9\xed\x0f.L\x17c\xb4')
```

Результат разбора:

```
{ 'A1': -0.3183001050587344,
  'A2': -0.21244114637374878,
```

```

'A3': 'dki',
'A4': {'B1': 62948,
      'B2': {'C1': 36852,
            'C2': [5323242743949077529,
                  -6051889553361858586,
                  891160910108697469,
                  6647913237413605952,
                  8219660390596494475,
                  -6369855869306661205,
                  8283983367615087052],
            'C3': -1745},
      'B3': 253},
'A5': 7100240319882548841,
'A6': {'D1': 1515067534, 'D2': 0.5953171253204346},
'A7': [{'E1': 0.4594391882419586,
      'E2': [441060479, 420249515, -1047684300, -1482664579]},
      {'E1': 0.5401052236557007,
      'E2': [1144726079, -1557494503, -1043965101, 772795849, -1268574388]}],
'A8': 387871948}

```

2. Двоичные данные:

```

(b'\xa0SUV\xe07P8l\xac\xa5?4\xff\x10\xbf\x02\x00>\x00\x00\x00|\x00\x00\x00 \x0f'
b'<\xaf\x02\xea\xe8\x81\x00\x00\x00>pj?\x03\x00\x89\x00\x00\x00M\xa3m?'
b'\x05\x00\x95\x00\x00\x00\xb3F\x18\x0edf\x14\r3\x16\xd4k\x11\xf7@\xe0\x87 '
b'\xbcXs\xe8\x01\xbe\x05`xc0\xa6\xb0^\x9e\xd9\x02\xcc7\xff7VJ\x161?'
b'\x10\xdd\xb1\xe1\x13\xa2\x18*:\x9b\x9b\xdd\xeb\xf6\xa9\xa6\x90\xc8\x0f&'
b"\x93\xfb\xee\x94'\n@\x00\xe2*\xcFU\xbd\xf2D1\xbf\xba\xf8\xa8\xe2\xb0\xc4\xc4"
b"\x8c?H$\x0ep\xc1&\xe1\xaa'Z(\xfe\xe0\xb3\x81\x0b\xaa\x8ez\x85\xc6\xd2")

```

Результат разбора:

```

{'A1': 0.042331106065929935,
'A2': -0.5663940906524658,
'A3': 'df',
'A4': {'B1': 2599,
      'B2': {'C1': 3348,
            'C2': [-2287556956508645837,
                  -4755264149418925945,
                  -2765669008354025467,
                  1606191015616039938,
                  -6767817660973433039,
                  -654185660458325480,
                  -318869299381885271],
            'C3': -27410},
      'B3': 226},

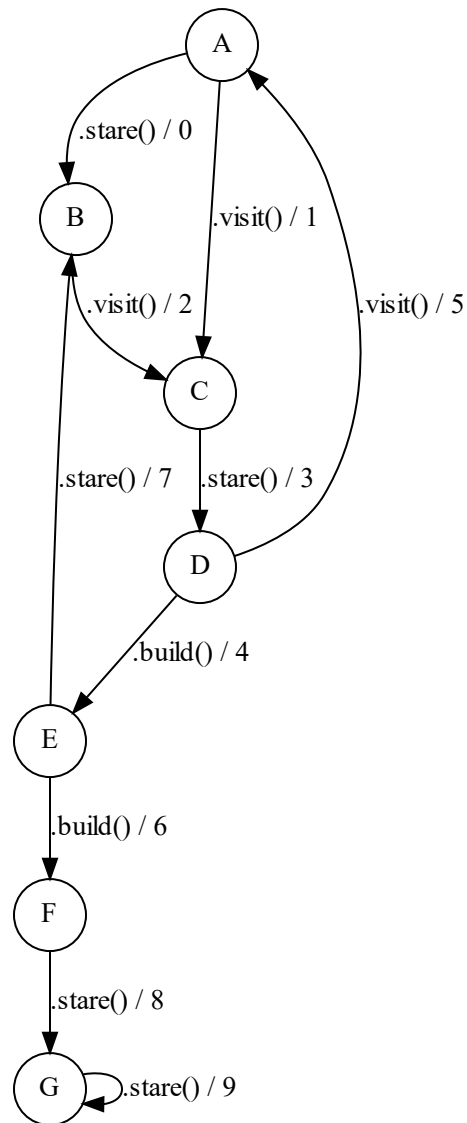
```

```

'A5': -1663514161059197152,
'A6': {'D1': 3176517418, 'D2': -0.6924582719802856},
'A7': [{'E1': 0.9157751798629761, 'E2': [-492242758, -1933261648, 237258815]},
        {'E1': 0.9282730221748352,
         'E2': [-517553808, 1512534186, -1277100504, -1901458559, -758741638]}],
'A8': 236471987}

```

Задача 3.2. Реализовать конечный автомат Мили в виде класса C32. Начальным состоянием автомата является A. Методы возвращают числовые значения. Если вызываемый метод не реализован для некоторого состояния, необходимо вызвать исключение RuntimeError.



1. Пример использования класса C32:

o = C32()	
o.visit()	1
o.stare()	3
o.visit()	5
o.stare()	0
o.visit()	2
o.stare()	3
o.build()	4
o.stare()	7
o.visit()	2
o.stare()	3
o.stare()	RuntimeError
o.build()	4
o.build()	6
o.stare()	8
o.stare()	9

2. Пример использования класса C32:

o = C32()	
o.stare()	0
o.visit()	2
o.stare()	3
o.build()	4
o.stare()	7
o.visit()	2
o.stare()	3
o.stare()	RuntimeError
o.visit()	5
o.visit()	1
o.stare()	3
o.build()	4
o.build()	6
o.stare()	8
o.visit()	RuntimeError
o.stare()	9

Вариант №15

Задача 3.1. Реализовать разбор двоичного формата данных (в духе формата WAD игры Doom или графического формата PNG). Данные начинаются с сигнатуры 0x44 0x4b 0x4e 0x56 0x50, за которой следует структура A. Порядок байт: от старшего к младшему. Адреса указаны в виде смещений от начала данных. В решении разрешено использовать модуль struct.

Структура A:	1	double
	2	Адрес (uint32) структуры B
	3	Адрес (uint16) структуры D
	4	uint8
	5	uint32
	6	uint32

Структура B:	1	Размер (uint16) и адрес (uint16) массива char
	2	Размер (uint16) и адрес (uint32) массива адресов (uint32) структур C
	3	int64
	4	int32
	5	uint16
	6	float
	7	int32

Структура C:	1	double
	2	int32
	3	uint16
	4	uint32
	5	int32

Структура D:	1	Размер (uint32) и адрес (uint32) массива double
	2	Массив int16, размер 2

Примеры разбора двоичного формата с помощью функции f31:

1. Двоичные данные:

```
(b'DKNVP\xbf\xec\x91RL[\x84\x1e\x00\x00\x00R\x00\x92\x81\xeb;q\xebbK\xf8%'  
b'sm\xbf\xe8\xc9\x82\xd1\x0e\xe7@\xae\xb8\xbc\xae+\x8b\x87\xf3\x1b\xa2'  
b'\x1f\x9f'\xd6\xbf\xd8\x80\x10GLXP\x87\xea|\xb1YB\x80?6\x15\x05\x19'  
b'\x98m\x00\x00\x00\x1e\x00\x00\x004\x00\x02\x00\x1c\x00\x02\x00\x00\x00J'  
b'Z\xc9\xb4\xc4\x16\xa6w\xc6-\x1d\xf0p\xe7#=\xad\x08\xd8,\xb3q\xc4\xbf\xd0'  
b'\xb9<\xa2R\x8f'\xbf\xe0k\xb4wx.?\xe6\xed\xf4\x99\xd5!r?\xdfR\xf5\x10\x06'  
b'\xf1\xa8\x00\x00\x00\x04\x00\x00\x00r\xfc#\xd1\xc7')
```

Результат разбора:

```
{'A1': -0.8927394381863711,
```

```

'A2': {'B1': 'sm',
      'B2': [{'C1': -0.7745985110443385,
                'C2': -1363624786,
                'C3': 11147,
                'C4': 2280856482,
                'C5': 530538710},
              {'C1': -0.3828163810989098,
                'C2': -2014675791,
                'C3': 22850,
                'C4': 2151626261,
                'C5': 85563501}],
      'B3': 6541958688018954182,
      'B4': 756936816,
      'B5': 59171,
      'B6': 0.08448952436447144,
      'B7': 749957572},
'A3': {'D1': [-0.2613059601956049,
              -0.5132349514713093,
              0.7165472988645389,
              0.4894383102908031],
      'D2': [-989, -11833]},
'A4': 129,
'A5': 3946476011,
'A6': 1649145893}

```

2. Двоичные данные:

```

(b'DKNVP\xbf\xe5\x9d\xdaR\xe5\xb6>\x00\x00\x00S\x00\x83\xb4\xcd\xed\xef\xe5'
b'| \xb3\xef_tug?\xcdoGK\xe6\xf2 @H\x19b\xa5\x9c\xfa\xeen\xeeB\xb6\xc8'
b'\xbc?\xb2\xa7R\x0e\xfd\xdc\xb0\xd9oj\x08\x08K\xc4\xc5\xd2\x03\xcf'
b'N\x98\x8c\x00\x00\x00\x1f\x00\x00\x00S\x00\x03\x00\x1c\x00\x02\x00\x00\x00'
b'K\xf7\xf1j*\x9f\xa0,2\xd4\xa7\x1d[C\xde\xbe\xba\x19\xa4\xfd\xb0\x10\x1f\xbf'
b'\xa6(\xde3y\x9c` \xbf\xdbg\x06\xb6b\xa6\xd0\x00\x00\x00\x02\x00\x00\x00s\xd6'
b'N\x99\xbc')

```

Результат разбора:

```

{'A1': -0.6755191439712644,
 'A2': {'B1': 'tug',
      'B2': [{'C1': 0.22995845037611762,
                'C2': 1078466914,
                'C3': 42396,
                'C4': 4209929966,
                'C5': 1119275196},
              {'C1': 0.07286560884482429,
                'C2': -647009784,

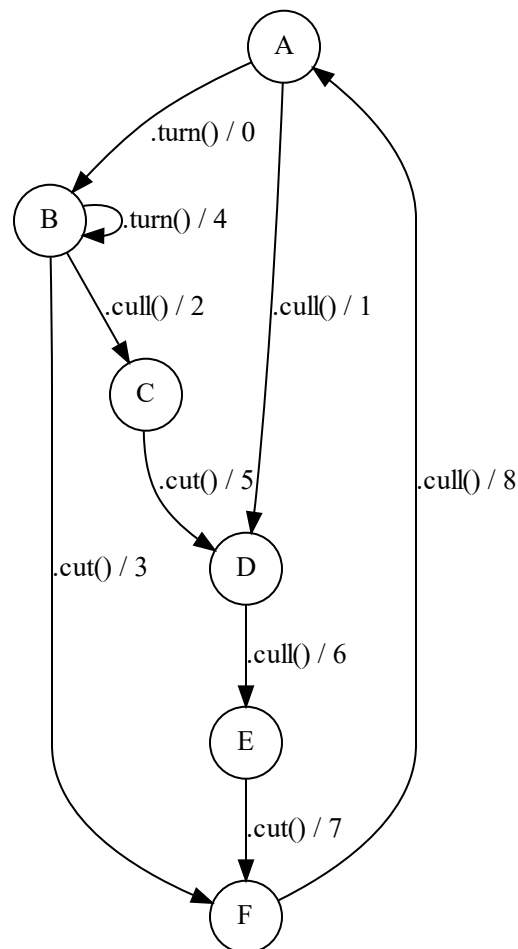
```

```

        'C3': 2123,
        'C4': 3301298691,
        'C5': -816932724}],
    'B3': -580566145654838222,
    'B4': -727245477,
    'B5': 17374,
    'B6': -0.36347687244415283,
    'B7': -38793185},
    'A3': {'D1': [-0.04328054788930369, -0.42816322146482566],
           'D2': [-10674, -26180]},
    'A4': 180,
    'A5': 3454922725,
    'A6': 2092167007}

```

Задача 3.2. Реализовать конечный автомат Мили в виде класса C32. Начальным состоянием автомата является A. Методы возвращают числовые значения. Если вызываемый метод не реализован для некоторого состояния, необходимо вызвать исключение RuntimeError.



1. Пример использования класса C32:

<code>o = C32()</code>	
<code>o.cut()</code>	RuntimeError
<code>o.turn()</code>	0
<code>o.cut()</code>	3
<code>o.cull()</code>	8
<code>o.turn()</code>	0
<code>o.turn()</code>	4
<code>o.cull()</code>	2
<code>o.cut()</code>	5
<code>o.cut()</code>	RuntimeError
<code>o.cull()</code>	6
<code>o.cut()</code>	7
<code>o.cut()</code>	RuntimeError
<code>o.cull()</code>	8
<code>o.cull()</code>	1

2. Пример использования класса C32:

<code>o = C32()</code>	
<code>o.cut()</code>	RuntimeError
<code>o.turn()</code>	0
<code>o.cut()</code>	3
<code>o.turn()</code>	RuntimeError
<code>o.cull()</code>	8
<code>o.cull()</code>	1
<code>o.cut()</code>	RuntimeError
<code>o.cull()</code>	6
<code>o.cut()</code>	7
<code>o.cut()</code>	RuntimeError
<code>o.cull()</code>	8
<code>o.turn()</code>	0
<code>o.turn()</code>	4
<code>o.cull()</code>	2
<code>o.cut()</code>	5

Вариант №16

Задача 3.1. Реализовать разбор двоичного формата данных (в духе формата WAD игры Doom или графического формата PNG). Данные начинаются с сигнатуры 0x10 0x47 0x53 0x52, за которой следует структура A. Порядок байт: от младшего к старшему. Адреса указаны в виде смещений от начала данных. В решении разрешено использовать модуль struct.

Структура A:	1	Размер (uint16) и адрес (uint16) массива char
	2	Структура B
	3	Массив int32, размер 2
	4	int8

Структура B:	1	Размер (uint16) и адрес (uint32) массива структур C
	2	Адрес (uint16) структуры D
	3	uint64

Структура C:	1	float
	2	int16

Структура D:	1	float
	2	float
	3	float
	4	Массив int8, размер 8
	5	int16
	6	int64
	7	uint32

Примеры разбора двоичного формата с помощью функции f31:

1. Двоичные данные:

```
(b"\x10GSR\x06\x00!\x00\x04\x00'\x00\x00\x00?\x00H'\xf3\xb8\x0310."
b'\x94\x10\x06\x87\xc9\xe7P\xacKkugppv0\xf5]\xbf\xf7u\x9c1\x8c>\x1b\xb36'
b'\xb5\xc3>\x8fI\x97H\xbb\xbd\xc9(LRL\xbdF\xdd5?\xeb@T\xbf\x94\xaa."\x05'
b'&\xcc\xf5:\xcf\xa1\xb2a,\xe5\x9b\x95\xfe\xbd\xe6\xe2V')
```

Результат разбора:

```
{ 'A1': 'kugppv',
  'A2': { 'B1': [ { 'C1': -0.8670243620872498, 'C2': 30199 },
                  { 'C1': 0.27381598949432373, 'C2': -19685 },
                  { 'C1': 0.3822419047355652, 'C2': 18831 },
                  { 'C1': -0.09144704788923264, 'C2': 10441 } ],
    'B2': { 'D1': -0.049883171916007996,
            'D2': 0.7104076147079468,
            'D3': -0.8291155695915222,
```

```

'D4': [-108, -86, 46, 34, 5, 38, -52, -11],
'D5': -12486,
'D6': -102004007951551839,
'D7': 1457710781},
'B3': 3328278887870455880},
'A3': [-2029645676, -1403983927],
'A4': 75}

```

2. Двоичные данные:

```

(b'\x10GSR\x05\x00!\x00\x06\x00&\x00\x00\x00J\x00\x8f\xa1,v\x1c\xe4\xd5\xe'
b'\xf0\x196]\xdb\xc2Y*\xa7dxxwj\t\xa8\xbc\xbd\xc2\t,*\x12?Y0Y\xb7'
b'\xf5\xbe\x85\x0e\xbd\xc1\xa3\xbdUq\xaa\x08\x9d\xbeo\x99q\x01\x03\xbf'
b'\xc3R\x16%L\xbf\xaas\xfd>}\xfc)\xbf*\x14\x1f\xc1\xd0\x1dh:\xe4\xb6Q\x12{% '
b'\4\4\4\4\xa6\x0c\x18\x8b')

```

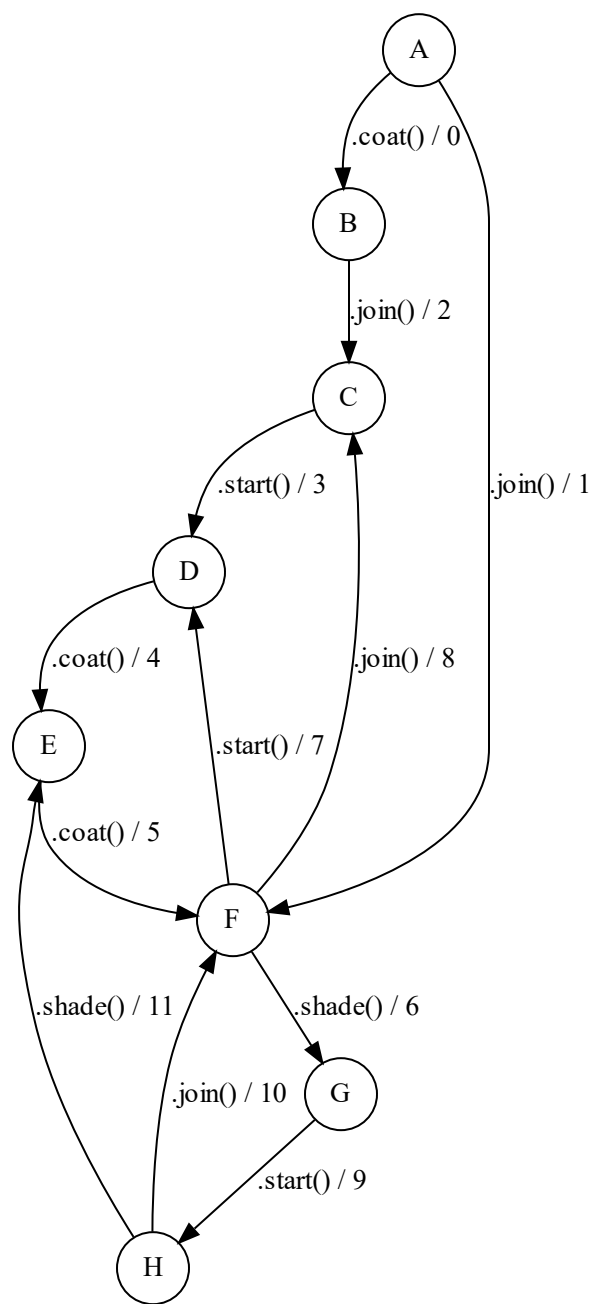
Результат разбора:

```

{'A1': 'dxxwj',
 'A2': {'B1': [{'C1': -0.09211737662553787, 'C2': 2498},
                {'C1': 0.5709559917449951, 'C2': 12377},
                {'C1': -0.47991445660591125, 'C2': 3717},
                {'C1': -0.07995937019586563, 'C2': 29013},
                {'C1': -0.3067067265510559, 'C2': -26257},
                {'C1': -0.5117407441139221, 'C2': 21187}],
        'B2': {'D1': -0.7974408864974976,
                'D2': 0.49502307176589966,
                'D3': -0.6640089154243469,
                'D4': [42, 20, 31, -63, -48, 29, 104, 58],
                'D5': -18716,
                'D6': 1467105148986397265,
                'D7': 2333609126},
        'B3': 2221932802070061455},
 'A3': [1563826672, 710525659],
 'A4': -89}

```

Задача 3.2. Реализовать конечный автомат Мили в виде класса C32. Начальным состоянием автомата является А. Методы возвращают числовые значения. Если вызываемый метод не реализован для некоторого состояния, необходимо вызвать исключение RuntimeError.



1. Пример использования класса C32:

o = C32()	
o.coat()	0
o.join()	2
o.start()	3
o.coat()	4
o.coat()	5
o.start()	7
o.coat()	4
o.coat()	5
o.join()	8
o.start()	3
o.coat()	4
o.coat()	5
o.shade()	6
o.start()	9
o.shade()	11
o.coat()	5

2. Пример использования класса C32:

o = C32()	
o.coat()	0
o.join()	2
o.start()	3
o.coat()	4
o.start()	RuntimeError
o.coat()	5
o.join()	8
o.start()	3
o.coat()	4
o.coat()	5
o.shade()	6
o.start()	9
o.join()	10
o.coat()	RuntimeError
o.start()	7

Вариант №17

Задача 3.1. Реализовать разбор двоичного формата данных (в духе формата WAD игры Doom или графического формата PNG). Данные начинаются с сигнатуры 0x91 0x57 0x54 0x4d 0x4e, за которой следует структура A. Порядок байт: от старшего к младшему. Адреса указаны в виде смещений от начала данных. В решении разрешено использовать модуль struct.

Структура A:	1	Массив адресов (uint32) структур B, размер 2
	2	Размер (uint16) и адрес (uint32) массива char
	3	Структура D

Структура B:	1	Массив uint8, размер 6
	2	Адрес (uint32) структуры C
	3	float
	4	int32

Структура C:	1	int32
	2	double
	3	int32
	4	int16

Структура D:	1	uint64
	2	Массив uint8, размер 4
	3	Размер (uint16) и адрес (uint16) массива double
	4	double

Примеры разбора двоичного формата с помощью функции f31:

1. Двоичные данные:

```
(b'\x91WTMN\x00\x00\x00=\x00\x00\x00a\x00\x02\x00\x00\x00s\xc5\x02\xab\x19\xb7'
b'\x04R\xd4\xe0\xef \r\x00\x02\x00u?\xdd\xd5\x8501\x9f\xd4m#u\xf3?'
b'\xe0\xb2\x95\xd3w\x8e\xaa\xbb\xf41\x97\xc7@\xe0g\xdf\xae\xc0\x98\x00'
b"\x00\x00+?Y\xf7w'\xea1\x99\xc1e\xa6\xc1?\xe3zV\x8dZ\x8f\xa8\x89\xef\xef\xad0"
b'\xa2\xa7 \x0b\xe7\x80\x9e\x00\x00\x000\xbe\xea\xde$\xfb\xaa\xb7uqs\xbf\xd7w'
b'K\xc4+\xa4\xcc\xbf\xed\xe3\xa8MGX\x98')
```

Результат разбора:

```
{'A1': [{'B1': [224, 103, 223, 174, 192, 152],
            'B2': {'C1': 1831040499,
                    'C2': 0.5217999582495925,
                    'C3': -1141609321,
                    'C4': -14528},
            'B3': 0.8514322638511658,
            'B4': 669674649},
```

```
{'B1': [167, 32, 11, 231, 128, 158],
  'B2': {'C1': -1050302783,
        'C2': 0.6086838494124873,
        'C3': -1980764243,
        'C4': 12450},
  'B3': -0.45872604846954346,
  'B4': -72697995}],
'A2': 'qs',
'A3': {'D1': 14196097102358270676,
      'D2': [224, 239, 32, 13],
      'D3': [-0.36665624768518223, -0.9340402135228656],
      'D4': 0.46615727170728616}}
```

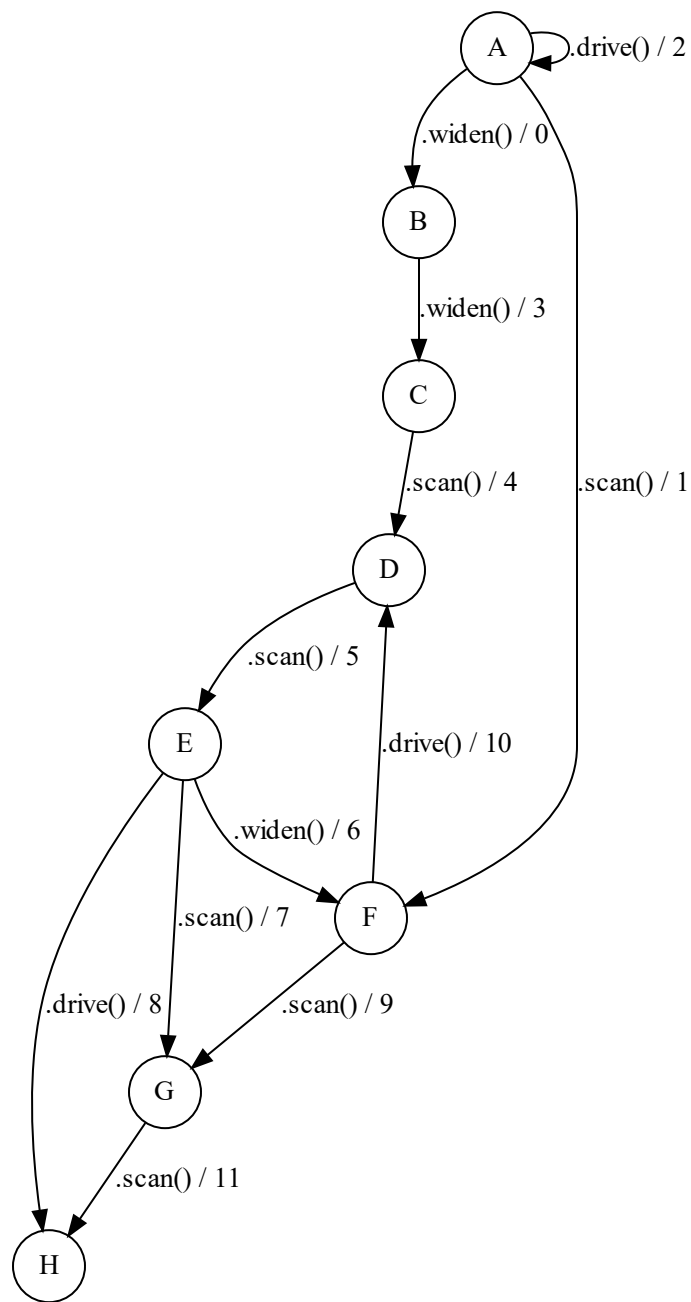
2. Двоичные данные:

```
(b'\x91WTMN\x00\x00\x00=\x00\x00\x00a\x00\x02\x00\x00\x00s\xd0)\x12\xd9, '
b'\x8d,\xf2\xe\xba?>\x00\x02\x00u\xbf\xea\xa3\xf6D>\xf4\xce\xd1\x8f\xe8G\xbf '
b'\xdbA4F\xba\x0f\xa4\xe1\xb8\x95\xe)V\xc4\x07\x82D y\x00\x00\x00+\xbe '
b'P6\xdb\x98\x95;\xcd\xd6\xf2C\xef?\xe0y\x0ePY\x08\xae%\x8d\xd0\xe7\xa9 '
b'p3\xc6\x7f\xe5\x06\xe5\x00\x00\x000?\x12\x17.\xbby\xdf\x8fdp\xbf\xd5\x1f '
b'\x900[\xeaH?\xediLz{\xf8\xea')
```

Результат разбора:

```
{'A1': [{'B1': [196, 7, 130, 68, 32, 121],
          'B2': {'C1': -779098041,
                  'C2': -0.42585474879182983,
                  'C3': -507996786,
                  'C4': 10582},
          'B3': -0.2033342570066452,
          'B4': -1735050291},
        {'B1': [51, 198, 127, 229, 6, 229],
          'B2': {'C1': -688765969,
                  'C2': 0.5147773331921337,
                  'C3': 630051047,
                  'C4': -22160},
          'B3': 0.5706661939620972,
          'B4': -1149640817}],
'A2': 'dp',
'A3': {'D1': 14999540757898800370,
      'D2': [142, 186, 63, 62],
      'D3': [-0.3300514671570096, 0.919103850581249],
      'D4': -0.8325148899963553}}
```

Задача 3.2. Реализовать конечный автомат Мили в виде класса C32. Начальным состоянием автомата является А. Методы возвращают числовые значения. Если вызываемый метод не реализован для некоторого состояния, необходимо вызвать исключение RuntimeError.



1. Пример использования класса C32:

o = C32()	
o.drive()	2
o.drive()	2
o.widen()	0
o.widen()	3
o.scan()	4
o.scan()	5
o.widen()	6
o.drive()	10
o.scan()	5
o.scan()	7
o.scan()	11

2. Пример использования класса C32:

o = C32()	
o.drive()	2
o.widen()	0
o.widen()	3
o.scan()	4
o.scan()	5
o.widen()	6
o.drive()	10
o.scan()	5
o.widen()	6
o.widen()	RuntimeError
o.scan()	9
o.widen()	RuntimeError
o.scan()	11

Вариант №18

Задача 3.1. Реализовать разбор двоичного формата данных (в духе формата WAD игры Doom или графического формата PNG). Данные начинаются с сигнатуры 0x5a 0x4a 0x4f 0xf5, за которой следует структура A. Порядок байт: от младшего к старшему. Адреса указаны в виде смещений от начала данных. В решении разрешено использовать модуль struct.

Структура A:	1	uint64
	2	Структура B

Структура B:	1	Размер (uint32) и адрес (uint32) массива структур C
	2	Структура D

Структура C:	1	uint16
	2	float
	3	int64

Структура D:	1	int32
	2	Адрес (uint32) структуры E
	3	int16

Структура E:	1	uint16
	2	int8
	3	Массив uint16, размер 6
	4	int32
	5	Массив int8, размер 7
	6	Размер (uint32) и адрес (uint32) массива int8

Примеры разбора двоичного формата с помощью функции f31:

1. Двоичные данные:

```
(b'ZJ0\xf5bI\xa1]\xc6\xb0Z\xc1\x05\x00\x00\x00\x1e\x00\x00\x00\xf0\xfb\xae6'
b'f\x00\x00\x00\x07p\x19\xcb\xbd\x1eT\xbf\xf6;\x81\x81M\xb0\x8ewg\xe1(t'
b'\xca\xbd^\xee\xa0<Z\x10\x06g\x93\x89\xee\x82;\xcd\x14\xaa\xb8[zi\x1a'
b'>\x05\xe6\xb8\xc0\xbd\x8d\xcd\xbb\xfcX\xf5\x17\xe3\n\xd7\xdf\xf14?{\xdaZ0'
b'\x0b\xdb%\x07q4\x8by\xa6\xbbp\xb8\xe9\xe0\x82\xdbp\xed\x0bG\xad\xdc\xe9'
b'\xfc5\xce!\x11|\x8f4\x02\x00\x00\x00d\x00\x00\x00')
```

Результат разбора:

```
{'A1': 13932642763244915042,
'A2': {'B1': [{'C1': 51993,
                'C2': -0.8285940289497375,
                'C3': 8615016984138038262},
         {'C1': 57703,
```

```

        'C2': -0.09885436296463013,
        'C3': 7423639015517122142},
    {'C1': 35219,
     'C2': 0.7324665784835815,
     'C3': 1905720126689907917},
    {'C1': 1342,
     'C2': -0.09410266578197479,
     'C3': -2082926540089471603},
    {'C1': 55050,
     'C2': 0.7068156599998474,
     'C3': 515058574026267259}],
    'B2': {'D1': -1503265808,
           'D2': {'E1': 31115,
                  'E2': -90,
                  'E3': [28859, 59832, 12521, 56194, 60784, 18187],
                  'E4': -51782483,
                  'E5': [53, -50, 33, 17, 124, -113, 52],
                  'E6': [113, 52]},
           'D3': 28679}}

```

2. Двоичные данные:

```

(b'ZJ0\xf5\xd8\xaa\xed\xd8R\xa\x18?\x05\x00\x00\x00\x1e\x00\x00\x00'
 b'\x90?\xca\xebf\x00\x00\x00\xd14Z]\xe5\xcb^?\x83\xd3y\xf0E-\xa3\xf9'
 b'\x87\x14\x12\x85\xae>\xb7\x80K.\x15\xdd\x19\x0e\xcf\x15\xf45\xbe'
 b'\xdbC\x1c\x14\\\xbb\xe0\x1b\x1b_T\x93N?d\x1d\x07|\xf9\x02\x80ef\xa1'
 b'\xd4\xbc]\xbfY \x83\x1a\x81\x98\xe6\xa4\xa5>\x981\x14\xf5\xa7\xd3'
 b'\x92]\xbf\xf8\xe5:\xfa\x1c\x98d\x98\x88Ay\xbe0\xa7\x98\xd6W\x02\x00\x00\x00'
 b'd\x00\x00\x00')

```

Результат разбора:

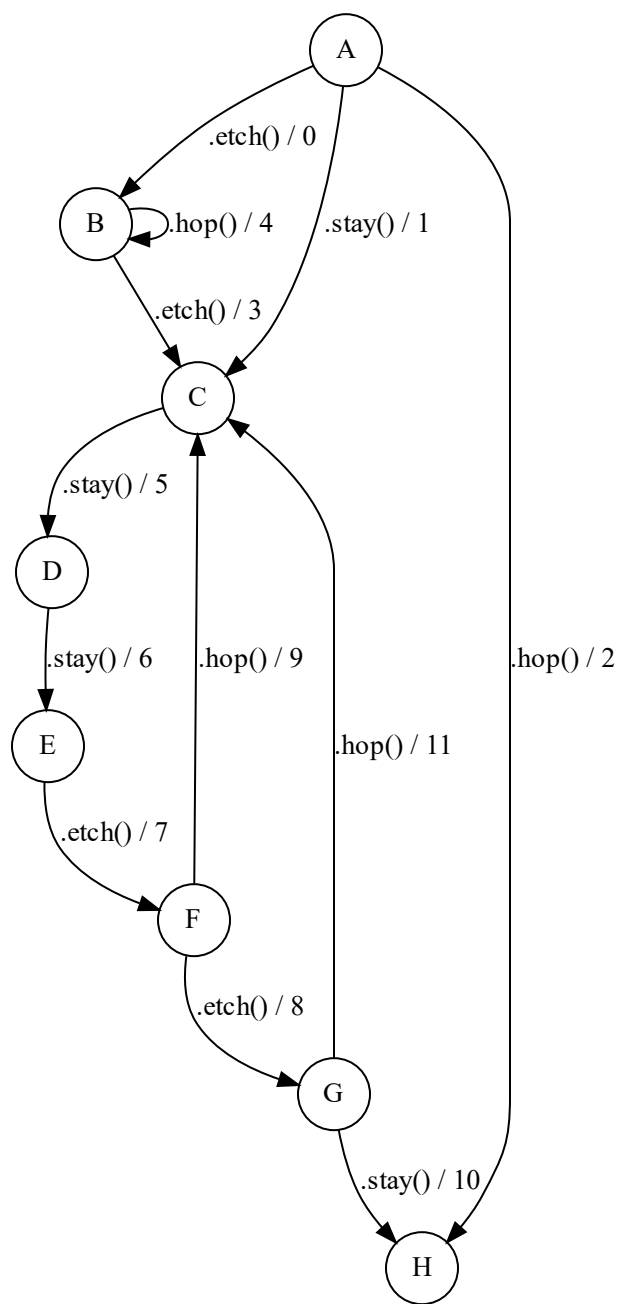
```

{'A1': 4546535912261921496,
 'A2': {'B1': [{'C1': 23898,
                 'C2': 0.8702986836433411,
                 'C3': -458472958651149437},
              {'C1': 5255,
               'C2': 0.34085899591445923,
               'C3': 1863777853167599799},
              {'C1': 53006,
               'C2': -0.20698578655719757,
               'C3': 2008811437956023259},
              {'C1': 24347,
               'C2': 0.8069355487823486,
               'C3': 7313849065400638820},
              {'C1': 41318,

```

```
'C2': -0.8661625385284424,  
'C3': -6564391726582914983}],  
'B2': {'D1': -339066992,  
      'D2': {'E1': 27800,  
            'E2': 20,  
            'E3': [42997, 37587, 48989, 58872, 64058, 38940],  
            'E4': 1099470948,  
            'E5': [121, -66, 79, -89, -104, -42, 87],  
            'E6': [-91, 62]}},  
'D3': 13521}}}
```

Задача 3.2. Реализовать конечный автомат Мили в виде класса C32. Начальным состоянием автомата является A. Методы возвращают числовые значения. Если вызываемый метод не реализован для некоторого состояния, необходимо вызвать исключение RuntimeError.



1. Пример использования класса C32:

<code>o = C32()</code>	
<code>o.etch()</code>	0
<code>o.stay()</code>	RuntimeError
<code>o.hop()</code>	4
<code>o.stay()</code>	RuntimeError
<code>o.hop()</code>	4
<code>o.etch()</code>	3
<code>o.stay()</code>	5
<code>o.stay()</code>	6
<code>o.etch()</code>	7
<code>o.etch()</code>	8
<code>o.hop()</code>	11
<code>o.stay()</code>	5
<code>o.stay()</code>	6
<code>o.etch()</code>	7
<code>o.stay()</code>	RuntimeError
<code>o.hop()</code>	9

2. Пример использования класса C32:

<code>o = C32()</code>	
<code>o.etch()</code>	0
<code>o.hop()</code>	4
<code>o.etch()</code>	3
<code>o.etch()</code>	RuntimeError
<code>o.stay()</code>	5
<code>o.stay()</code>	6
<code>o.etch()</code>	7
<code>o.hop()</code>	9
<code>o.stay()</code>	5
<code>o.stay()</code>	6
<code>o.etch()</code>	7
<code>o.stay()</code>	RuntimeError
<code>o.etch()</code>	8
<code>o.etch()</code>	RuntimeError
<code>o.hop()</code>	11

Вариант №19

Задача 3.1. Реализовать разбор двоичного формата данных (в духе формата WAD игры Doom или графического формата PNG). Данные начинаются с сигнатуры 0x47 0x52 0x52 0x71, за которой следует структура A. Порядок байт: от старшего к младшему. Адреса указаны в виде смещений от начала данных. В решении разрешено использовать модуль struct.

Структура A:	1	uint16
	2	uint16
	3	uint32
	4	Адрес (uint32) структуры B
	5	Структура E
	6	Структура F

Структура B:	1	Размер (uint16) и адрес (uint32) массива структур C
	2	float
	3	double

Структура C:	1	Адрес (uint32) структуры D
	2	double

Структура D:	1	int32
	2	int16

Структура E:	1	Размер (uint32) и адрес (uint32) массива char
	2	Массив uint8, размер 4
	3	uint64
	4	double

Структура F:	1	uint16
	2	int64
	3	int32
	4	float
	5	uint16
	6	Размер (uint16) и адрес (uint16) массива int8

Примеры разбора двоичного формата с помощью функции f31:

1. Двоичные данные:

```
(b'GRRq\xf1\xc3\x80Z\xaf\xf4\xc0\xf6\x00\x00\x00z\x00\x00\x00\x03'  
b'\x00\x00\x00\x8cB\xdb\x98\xd8&\xc4\xb6\xc3S\xee\xd8\xc1?\xec\x94\xe9'  
b'\x11eh\xa0\xa6<\xcd?\x0fJ+\xfb\x8e\x81\xcd\x19\xd6r?\x15,\xec!&'  
b'\x00\x04\x00\x8f\xc7hZ;hMEd[\xfd\xe2\x8bU\xbb8\xbc\xa8\x0e\x00\x00'  
b'\x00D\xbf\xefk\xa1\x01[\x1c\x06\x00\x00\x00J\xbf\xc2<\xe9r\xc5D\xf8\x00\x00'  
b'\x00P\xbf\xd17\xcc\xa0\xf5_X\x00\x03\x00\x00\x00V\xbf\x04\x80\xe9'  
b'\xbf\xcd\x95D*k\xa0\xf8fcn\xafK\x0e<')
```

Результат разбора:

```
{'A1': 61891,
'A2': 32858,
'A3': 2952052982,
'A4': {'B1': [{'C1': {'D1': -949462469, 'D2': 26701},
               'C2': -0.9818882967455245},
         {'C1': {'D1': 1164205053, 'D2': -7541},
          'C2': -0.14248388389618305},
         {'C1': {'D1': 1438333116, 'D2': -22514},
          'C2': -0.2690307208881584}],
'B2': -0.5175920128822327,
'B3': -0.23111774512586414},
'A5': {'E1': 'fcn',
'E2': [66, 219, 152, 216],
'E3': 2793558618919590081,
'E4': 0.8931775417106742},
'A6': {'F1': 42556,
'F2': -3657187561161650559,
'F3': -853944718,
'F4': 0.582716703414917,
'F5': 8486,
'F6': [-81, 75, 14, 60]}}
```

2. Двоичные данные:

```
(b'GRRq*%U\xd2\xca\x8f\xa5\xdf\x00\x00\x00z\x00\x00\x00\x03\x00\x00\x00\x8c'
b'\x1b;\x13\x1cQ8\xe5\xb7\x83\x89\x95\xd9\xbf\xdc\x8c\xce\xb0Hx\xe5A\x9a'
b'<ZN\xbb\xac\x90.\x80Gr?\x1fB\xbc\xcc\xa5\x00\x06\x00\x8f\xa9P\xb6\x02'
b'd?\x07\x895\x1e\xeb\xa4\xec\x00\xb1\xfa\xe7<\x00\x00\x00D?\xe1'
b'\xeb\xce\xfa\xfd\xfa4\x00\x00\x00J?\xee\xc8p\xa6\x82\xe8\x92\x00\x00'
b'\x00P\xbf\xe4\xcaU\xb2\xfd\xe8J\x00\x03\x00\x00\x00V>Q\xee[\xbf\xefb\xf4'
b"\x86V\x04Bofn'\xf1+\xab\xc7\x8b")
```

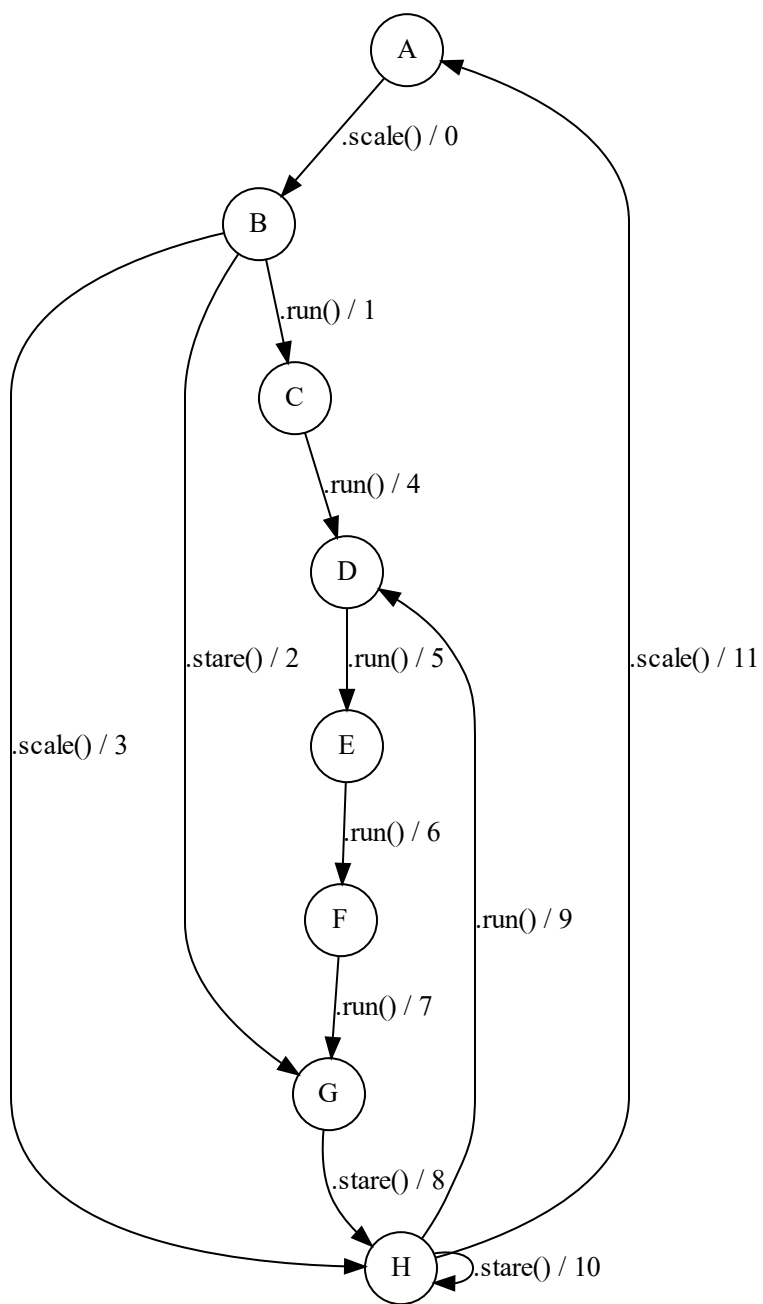
Результат разбора:

```
{'A1': 10789,
'A2': 21970,
'A3': 3398411743,
'A4': {'B1': [{'C1': {'D1': -1454328318, 'D2': 25663},
               'C2': 0.5600352194012044},
         {'C1': {'D1': 126432542, 'D2': -5212}, 'C2': 0.9619677784132981},
         {'C1': {'D1': -335498758, 'D2': -6340},
          'C2': -0.6496990676817032}],
'B2': 0.2050108164548874,
'B3': -0.9808294891119049},
```



```
'A5': {'E1': 'ofn',  
      'E2': [27, 59, 19, 28],  
      'E3': 5852680292116567513,  
      'E4': -0.444044482714848},  
'A6': {'F1': 30949,  
      'F2': 4727157117444402320,  
      'F3': 780158834,  
      'F4': 0.6221120357513428,  
      'F5': 52389,  
      'F6': [39, -15, 43, -85, -57, -117]}}
```

Задача 3.2. Реализовать конечный автомат Мили в виде класса C32. Начальным состоянием автомата является А. Методы возвращают числовые значения. Если вызываемый метод не реализован для некоторого состояния, необходимо вызвать исключение RuntimeError.



1. Пример использования класса C32:

o = C32()	
o.scale()	0
o.run()	1
o.run()	4
o.run()	5
o.run()	6
o.run()	7
o.stare()	8
o.scale()	11
o.scale()	0
o.stare()	2
o.stare()	8
o.stare()	10
o.scale()	11
o.stare()	RuntimeError
o.scale()	0
o.scale()	3
o.run()	9

2. Пример использования класса C32:

o = C32()	
o.scale()	0
o.scale()	3
o.run()	9
o.run()	5
o.run()	6
o.run()	7
o.scale()	RuntimeError
o.stare()	8
o.scale()	11
o.scale()	0
o.stare()	2
o.stare()	8
o.stare()	10
o.scale()	11
o.scale()	0
o.run()	1
o.run()	4

Вариант №20

Задача 3.1. Реализовать разбор двоичного формата данных (в духе формата WAD игры Doom или графического формата PNG). Данные начинаются с сигнатуры 0x5a 0x46 0x5a 0x65, за которой следует структура А. Порядок байт: от старшего к младшему. Адреса указаны в виде смещений от начала данных. В решении разрешено использовать модуль struct.

Структура А:	1	int64
	2	Размер (uint32) и адрес (uint32) массива адресов (uint32) структур В
	3	int8

Структура В:	1	Структура С
	2	int32
	3	Массив int8, размер 8

Структура С:	1	int32
	2	uint8
	3	Размер (uint16) и адрес (uint32) массива uint16
	4	int8
	5	Структура D

Структура D:	1	float
	2	double
	3	int32
	4	uint64
	5	int16
	6	int8
	7	Массив float, размер 4

Примеры разбора двоичного формата с помощью функции f31:

1. Двоичные данные:

```
(b'ZFZe\xd4\x9dT\x03x\x90>\xd9\x00\x00\x00\x02\x00\x00\x00\xa7\x9f\xbb=\xdd'
b'\xec\x01c\xd0dm\xc1\xf5\x00\x03\x00\x00\x00\x15\xd5\xbf0\xcc\x9b\xbf'
b'\xd5\x86Z\xb3\xf6\x00\xac\xdf\x18R\xbez\x8c"|\xb5v\x1c=A\x11,\xbc\xef'
b"\x8c\x1c\xbfQ\xe8!\xbf'Y/\xbfd\xab3#\xb6\xf8<\xb3\xd6N\xa2^xc3\x94#\xb6\xaf"
b'\xb6~(\x1f\x02\x7f\xaf3\x8c\x00\x03\x00\x00\x00^@\xbf\x08\xd9\x9c'
b'\xbf\xe9\xdb\xb1\xbf\x81\xb8r\xa4bH\xb6\xd9I7\xc5\xf6\xea\xf0\x00s\x1b\xac>'
b'\xe9\xcc\xb9\xbeJ;\x98?*xb4\xf8?Y\xc5\x95\n\xee\x90KG\xc5\x01\xb2D'
b'\x110\xb3\x00\x00\x00\x1b\x00\x00\x00d')
```

Результат разбора:

```
{'A1': -3126250192440901927,
'A2': [{'B1': {'C1': -798724671,
```

```

'C2': 245,
'C3': [47933, 56812, 355],
'C4': -43,
'C5': {'D1': -0.8117157816886902,
       'D2': -0.33632533621131233,
       'D3': -552054082,
       'D4': 8830470888382405693,
       'D5': 16657,
       'D6': 44,
       'D7': [-0.029241614043712616,
              -0.8199482560157776,
              -0.6537045836448669,
              -0.893237292766571]}},
'B2': 599193660,
'B3': [-77, -42, 78, -94, 94, -61, -108, 35]},
{'B1': {'C1': 41922355,
       'C2': 140,
       'C3': [46767, 46718, 10271],
       'C4': 64,
       'C5': {'D1': -0.5345704555511475,
              'D2': -0.8080681553235662,
              'D3': -1537062730,
              'D4': 15657106902920916992,
              'D5': 29467,
              'D6': -84,
              'D7': [0.4566400349140167,
                     -0.19749295711517334,
                     0.6668238639831543,
                     0.850671112537384]}},
'B2': 183406667,
'B3': [71, -59, 1, -78, 68, 17, 48, -77]}],
'A3': -97}

```

2. Двоичные данные:

```

(b'ZFZe\b5\b7\\ d\xa5Z\b1\x00\x00\x00\x02\x00\x00\x00\b1\b5\b81\b5\xca'
 b'\x18\x10{Y\xed\x16\xd9M\xc7\x0b\xe7\xd1\x00\x05\x00\x00\x00\x15W\xbf'
 b"6\b5\x1e?\xd7'\xe6\xe8\x19t\x84\x14\b7\xf3\xf5\xf1\xa5\x11\x06\xda"
 b'\x17\xfa\x0e\x92K\xf5\xbe\xba[\xe1>0g"=\x89J\x8b\xbeTu\xf3\x9fT6rs\x0b'
 b'me\xa0\x01E_\xbbc0c\xd81-\xfe=\xd6\x9eb\xfdY\xf8\xeaZ\x04\x00'
 b'\x06\x00\x00\x00b\xbb>\xab\xce\x98\xbf\xd36\x9a\xb2\x0e\x17T\xc2\x9e'
 b'l\x1a\x03\xad\x8b\x8a\xe6\xbd\xe3\xa3\x88V(? \x1c&\xc1?\x1fY\xf2\xbf-!'
 b'\xce\xbe\xd1)\x86\b7\b81\x15\x19X\x1b\xa8\xceR\x93 %\x00\x00\x00'
 b'\x1f\x00\x00\x00n')

```

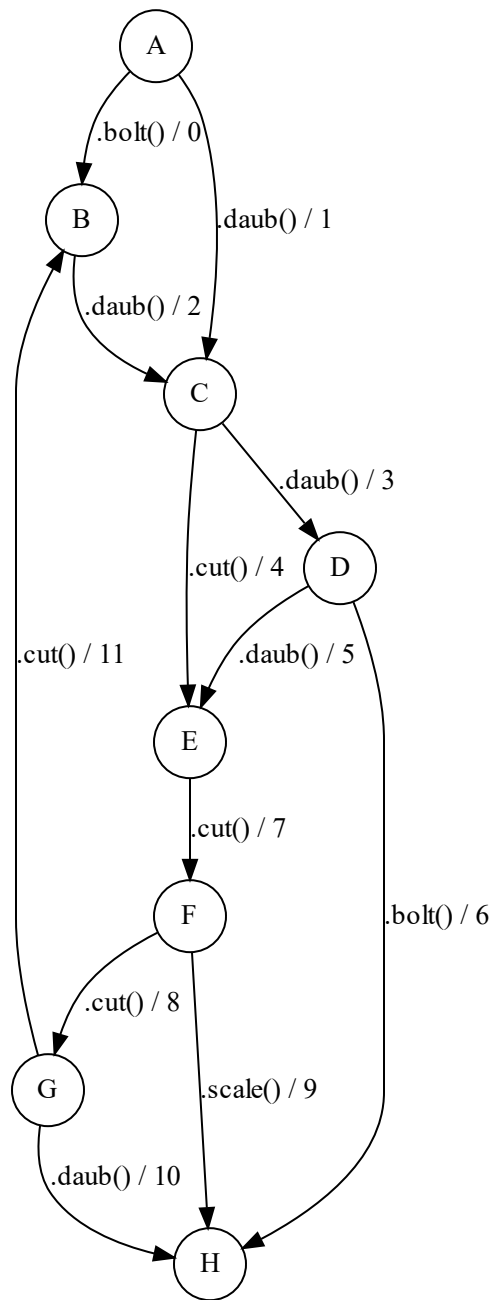
Результат разбора:

```

{'A1': -5352708337909278031,
 'A2': [{'B1': {'C1': 1304890343,
                'C2': 209,
                'C3': [33205, 51736, 4219, 23021, 5849],
                'C4': 87,
                'C5': {'D1': -0.7137011289596558,
                      'D2': 0.3618104235263859,
                      'D3': 347599861,
                      'D4': 17412342255424371214,
                      'D5': -28085,
                      'D6': -11,
                      'D7': [-0.363982230424881,
                            0.17226842045783997,
                            0.0670367106795311,
                            -0.20748119056224823]}}},
        'B2': -1621870990,
        'B3': [115, 11, 109, 101, -96, 1, 69, 95]},
 {'B1': {'C1': 1509485146,
        'C2': 4,
        'C3': [48064, 25560, 12589, 65085, 54942, 25341],
        'C4': -69,
        'C5': {'D1': 0.33556056022644043,
              'D2': -0.30020778072536314,
              'D3': -1029805030,
              'D4': 265021381777679267,
              'D5': -30634,
              'D6': 40,
              'D7': [0.6099663376808167,
                    0.6224662065505981,
                    -0.6762970685958862,
                    -0.4085199236869812]}}},
        'B2': -1216277223,
        'B3': [88, 27, -88, -50, 82, -109, 32, 37]}],
 'A3': -75}

```

Задача 3.2. Реализовать конечный автомат Мили в виде класса C32. Начальным состоянием автомата является А. Методы возвращают числовые значения. Если вызываемый метод не реализован для некоторого состояния, необходимо вызвать исключение RuntimeError.



1. Пример использования класса C32:

<code>o = C32()</code>	
<code>o.daub()</code>	1
<code>o.cut()</code>	4
<code>o.cut()</code>	7
<code>o.cut()</code>	8
<code>o.cut()</code>	11
<code>o.daub()</code>	2
<code>o.daub()</code>	3
<code>o.daub()</code>	5
<code>o.cut()</code>	7
<code>o.daub()</code>	RuntimeError
<code>o.scale()</code>	9

2. Пример использования класса C32:

<code>o = C32()</code>	
<code>o.daub()</code>	1
<code>o.daub()</code>	3
<code>o.daub()</code>	5
<code>o.cut()</code>	7
<code>o.cut()</code>	8
<code>o.cut()</code>	11
<code>o.cut()</code>	RuntimeError
<code>o.daub()</code>	2
<code>o.cut()</code>	4
<code>o.cut()</code>	7
<code>o.cut()</code>	8
<code>o.daub()</code>	10

Вариант №21

Задача 3.1. Реализовать разбор двоичного формата данных (в духе формата WAD игры Doom или графического формата PNG). Данные начинаются с сигнатуры 0x10 0x4a 0x43 0x48 0x50, за которой следует структура A. Порядок байт: от младшего к старшему. Адреса указаны в виде смещений от начала данных. В решении разрешено использовать модуль struct.

Структура A:	1	Структура B
	2	double
	3	uint16
	4	Массив char, размер 8
	5	Структура D

Структура B:	1	Массив структур C, размер 4
	2	uint8
	3	uint64
	4	float

Структура C:	1	uint8
	2	uint64

Структура D:	1	Массив int8, размер 6
	2	Массив int8, размер 3

Примеры разбора двоичного формата с помощью функции f31:

1. Двоичные данные:

```
(b'\x10JCHP\xcb\\\xb8\xb8\xa6\xff\x15\xe0\xb8\xe0\x956\xc4=\xa8\xcb\x95\x1a\xb9'
b'\x85\xa2\xdf\x96\xe4\xc5\x1c\x0eV#\xd5\xaam]\xab\xc7\x05J5\xad\x14DA('
b'\xa8\xfb\xf3\x94\xf6=\x00p\xa1\xe5|\xf1\x92\xbf\x85\xf0yuupamom\xe8\xff\x81~'
b'\xa90\x03\xe1\x08')
```

Результат разбора:

```
{'A1': {'B1': [{'C1': 203, 'C2': 13321671885519894620},
               {'C1': 224, 'C2': 1915661139967227541},
               {'C1': 185, 'C2': 1016905201453343365},
               {'C1': 86, 'C2': 416489908315280675}],
        'B2': 74,
        'B3': 18133788160387493173,
        'B4': 0.1204012855887413},
'A2': -0.01849932815126465,
'A3': 61573,
'A4': 'yuupamom',
'A5': {'D1': [-24, -1, -127, 126, -87, 48], 'D2': [3, -31, 8]}}
```

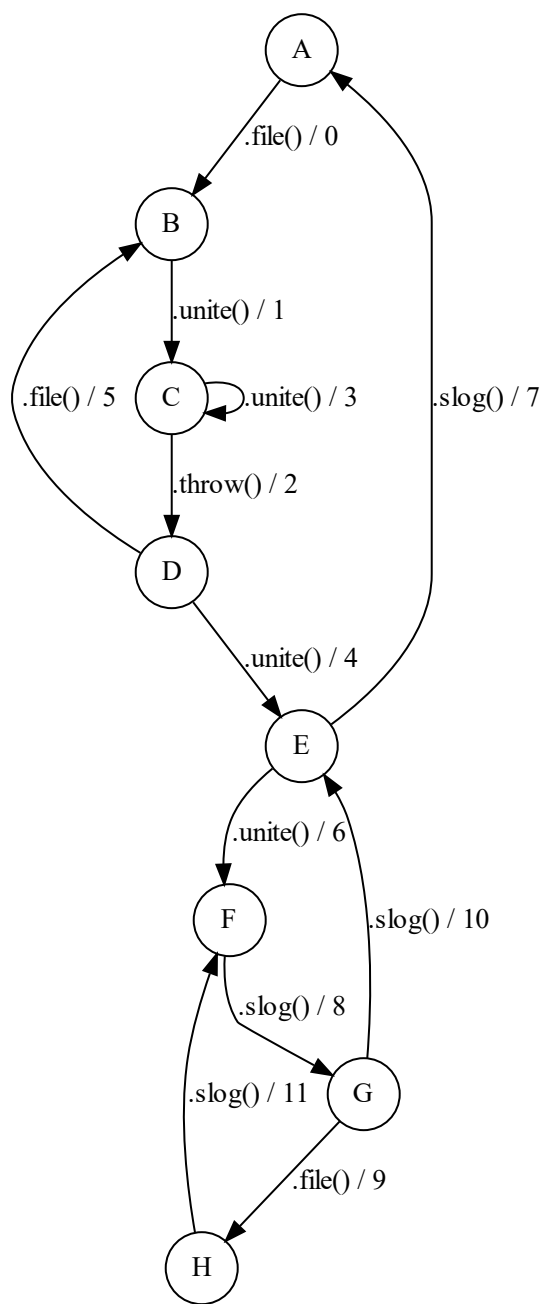
2. Двоичные данные:

```
(b"\x10JCHP\xf1;l\x9f\xc7\xd0\x15F\xb8\x9fk1\xbb\xdf\x95\xfe\n\x98\xfb\xcfE'e"
b'\x1f\xb7\x12t\x0b\xc6\xe4d\xf0M8\xfc\x18\xe0\xdbu\x97\x91`v\xda\nM\x0e'
b'\xdc>\xb0X^\x81\x96\xb6\xe9?w\x0ewclyjoamvp\xfd\xa7\xe2\xb9G\xc1\xee')
```

Результат разбора:

```
{'A1': {'B1': [{'C1': 241, 'C2': 13278324537794980923},
               {'C1': 159, 'C2': 10955848963189322091},
               {'C1': 251, 'C2': 8363948803449374159},
               {'C1': 11, 'C2': 1800375858438137030}],
        'B2': 224,
        'B3': 782067642433762779,
        'B4': 0.42979660630226135},
'A2': 0.8035385634164154,
'A3': 3703,
'A4': 'wclyjoam',
'A5': {'D1': [118, 112, -3, -89, -30, -71], 'D2': [71, -63, -18]}}
```

Задача 3.2. Реализовать конечный автомат Мили в виде класса C32. Начальным состоянием автомата является А. Методы возвращают числовые значения. Если вызываемый метод не реализован для некоторого состояния, необходимо вызвать исключение RuntimeError.



1. Пример использования класса C32:

o = C32()	
o.file()	0
o.unite()	1
o.throw()	2
o.file()	5
o.unite()	1
o.unite()	3
o.throw()	2
o.unite()	4
o.unite()	6
o.slog()	8
o.file()	9
o.slog()	11
o.slog()	8
o.slog()	10
o.slog()	7

2. Пример использования класса C32:

o = C32()	
o.file()	0
o.unite()	1
o.throw()	2
o.file()	5
o.unite()	1
o.unite()	3
o.throw()	2
o.unite()	4
o.unite()	6
o.slog()	8
o.file()	9
o.slog()	11
o.slog()	8
o.slog()	10
o.file()	RuntimeError
o.slog()	7
o.file()	0

Вариант №22

Задача 3.1. Реализовать разбор двоичного формата данных (в духе формата WAD игры Doom или графического формата PNG). Данные начинаются с сигнатуры 0xf6 0x57 0x4b 0x42, за которой следует структура A. Порядок байт: от старшего к младшему. Адреса указаны в виде смещений от начала данных. В решении разрешено использовать модуль struct.

Структура A:	1	uint8
	2	Адрес (uint16) структуры B
	3	double
	4	uint8
	5	uint16
	6	int64
	7	Структура F

Структура B:	1	int8
	2	float
	3	Массив адресов (uint16) структур C, размер 3
	4	Структура E
	5	Массив uint8, размер 3
	6	int64
	7	Размер (uint16) и адрес (uint32) массива uint32

Структура C:	1	int32
	2	Адрес (uint16) структуры D
	3	uint32

Структура D:	1	Размер (uint16) и адрес (uint32) массива int16
	2	uint8

Структура E:	1	int16
	2	int64
	3	Массив int8, размер 2
	4	uint64
	5	int32

Структура F:	1	int16
	2	float

Примеры разбора двоичного формата с помощью функции f31:

1. Двоичные данные:

```
(b'\xf6WKBt\x00g\xbf\xd5\xb9\x8a\xb5/\xea\x92\x94uxU\xd3\xf5\x08\xedxXu\x88'  
b'>7\x83\xdf\xc8{^\x15\x00\x02\x00\x00\x00 \x12-\xa0,M\x00$\x82M\xa2\x917pK'  
b'\xaa\x00\x02\x00\x00\x005\x1e\x08\x7f\xec\xb6\x009\x1e\xff\x92\xa1\xd5+'
```

```

b'\x8e\x1c\x00\x02\x00\x00\x00J.U;Y\x82\x00N\x82\xb4s0H\xd5K\x1f\x12'
b'\xe3y\xc2\xe7?\x01]\x08\x00+\x00@\x00U\xccMo)\xad\xa0\xcdi\x13$s^\xdb\x8a'
b'\x02b\x17\xe2\xaf\x9e=\x0e\xfe<\xd7\xd8\xaa\x07\x93\xa9\x05\x9a\x0b\x0b'
b'\xad\x00\x02\x00\x00\x00_')

```

Результат разбора:

```

{'A1': 116,
 'A2': {'B1': -25,
        'B2': 0.5053257942199707,
        'B3': [{'C1': 765471821,
                  'C2': {'D1': [-14213, 24085], 'D2': 18},
                  'C3': 2186125969},
                {'C1': 142601398,
                  'C2': {'D1': [14192, 19370], 'D2': 30},
                  'C3': 520065697},
                {'C1': 1429952898,
                  'C2': {'D1': [-10965, -29156], 'D2': 46},
                  'C3': 2192864048}],
        'B4': {'E1': -13235,
                  'E2': 8010124318407725860,
                  'E3': [115, 94],
                  'E4': 15819459261423071134,
                  'E5': 1024392764},
        'B5': [215, 216, 170],
        'B6': 545965821366307757,
        'B7': [1221937951, 316897730]},
 'A3': -0.33944957441386814,
 'A4': 146,
 'A5': 38005,
 'A6': 8671069706941986904,
 'A7': {'F1': 30088, 'F2': 0.1792139858007431}}

```

2. Двоичные данные:

```

(b'\xf6wKB\xce\x00i\xbf\xe2\xa7\xbd\x0e\xaf\xe2\xc2\x00\x8c\x05I\xfe'
 b'b\xd9\xd6\x93a\xd7]!\xbe\xd1$!\x1b),\x1a\x00\x02\x00\x00\x00 t\x7f'
 b'\xde\xe8\xfd\x00$\xa2\xa0\x94\xe5\xe4\x05\x9f\x00\x02\x00\x00\x005\xb3'
 b'"s\x88e\x009\xbbc\xf5\xb4\xd3\xe1L\xbe\xe4p\x00\x03\x00\x00\x00J \xcc"
 b'\xbd\xb5\xb4\x00P\xdf\xaas\t\xf0n\x05B\xf6\xf45\x89\xa4\xbf:FI\x00+'
 b"\x00@\x00W9\x99\x96\x88\x0e'VxN}I9#\x17+[j1\xd2<\x10\xed\xd2\xeb"
 b'\x8d,\xae\xdc\x9c\xf4\xf3\xb6\xd7)\xe7\x00\x02\x00\x00\x00a')

```

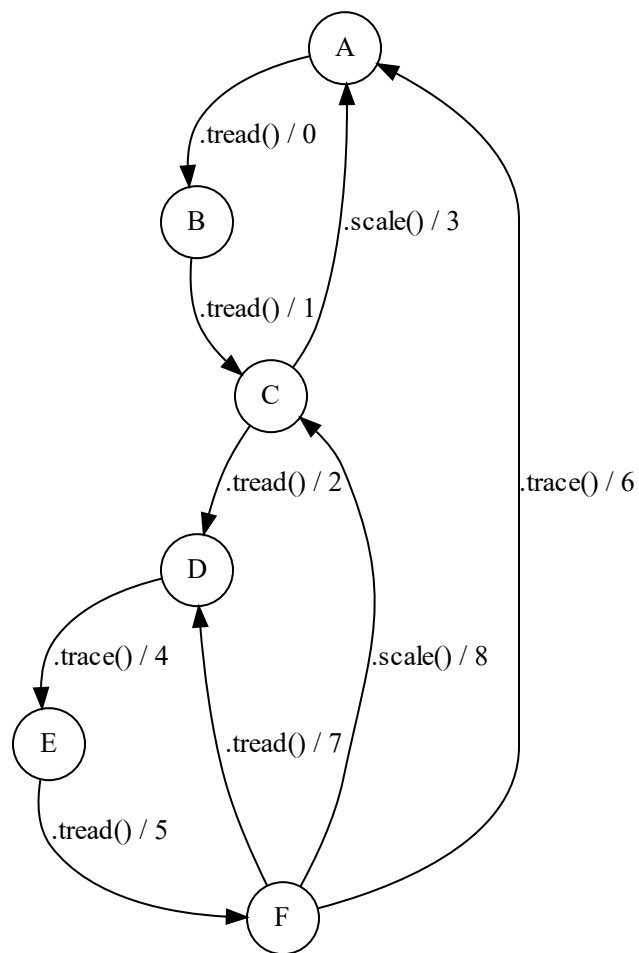
Результат разбора:

```

{'A1': 206,
 'A2': {'B1': -92,
        'B2': -0.7276349663734436,
        'B3': [{'C1': 2145315069,
                  'C2': {'D1': [6953, 11290], 'D2': 116},
                  'C3': 2728432869},
                 {'C1': 661882981,
                  'C2': {'D1': [-7163, -24721], 'D2': 179},
                  'C3': 3143890356},
                 {'C1': -859982412,
                  'C2': {'D1': [-11295, 19646, -7056], 'D2': 32},
                  'C3': 3752489737}],
        'B4': {'E1': 14745,
                  'E2': -7599808809070473603,
                  'E3': [73, 57],
                  'E4': 2528537387415491132,
                  'E5': 284021483},
        'B5': [141, 44, 174],
        'B6': -2549893961416758809,
        'B7': [4033742146, 4143199625]},
 'A3': -0.5829758917671699,
 'A4': 0,
 'A5': 35845,
 'A6': 5331807696600654295,
 'A7': {'F1': 23841, 'F2': -0.4084787666797638}}

```

Задача 3.2. Реализовать конечный автомат Мили в виде класса C32. Начальным состоянием автомата является А. Методы возвращают числовые значения. Если вызываемый метод не реализован для некоторого состояния, необходимо вызвать исключение RuntimeError.



1. Пример использования класса C32:

```

o = C32()
o.tread()      0
o.tread()      1
o.tread()      2
o.trace()       4
o.tread()      5
o.tread()      7
o.trace()       4
o.tread()      5
o.scale()       8
o.scale()       3
  
```

2. Пример использования класса C32:

o = C32()	
o.tread()	0
o.tread()	1
o.tread()	2
o.trace()	4
o.tread()	5
o.tread()	7
o.trace()	4
o.tread()	5
o.trace()	6
o.tread()	0
o.tread()	1
o.scale()	3

Вариант №23

Задача 3.1. Реализовать разбор двоичного формата данных (в духе формата WAD игры Doom или графического формата PNG). Данные начинаются с сигнатуры 0x41 0x45 0x4d 0x50, за которой следует структура A. Порядок байт: от старшего к младшему. Адреса указаны в виде смещений от начала данных. В решении разрешено использовать модуль struct.

Структура A:	1	Размер (uint16) и адрес (uint16) массива char
	2	Структура B
	3	float
	4	double
	5	uint8
	6	Адрес (uint32) структуры E
	7	Массив double, размер 5

Структура B:	1	int16
	2	Адрес (uint32) структуры C
	3	int16
	4	uint64
	5	Массив структур D, размер 3

Структура C:	1	uint8
	2	uint16
	3	double

Структура D:	1	double
	2	Массив int16, размер 5

Структура E:	1	uint8
	2	int32
	3	int64
	4	Массив uint8, размер 5
	5	uint8

Примеры разбора двоичного формата с помощью функции f31:

1. Двоичные данные:

```
(b' AEMP\x00\x03\x00\x87<\x00\x00\x00\x8a\xc6w\x1a\xc2$\xf8%\x1c?z?\xed\xac\x8f'
 b'<\xde\xbf \xcd\x00\xfei\x1b\xc5\xa0\xc8\x88\xd1?\xdf\xe3[\xbdB\xd0\xf8\nW'
 b'\xdb\n\xaf\xc5\xb6\x1a\xbb3?\xea9\xcaA\x9b\xa28w\x00\xf2\xca\x9a\x87u\xc5'
 b'\xb4\xa6>\xb40\?\xd0\xaeq\xb0\x1aYD\xd7\x00\x00\x00\x95?\xd7\xbdJ\xac'
 b'\xc6\x18D\xbf\xe9\x84N\\\xb7\x92f?\xce\xed\\\^x80P@\xbf\xdb\xa6\r\xcf'
 b';\xec\xa0?\xb6\xbe\x87\x9e|\x94\x90asf\xb7\xe5\xfb?\xec\x80\xb0_1\xd1'
 b'\xaa^\x8f\xfd\xc1MEU\xe2\xf5"\xbb/\xcf\x0c{\x0c6\xa8x')
```

Результат разбора:

```
{'A1': 'asf',
'A2': {'B1': 15480,
'B2': {'C1': 183, 'C2': 58875, 'C3': 0.8907091006536139},
'B3': -14729,
'B4': 1928144238661091194,
'B5': [{'D1': 0.9273143948430267,
'D2': [-13056, -407, 7109, -24376, -30511]},
{'D1': 0.49825185282249107,
'D2': [2647, -9462, -20539, -18918, 30899]},
{'D1': 0.8195544511227757,
'D2': [30464, -3382, -25977, 30149, -19290]}]},
'A3': 0.35193145275115967,
'A4': 0.2606472224955676,
'A5': 215,
'A6': {'E1': 94,
'E2': -1879195315,
'E3': 4996148904114991055,
'E4': [12, 123, 12, 54, 168],
'E5': 120},
'A7': [0.370928448414251,
-0.7974006472473547,
0.24161867727582553,
-0.43201012838080466,
0.08884475344946652]}
```

2. Двоичные данные:

```
(b'AEMP\x00\x05\x00\x87z\x1e\x00\x00\x00\x8c\xdf\x0f\xdf-m\x12M\xc8r|'
b'\xbf\xe6\xe6h\x0f\xc9\xb7\x1e\xe8\xa1qDw\xb1\xc7\x88\x13\xcc\xbf\xe2'
b'\xd7\xdd\xc9\xc4<\xee\xa3\xa3\xae\xec(@\x81\xa2\x05]\xbf\xc8C4\xc5\xff0`'
b'(Y,\xf4\x91v\x14\xc68\x96\xbf_\xd6?\xe2\x0ftzt\x90V\x19\x00\x00\x00\x97?'
b'\xdbY\xb6\x08A[\xcc?\xec\t\xf4r\x18\x83T?\xd3\x0c\x1a\x8f\xc4\xbe@?'
b'\xdf\x9b\xf0\x07\x95\xf5\xb8?\xe4\xd8\xf1\x19}\x19>fofnw\xf3[\xe4?\xd9e J'
b'\xaf\x1a\xecAb\xe4\x85\xdb\x1a\xa7\x01p=U\x17\xd3\x87\xf2,\xbc@\x91')
```

Результат разбора:

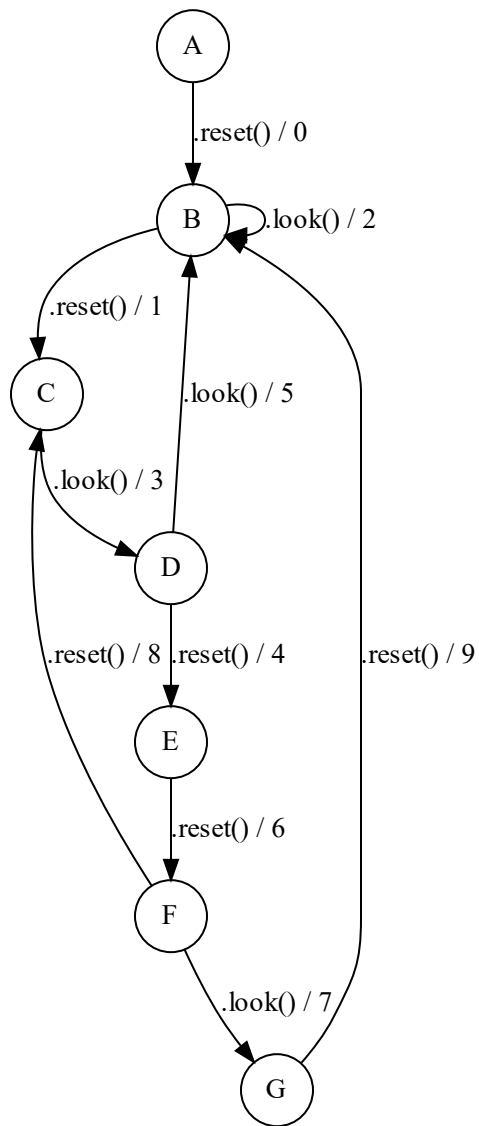
```
{'A1': 'fofnw',
'A2': {'B1': 31262,
'B2': {'C1': 243, 'C2': 23524, 'C3': 0.39679724973068065},
'B3': -8433,
'B4': 16081629769791704444,
'B5': [{'D1': -0.7156257923468023,
'D2': [-5983, 28996, 30641, -14456, 5068]},
{'D1': -0.5888508740189573,
'D2': [-23645, -20756, 10304, -32350, 1373]}]}
```

```

        {'D1': -0.18955096881711153,
         'D2': [10329, 11508, -28298, 5318, 14486]}]},
'A3': -0.8720830678939819,
'A4': 0.5643865958886731,
'A5': 25,
'A6': {'E1': 65,
       'E2': 1659143643,
       'E3': 1920505347673757651,
       'E4': [135, 242, 44, 188, 64],
       'E5': 145},
'A7': [0.4273505287760002,
       0.8762151935990041,
       0.29761375465293227,
       0.4938926767425431,
       0.6514821527314678]}

```

Задача 3.2. Реализовать конечный автомат Мили в виде класса C32. Начальным состоянием автомата является А. Методы возвращают числовые значения. Если вызываемый метод не реализован для некоторого состояния, необходимо вызвать исключение RuntimeError.



1. Пример использования класса C32:

o = C32()	
o.reset()	0
o.look()	2
o.reset()	1
o.look()	3
o.reset()	4
o.look()	RuntimeError
o.reset()	6
o.look()	7
o.reset()	9
o.reset()	1
o.look()	3
o.look()	5

2. Пример использования класса C32:

o = C32()	
o.reset()	0
o.look()	2
o.reset()	1
o.look()	3
o.reset()	4
o.reset()	6
o.look()	7
o.reset()	9
o.reset()	1
o.look()	3
o.look()	5
o.look()	2

Вариант №24

Задача 3.1. Реализовать разбор двоичного формата данных (в духе формата WAD игры Doom или графического формата PNG). Данные начинаются с сигнатуры 0x4b 0x58 0x54 0x59 0x3e, за которой следует структура A. Порядок байт: от старшего к младшему. Адреса указаны в виде смещений от начала данных. В решении разрешено использовать модуль struct.

Структура A:	1	int8
	2	Массив char, размер 8
	3	Структура B
	4	uint32
	5	Массив адресов (uint32) структур C, размер 2
	6	Размер (uint16) и адрес (uint16) массива char
	7	Адрес (uint16) структуры D

Структура B:	1	int16
	2	uint8

Структура C:	1	uint8
	2	int32
	3	int16
	4	uint32
	5	uint32
	6	int64
	7	int8

Структура D:	1	float
	2	Массив double, размер 3
	3	uint8
	4	int8

Примеры разбора двоичного формата с помощью функции f31:

1. Двоичные данные:

```
(b'KXTY>{aaybxzyt\x00\xe5\xb8\x97\xd0\xe1\xa6\x00\x00\x00#\x00\x00\x00'
b';\x00\x02\x00S\x00U0\xcc\xa2\xaa9\xe94S\xcc\x15\xd4\xc2\x12\x88v\x80Y'
b'm\xe7r\x1c\xdbN4#\xf0%\x19>\x91\x9e+[\xfe\xb8\xaa\xe7\x19\xff\x8d\xff'
b'\xf5\x99L,\xf8\xcb\x08zc\xbf\x1d[\x88\xbf\xe9\xcd>\xc8o\x83\xde\xbf\xe3L'
b'\xa1\xbc\xcbYt?\xce\x8d\xa0\xaf\xe4\xf80L\xf1')
```

Результат разбора:

```
{'A1': 123,
'A2': 'aaybxzyt',
'A3': {'B1': 229, 'B2': 184},
```

```

'A4': 2547048870,
'A5': [{ 'C1': 79,
         'C2': -861754823,
         'C3': -5836,
         'C4': 1405883860,
         'C5': 3255994486,
         'C6': -9198199923108160690,
         'C7': 52},
      { 'C1': 35,
        'C2': -266004162,
        'C3': -28258,
        'C4': 727449272,
        'C5': 2867272191,
        'C6': -8214577156543678261,
        'C7': 8}],
'A6': 'zc',
'A7': { 'D1': -0.614677906036377,
        'D2': [-0.8063043513067176, -0.6031044661893206, 0.23869713390917147],
        'D3': 76,
        'D4': -15}}

```

2. Двоичные данные:

```

(b'KXTY>\x80gubccjht\xc6\xa1\xe6\xc9\xdc\x1d\xba\x00\x00\x00#\x00\x00\x00'
 b';\x00\x02\x00S\x00U\xac\xce\xf4\xe0\xd1~i\x98}\x0cd\x14\xac k\x81\x91'
 b'\xe5@\xfj\xaf8dV\xd1\xc5\xd0\xee*\x15\xb7f\x00\xaf\xea\xea0\xed\xd6C'
 b'\xe2M\xe3\x12\xf61Yse?\x03\xe8\xf5?\xee\xb1}\x1a\xf6\xea\xac\xbf\xe4\xb7'
 b'\xe8.\xc2\x04\xa8?\xcd\xbe\x8f!\xc3\xb5\x08]'')

```

Результат разбора:

```

{ 'A1': -128,
  'A2': 'gubccjht',
  'A3': { 'B1': -14687, 'B2': 230},
  'A4': 3386645946,
  'A5': [{ 'C1': 172,
           'C2': -822812463,
           'C3': 32361,
           'C4': 2558332004,
           'C5': 346824811,
           'C6': -9110248507894485192,
           'C7': 100},
        { 'C1': 86,
          'C2': -775565074,
          'C3': 10773,
          'C4': 3076915375,

```

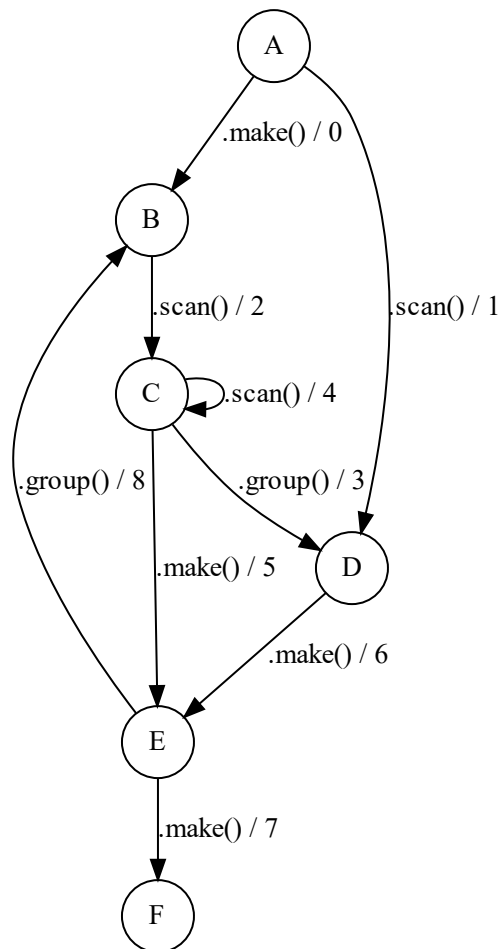


```

'C5': 3941216493,
'C6': -3007311302003329487,
'C7': 89}],
'A6': 'se',
'A7': {'D1': 0.51527339220047,
'D2': [0.9591661001826046, -0.6474495805015481, 0.23237790249496748],
'D3': 93,
'D4': 34}}

```

Задача 3.2. Реализовать конечный автомат Мили в виде класса C32. Начальным состоянием автомата является A. Методы возвращают числовые значения. Если вызываемый метод не реализован для некоторого состояния, необходимо вызвать исключение RuntimeError.



1. Пример использования класса C32:

```

o = C32()
o.make()          0
o.scan()          2
o.scan()          4
o.scan()          4
o.group()         3
o.make()          6
o.group()         8
o.scan()          2
o.make()          5
o.make()          7

```

2. Пример использования класса C32:

```

o = C32()
o.scan()          1
o.make()          6
o.group()         8
o.make()          RuntimeError
o.scan()          2
o.scan()          4
o.group()         3
o.make()          6
o.group()         8
o.scan()          2
o.make()          5
o.make()          7

```

Вариант №25

Задача 3.1. Реализовать разбор двоичного формата данных (в духе формата WAD игры Doom или графического формата PNG). Данные начинаются с сигнатуры 0x50 0x50 0x45 0x4f, за которой следует структура A. Порядок байт: от старшего к младшему. Адреса указаны в виде смещений от начала данных. В решении разрешено использовать модуль struct.

Структура A:	1	uint64
	2	Массив адресов (uint32) структур B, размер 4
	3	Адрес (uint16) структуры D
	4	Массив char, размер 7
	5	uint64
	6	Массив char, размер 8
	7	Массив uint16, размер 2
	8	Размер (uint16) и адрес (uint16) массива int16

Структура B:	1	Адрес (uint16) структуры C
	2	int16
	3	int8

Структура C:	1	uint8
	2	int32
	3	float

Структура D:	1	float
	2	uint32
	3	uint16
	4	float
	5	int32
	6	double
	7	uint8
	8	double

Примеры разбора двоичного формата с помощью функции f31:

1. Двоичные данные:

```
(b'PPE0\xc4H0\x8c\xa9\xab_a\x00\x00\x00F\x00\x00\x00T\x00\x00\x00b\x00\x00\x00p'
b'\x00urysgdwd"(\xc7\x86Gwuvnmjhubyfx\xc6\xf1\xca\x00\x07\x00\x98\xbczu'
b'\x80V?\x1b-\x06\x00=4\x18A\xba\xb7\x08q\xd6\xbfX\x05\x16\x00K\x14:'
b'\xcF\xbbS\xa5\xf6C?=\xf9\xcd\x00Y\x94V`#\xed\xbb\xa3\xa3>\xbbc\xe5\x00gB\x86'
b'\x16\xbe\xa1c?\xf5\x13ns\xab\x8f\xbd\xa6\x11-\xee\xc4}\xc8\xbf\xe1\x95\xe6/'
b'\xdc\xfd\x80n\xbf\xeeP\x0b\x83\x03^65\x9f\xb3Ix\x11k\xda\x8c\xnb5\x92\xdew')
```

Результат разбора:

```
{'A1': 14143608010457177953,
'A2': [{'B1': {'C1': 188, 'C2': 2054520918, 'C3': 0.6061557531356812},
        'B2': 13336,
        'B3': 65},
        {'B1': {'C1': 186, 'C2': -1224183338, 'C3': -0.8438276052474976},
        'B2': 5178,
        'B3': -49},
        {'B1': {'C1': 187, 'C2': 1940256323, 'C3': 0.7420929074287415},
        'B2': -27562,
        'B3': 96},
        {'B1': {'C1': 35, 'C2': -306469981, 'C3': 0.36599650979042053},
        'B2': 17030,
        'B3': 22}],
'A3': {'D1': -0.3152103126049042,
        'D2': 4111691379,
        'D3': 43919,
        'D4': -0.08108744770288467,
        'D5': -289112632,
        'D6': -0.5495482382611243,
        'D7': 110,
        'D8': -0.9472711142397603},
'A4': 'rysgdwd',
'A5': 2461436575894435190,
'A6': 'nmjhubyf',
'A7': [30918, 61898],
'A8': [13727, -19639, 30737, 27610, -29586, -19054, -8585]}
```

2. Двоичные данные:

```
(b'PPE00\x1f\xd7\xa2R\x7f%W\x00\x00\x00F\x00\x00\x00T\x00\x00\x00b\x00\x00\x00p'
b'\x00usnzlyrsq\xa6\xfb\x90\xc8\xdcq\xfdtussxbdvhc\x5s\x00\x07\x00'
b'\x980\xeb\xa1\xc30?2\xa0\x84\x00=HQB\xbc\x88\x1b\xvc3?]\x15\xb0\x00K\x13r'
b"\x9a0&\xe8\x0e\xc6\xbfB.\x1b\x00Y\x9f\xa8\xfaE'\x00s\x02>\xc0H\x9f\x00g|\xe4"
b'\xf3?4\x91onm\x12iY0\xbf8\xb7\x83Wc2\x04\xbf\xd7|1?\x0c\xeb\xd0k\xbf\xd1,Q'
b'\x8fH\xe1\x88\xd5\xfb\xbaQh5\x89\xa8\xcf\xb8iZ\x12\x89')
```

Результат разбора:

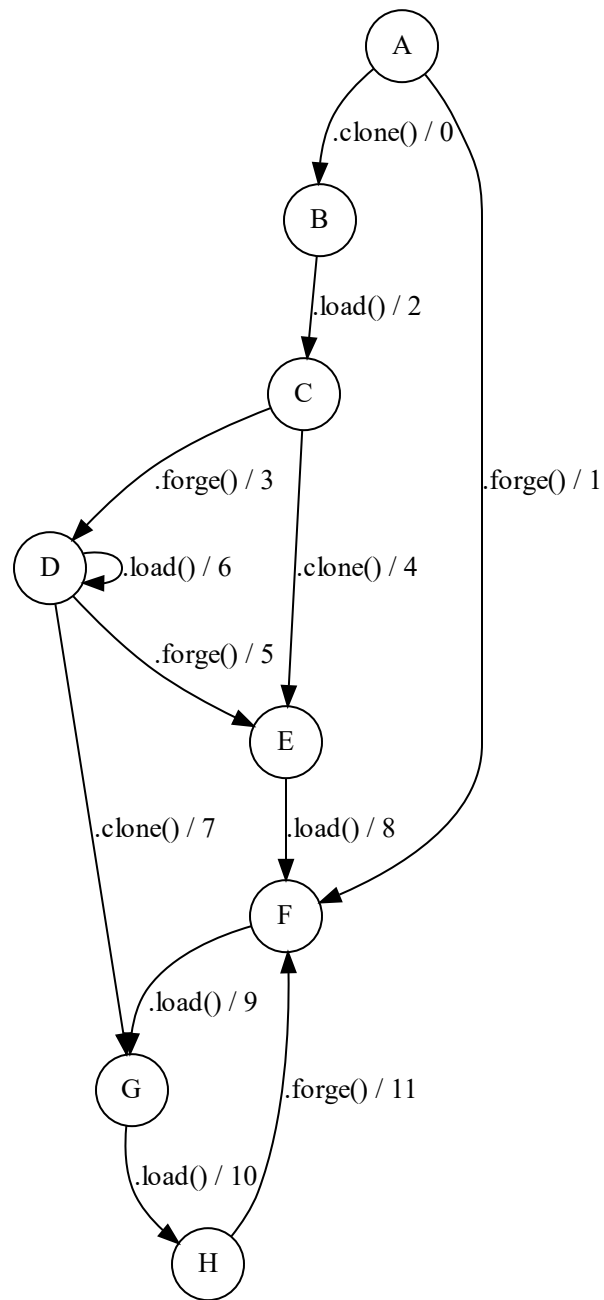
```
{'A1': 3467727330267309399,
'A2': [{'B1': {'C1': 79, 'C2': -341720272, 'C3': 0.6977617740631104},
        'B2': 18513,
        'B3': 66},
        {'B1': {'C1': 188, 'C2': -2011466045, 'C3': 0.863612174987793},
        'B2': 4978,
        'B3': -102},
        {'B1': {'C1': 48, 'C2': 652742342, 'C3': -0.7585160136222839},
```

```

        'B2': -24664,
        'B3': -6},
    {'B1': {'C1': 69, 'C2': 654340866, 'C3': 0.37555405497550964},
     'B2': 31972,
     'B3': -13}],
'A3': {'D1': 0.7053441405296326,
      'D2': 1852641897,
      'D3': 22863,
      'D4': -0.7215501666069031,
      'D5': 1466118660,
      'D6': -0.366955100605705,
      'D7': 107,
      'D8': -0.26832999222417486},
'A4': 'snzlyrs',
'A5': 8189509571683578365,
'A6': 'tussxbdv',
'A7': [26723, 62835],
'A8': [-10757, -17839, 26677, -30296, -12360, 26970, 4745]}

```

Задача 3.2. Реализовать конечный автомат Мили в виде класса C32. Начальным состоянием автомата является A. Методы возвращают числовые значения. Если вызываемый метод не реализован для некоторого состояния, необходимо вызвать исключение RuntimeError.



1. Пример использования класса C32:

<code>o = C32()</code>	
<code>o.clone()</code>	0
<code>o.clone()</code>	RuntimeError
<code>o.load()</code>	2
<code>o.forge()</code>	3
<code>o.load()</code>	6
<code>o.forge()</code>	5
<code>o.load()</code>	8
<code>o.clone()</code>	RuntimeError
<code>o.load()</code>	9
<code>o.load()</code>	10
<code>o.forge()</code>	11

2. Пример использования класса C32:

<code>o = C32()</code>	
<code>o.load()</code>	RuntimeError
<code>o.clone()</code>	0
<code>o.load()</code>	2
<code>o.forge()</code>	3
<code>o.load()</code>	6
<code>o.forge()</code>	5
<code>o.forge()</code>	RuntimeError
<code>o.load()</code>	8
<code>o.load()</code>	9
<code>o.load()</code>	10
<code>o.load()</code>	RuntimeError
<code>o.forge()</code>	11
<code>o.load()</code>	9
<code>o.load()</code>	10
<code>o.forge()</code>	11

Вариант №26

Задача 3.1. Реализовать разбор двоичного формата данных (в духе формата WAD игры Doom или графического формата PNG). Данные начинаются с сигнатуры 0x49 0x46 0x46 0x9d, за которой следует структура A. Порядок байт: от старшего к младшему. Адреса указаны в виде смещений от начала данных. В решении разрешено использовать модуль struct.

Структура A:	1	Адрес (uint16) структуры B
	2	Адрес (uint16) структуры F
	3	int8

Структура B:	1	Адрес (uint16) структуры C
	2	Массив структур E, размер 4

Структура C:	1	uint8
	2	uint8
	3	int16
	4	int32
	5	int8
	6	int64
	7	Структура D

Структура D:	1	uint8
	2	Размер (uint32) и адрес (uint16) массива float
	3	uint64

Структура E:	1	float
	2	Массив int16, размер 5
	3	uint32

Структура F:	1	Размер (uint32) и адрес (uint16) массива float
	2	int8

Примеры разбора двоичного формата с помощью функции f31:

1. Двоичные данные:

```
(b'IFF\x9d\x00\x00\x8b\xf2>\xcdX\xdd\xbff\xd4\xdb\xbe\x03!\x80=\xc2\xfd'
b'\xf0a\xcd4\xe3\x91\x0e\x14\x04\xccI\xda\x1d\x86G\xa5r\xc14\x00'
b'\x00\x00\x04\x00\tY\x1fKL\xbedm\xe6\x00\x19?\x17\x06\xd5\x88\xb3\xd3s\xd7'
b'p\xea\x16\xc2\xd9\x05\tGm\xbf\x0c1b\xaa\xeb? \xc9&j\x8f\xebW\xd8'
b'\xbf\x17\x1b\xbe\xa02"M?\xa3\x08\x8c\xba\xab\xd8 \xfc\xfc\xae\xc3'
b'\x90\xbe\xdf\xf2\x1d\x0b\xf0\xad\x85v\x81:\x18\xd3\xf4\x9c\x99\x890='
b'\xe1u\xe3>\xd7,\xca\x00\x00\x00\x02\x00\x83b')
```

Результат разбора:


```
{'A1': {'B1': {'C1': 97,
               'C2': 205,
               'C3': 13539,
               'C4': -1861348348,
               'C5': -52,
               'C6': 5321598372256510657,
               'C7': {'D1': 52,
                      'D2': [0.4010685980319977,
                              -0.9016854166984558,
                              -0.12805747985839844,
                              0.09521090984344482],
                      'D3': 6421934386637467110}},
        'B2': [{'E1': 0.5899479985237122,
                  'E2': [-30541, -11405, -10384, -5610, -15655],
                  'E3': 84494189},
                {'E1': -0.548528790473938,
                  'E2': [-21781, 16160, -14042, 27279, -5289],
                  'E3': 3636401947},
                {'E1': -0.3128824830055237,
                  'E2': [19775, -23800, -29510, -21544, 8444],
                  'E3': 4138648464},
                {'E1': -0.43739405274391174,
                  'E2': [3056, -21115, 30337, 14872, -11276],
                  'E3': 2627307824}]},
        'A2': {'F1': [0.11008813232183456, 0.4202635884284973], 'F2': 98},
        'A3': -14}
```

2. Двоичные данные:

```
(b' IFF\x9d\x005\x00\x87>>\x85\xd4\x99>\xb3\x87\x98>\x91\x83\xa7\x8b\xcb|H^\x8ab'
b'\x87\xb0\xde\x14\xe05\x88\xa6|\x17\x13\x00\x00\x00\x03\x00\tml\x9b~'
b"\xa9\xd1\xbbX\xf9\x00\x15?\x1bv\xb4\x06gG\xe5]'\x88@\x0e?Y\x15N"
b'\xe3\xbd\xc9\xfb~"*xa8PN\x06\x9f\x08\xa0\xd3\xf4\xfc8\xb7?3\xcf\xab\x04'
b'\x02\x14#\t(.\x0b\<\xeem\xd6?lM#{\xf0\xa6\xfb0\xc2\xcb)\xf5\x9d'
b'\x8a\xe7\x88\xbf\x1c.\x1f\xbe6\xb5\xd7\x00\x00\x00\x02\x00\x7f^')
```

Результат разбора:

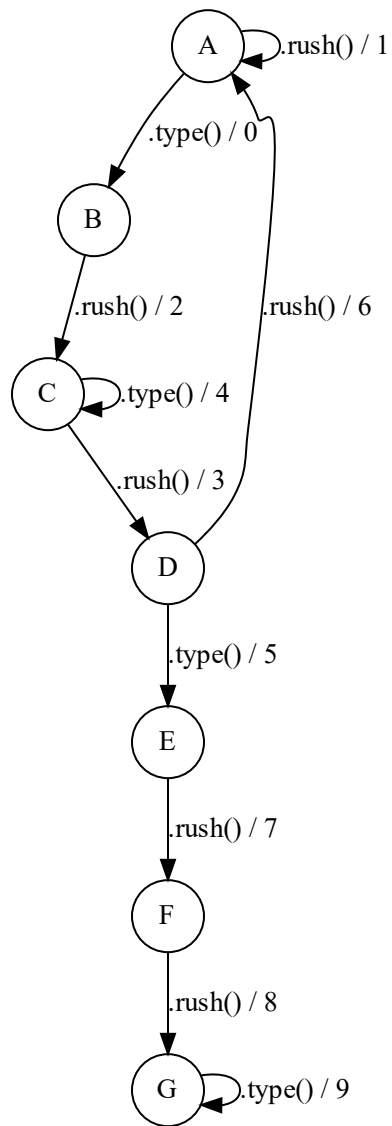
```
{'A1': {'B1': {'C1': 139,
               'C2': 203,
               'C3': 31816,
               'C4': 1586127495,
               'C5': -80,
               'C6': -2444082048375817193,
               'C7': {'D1': 19,
                      'D2': [0.2613876163959503,
```

```

        0.3506438732147217,
        0.2842075526714325],
        'D3': 7898045639357585657}}},
    'B2': [{ 'E1': 0.6072800159454346,
              'E2': [1639, 18405, 23847, -30656, 3647],
              'E3': 1494568675},
            { 'E1': -0.09862421452999115,
              'E2': [8746, -22448, 19974, -24824, -24365],
              'E3': 4110170295},
            { 'E1': 0.7023875117301941,
              'E2': [1026, 5155, 2344, 11837, 2908],
              'E3': 1022258646},
            { 'E1': 0.9230520129203796,
              'E2': [31728, -22789, 28610, -13447, 10741],
              'E3': 2643126152}}]},
    'A2': { 'F1': [-0.610078752040863, -0.1784280389547348], 'F2': 94},
    'A3': 62}

```

Задача 3.2. Реализовать конечный автомат Мили в виде класса C32. Начальным состоянием автомата является A. Методы возвращают числовые значения. Если вызываемый метод не реализован для некоторого состояния, необходимо вызвать исключение RuntimeError.



1. Пример использования класса C32:

```

o = C32()
o.type()          0
o.type()          RuntimeError
o.rush()          2
o.rush()          3
o.rush()          6
o.rush()          1
o.type()          0
o.rush()          2
o.type()          4
o.rush()          3
o.type()          5
o.rush()          7
o.rush()          8
o.type()          9

```

2. Пример использования класса C32:

```

o = C32()
o.type()          0
o.rush()          2
o.type()          4
o.rush()          3
o.rush()          6
o.rush()          1
o.type()          0
o.rush()          2
o.type()          4
o.rush()          3
o.type()          5
o.rush()          7
o.rush()          8
o.type()          9

```

Вариант №27

Задача 3.1. Реализовать разбор двоичного формата данных (в духе формата WAD игры Doom или графического формата PNG). Данные начинаются с сигнатуры 0x41 0x44 0x5a 0xce, за которой следует структура A. Порядок байт: от старшего к младшему. Адреса указаны в виде смещений от начала данных. В решении разрешено использовать модуль struct.

Структура A:	1	Структура B
	2	Массив структур C, размер 2
	3	Массив char, размер 8
	4	Адрес (uint16) структуры D
	5	uint8
	6	Массив uint16, размер 7

Структура B:	1	uint16
	2	int16
	3	double
	4	uint64
	5	uint8

Структура C:	1	Массив char, размер 5
	2	double
	3	int16

Структура D:	1	uint8
	2	Размер (uint32) и адрес (uint16) массива float

Примеры разбора двоичного формата с помощью функции f31:

1. Двоичные данные:

```
(b'ADZ\xceP\xc1\x01c?\xdc&\xa5*h\xf8\xe8\x13\xde\xc7\xd5\x94#\xcf\x81\xd3reb'  
b'qa\xbf\xdc\x0f\xb2\xafH\x99|mzobxny\xbf\xe9yEt\xbbI\x88\xe4\x85nyaxzbjh\x00'  
b'XL\x81\x9f\xe1\xd9!\xcc\x96\xde\xf1 \t\xc8?\xcb?7\xe27=w\xd0$'  
b'\xb2\x00\x00\x00\x02\x00P')
```

Результат разбора:

```
{'A1': {'B1': 20673,  
        'B2': 355,  
        'B3': 0.43985871450048686,  
        'B4': 1431801451677732737,  
        'B5': 211},  
 'A2': [{'C1': 'rebqa', 'C2': -0.43845812909639625, 'C3': 28026},  
        {'C1': 'obxny', 'C2': -0.7960536270059473, 'C3': -7035}],  
 'A3': 'nyaxzbjh',
```

```
'A4': {'D1': 178, 'D2': [0.7182955145835876, 0.060501232743263245]},
'A5': 76,
'A6': [33183, 57817, 8652, 38622, 61728, 2504, 16331]}
```

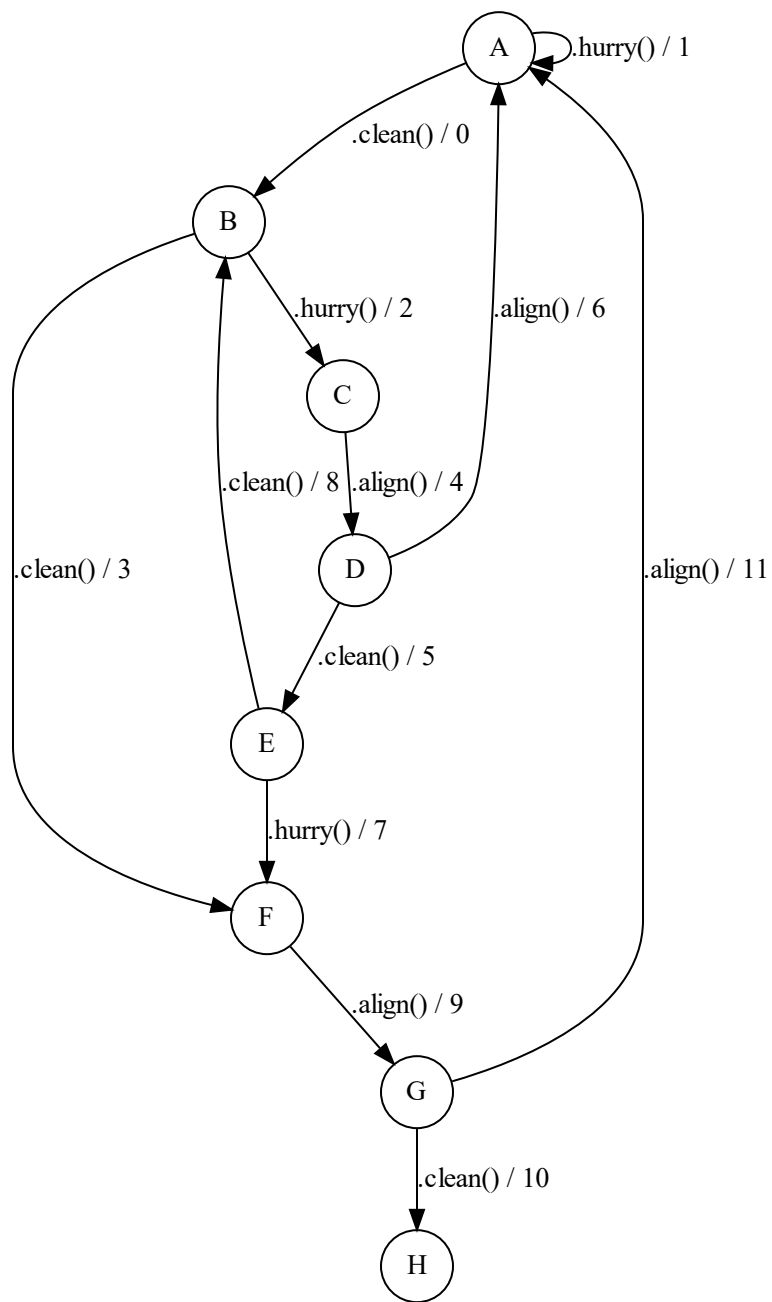
2. Двоичные данные:

```
(b'ADZ\xceMc\x0b\xc9?\xcam*G\nv\x18c\x15\xca\x9ca\xcd\xf2\x01Ekdirh?\xc9\xb4`#M'
b'B\x00\x7fZuozbk?\xe2\xf50\xee\x9c\xeb\xe4\xaf\xf3llsjawu\x00`\xf3\xfex'
b"\xdd\xb8\xd0s\x1e\x12\xe1\xbb\xa4V[b>\x02\xc0\xbe\xbe\xeb\xfd\xd3\xbe\xe5'J"
b'=\xdf[s\x05\x00\x00\x00\x04\x00P')
```

Результат разбора:

```
{'A1': {'B1': 19811,
        'B2': 3017,
        'B3': 0.20645645588676653,
        'B4': 7139835557270385153,
        'B5': 69},
'A2': [{'C1': 'kdirh', 'C2': 0.20081712459297307, 'C3': 32602},
        {'C1': 'uozbk', 'C2': 0.5924453411492974, 'C3': -20493}],
'A3': 'llsjawu',
'A4': {'D1': 5,
        'D2': [0.12768837809562683,
                -0.46092090010643005,
                -0.44756537675857544,
                0.10906114429235458]},
'A5': 243,
'A6': [65144, 56760, 53363, 7698, 57787, 42070, 23394]}
```

Задача 3.2. Реализовать конечный автомат Мили в виде класса C32. Начальным состоянием автомата является А. Методы возвращают числовые значения. Если вызываемый метод не реализован для некоторого состояния, необходимо вызвать исключение RuntimeError.



1. Пример использования класса C32:

o = C32()	
o.hurry()	1
o.clean()	0
o.hurry()	2
o.align()	4
o.clean()	5
o.hurry()	7
o.clean()	RuntimeError
o.align()	9
o.align()	11
o.clean()	0
o.hurry()	2
o.hurry()	RuntimeError
o.align()	4
o.align()	6
o.align()	RuntimeError
o.clean()	0
o.clean()	3
o.align()	9
o.hurry()	RuntimeError
o.clean()	10

2. Пример использования класса C32:

o = C32()	
o.clean()	0
o.hurry()	2
o.hurry()	RuntimeError
o.align()	4
o.clean()	5
o.hurry()	7
o.align()	9
o.align()	11
o.hurry()	1
o.clean()	0
o.align()	RuntimeError
o.clean()	3
o.align()	9
o.clean()	10

Вариант №28

Задача 3.1. Реализовать разбор двоичного формата данных (в духе формата WAD игры Doom или графического формата PNG). Данные начинаются с сигнатуры 0x73 0x53 0x4e 0x52 0x53, за которой следует структура A. Порядок байт: от старшего к младшему. Адреса указаны в виде смещений от начала данных. В решении разрешено использовать модуль struct.

Структура A:	1	Адрес (uint32) структуры B
	2	int32
	3	Массив структур D, размер 4

Структура B:	1	uint8
	2	float
	3	Структура C
	4	Массив char, размер 6

Структура C:	1	Размер (uint16) и адрес (uint16) массива uint16
	2	uint16

Структура D:	1	int64
	2	Размер (uint16) и адрес (uint16) массива int16
	3	int16
	4	int16
	5	uint32
	6	uint16

Примеры разбора двоичного формата с помощью функции f31:

1. Двоичные данные:

```
(b'sSNRS\x00\x00i\tF\xaf\xe8\xb0Kv\xcd\x07\x9a\xb5\xfb\x00\x04\x00'
b'z\xce\x19=q+\x8b\x1c=\xde\x08\xa6<\x7f\xf6i\xe8Hy\x00\x03\x00\x82\xe'
b'\xb5,\x8f\x0e\xc2P\xf8\xec\xa9qM\xd3\xe9\xffJ9\xb8\x00\x03\x00\x88R\x05j'
b"\xc6=\xd8\x852\x8d\x02\x8c\xd7&' '\xa4+U\x00\x04\x00\x8e\x08\x8c\xa2\xb6\xbb"
b'\x8a\xbet\x1e\x04=\xe9\xe6\x83\xbe>\xc68\x97\x00\x02\x00e\xcb\xa9dyei'
b'db\xd2\x05\x0bwm\x92,\x8c\xb5\xc7\x0c\x18\xcfZ1\xfa\x17Q\x17W\xd0"\xcbH=\x03'
b'yc')
```

Результат разбора:

```
{ 'A1': { 'B1': 190,
          'B2': 0.38715049624443054,
          'B3': { 'C1': [15849, 59011], 'C2': 52137 },
          'B4': 'dyeidb' },
  'A2': 155627496,
  'A3': [ { 'D1': -5743366276812982789,
```

```

'D2': [-11771, 2935, 28050, 11404],
'D3': -12775,
'D4': 15729,
'D5': 730537021,
'D6': 56840},
{'D1': -6468154268495361927,
'D2': [-19001, 3096, -12454],
'D3': 3765,
'D4': 11407,
'D5': 247615736,
'D6': 60585},
{'D1': 8164414701456472504,
'D2': [27898, 5969, 5975],
'D3': 20997,
'D4': 27334,
'D5': 1037600050,
'D6': 36098},
{'D1': -8298121838796199083,
'D2': [-12254, -13496, 15619, 31075],
'D3': 2188,
'D4': -23882,
'D5': 3146432116,
'D6': 7684}}}]

```

2. Двоичные данные:

```

(b'sSNRS\x00\x00\x00oW{\xa9\r\r\xbe\xe0\x19\xae_\xff`\x00\x04\x00\x80\xfd\x03W'
b'#\x91\xd3\x1d\xfcH\xa7\xec)\xdc\xd4\xa6z\x81\x1b\x00\x03\x00\x88]N\x8f.\xc2'
b'\x91\x94\xa7\x8e\xca1Q\x9e\x8e\xfb\xb7\xab\xd8\x00\x03\x00\x8e\x1d\xb7\xbb'
b'\x0fC\xb7\xc7\xeab\x94U\xd0\xaaaf5j0_\x00\x02\x00\x94\xae\xcd\x94*\x97'
b'l p\x92\xe4\xdeQ\xd4\xaf\x80>\xc4r\xe3E\x11i\xbd i0"\x00\x05\x00e\xb4\xe1h'
b'gzfpN5A\xd3\xc4\xfd\xa6\xbb\x03=\xea\xf3\x89Z\x00jWI\xef\xa8J\xcb\xd4\n')

```

Результат разбора:

```

{'A1': {'B1': 105,
'B2': -0.05693066865205765,
'B3': {'C1': [20948, 44928, 16068, 29411, 17681], 'C2': 46222},
'B4': 'lhgzfp'},
'A2': 1467721997,
'A3': [{'D1': 990475368972418912,
'D2': [20021, 16851, -15107, -22853],
'D3': -765,
'D4': 22307,
'D5': 2446532092,
'D6': 18599},

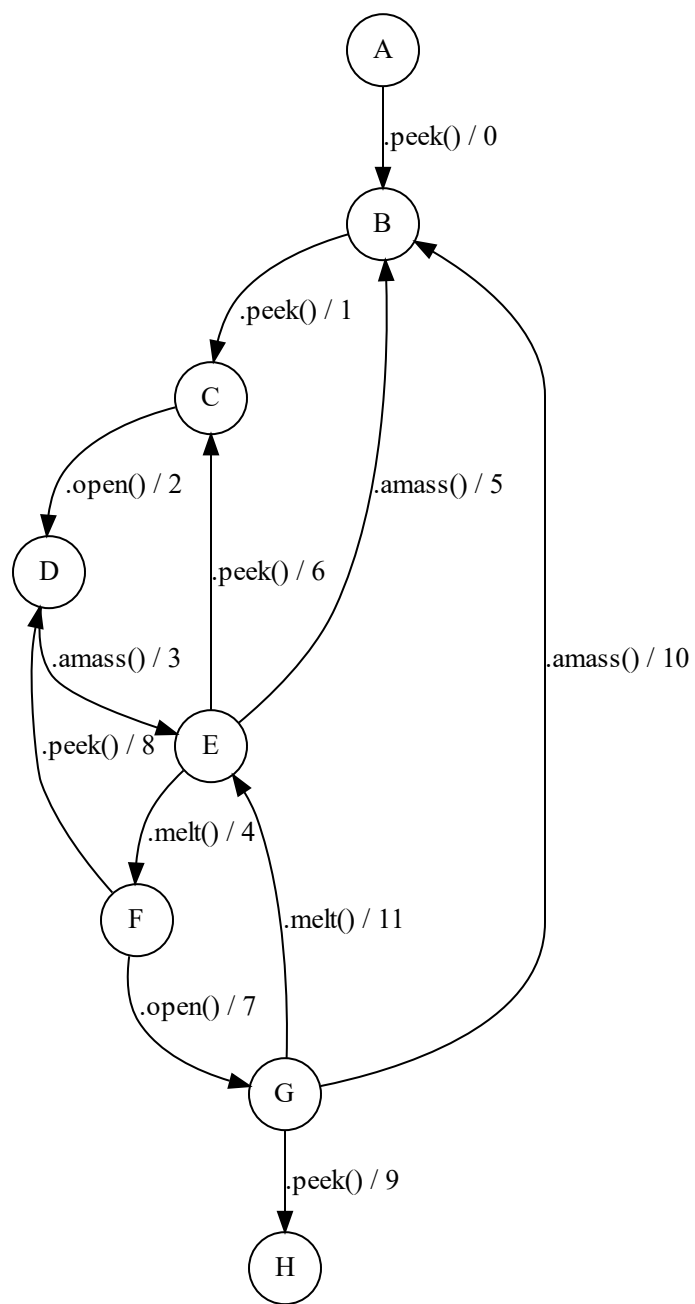
```

```

{'D1': -1429368600829198053,
 'D2': [829, -5389, -30374],
 'D3': 23886,
 'D4': -28882,
 'D5': 3264320679,
 'D6': 36554},
{'D1': 3553795917917694936,
 'D2': [106, 22345, -4184],
 'D3': 7607,
 'D4': -17649,
 'D5': 1136117738,
 'D6': 25236},
{'D1': 6183629644339236703,
 'D2': [19147, -11254],
 'D3': -20787,
 'D4': -27606,
 'D5': 2540466322,
 'D6': 58590}]]}

```

Задача 3.2. Реализовать конечный автомат Мили в виде класса C32. Начальным состоянием автомата является А. Методы возвращают числовые значения. Если вызываемый метод не реализован для некоторого состояния, необходимо вызвать исключение RuntimeError.



1. Пример использования класса C32:

o = C32()	
o.peek()	0
o.peek()	1
o.open()	2
o.amass()	3
o.melt()	4
o.peek()	8
o.amass()	3
o.amass()	5
o.peek()	1
o.open()	2
o.amass()	3
o.melt()	4
o.open()	7
o.open()	RuntimeError
o.melt()	11
o.open()	RuntimeError
o.peek()	6

2. Пример использования класса C32:

o = C32()	
o.peek()	0
o.peek()	1
o.open()	2
o.amass()	3
o.amass()	5
o.peek()	1
o.open()	2
o.amass()	3
o.melt()	4
o.peek()	8
o.amass()	3
o.melt()	4
o.open()	7
o.melt()	11
o.peek()	6

Вариант №29

Задача 3.1. Реализовать разбор двоичного формата данных (в духе формата WAD игры Doom или графического формата PNG). Данные начинаются с сигнатуры 0x51 0x4e 0x46 0x4b 0x6, за которой следует структура A. Порядок байт: от старшего к младшему. Адреса указаны в виде смещений от начала данных. В решении разрешено использовать модуль struct.

Структура A:	1	uint32
	2	Массив адресов (uint32) структур B, размер 2
	3	Структура C
	4	Массив char, размер 7
	5	Адрес (uint32) структуры D
	6	int64

Структура B:	1	float
	2	double
	3	uint8

Структура C:	1	int64
	2	int8
	3	Размер (uint32) и адрес (uint32) массива char

Структура D:	1	int16
	2	uint8
	3	Массив uint8, размер 2
	4	double
	5	int8
	6	uint64

Примеры разбора двоичного формата с помощью функции f31:

1. Двоичные данные:

```
(b'QNFK\x06=Me\x8a\x00\x00\x005\x00\x00\x00B\x1f\xd9k\x9d\xa4\xcbF\x7f:\x00\x00'
b'\x00\x03\x00\x00\x000ouogpio\x00\x00\x00R\x14\x93^\xc7"\x9b\xbe'
b'\xd8\xbd\xf0\xadk?\x96\x81.\xaeY\xe1@\xd6\xbe\xef\xd8\x82?\xee'
b'\x0b\x80\xc8\xa0\xc4D.loq]W-w\xcc?\xe2\xa7q\x85\x91\x00\xa6\xee\xad\xf0\xdc&'
b'B\xfb\xd0\x95')
```

Результат разбора:

```
{ 'A1': 1028482442,
  'A2': [{ 'B1': -0.11751826852560043, 'B2': 0.02197716654924986, 'B3': 214 },
         { 'B1': -0.4684486985206604, 'B2': 0.9389041822925113, 'B3': 46 } ],
  'A3': { 'C1': 2294983809940801151, 'C2': 58, 'C3': 'loq' },
  'A4': 'ouogpio',
```

```
'A5': {'D1': 23895,
      'D2': 45,
      'D3': [119, 204],
      'D4': 0.5829398735732141,
      'D5': -18,
      'D6': 12533759819862757525},
'A6': 1482632911707160280}
```

2. Двоичные данные:

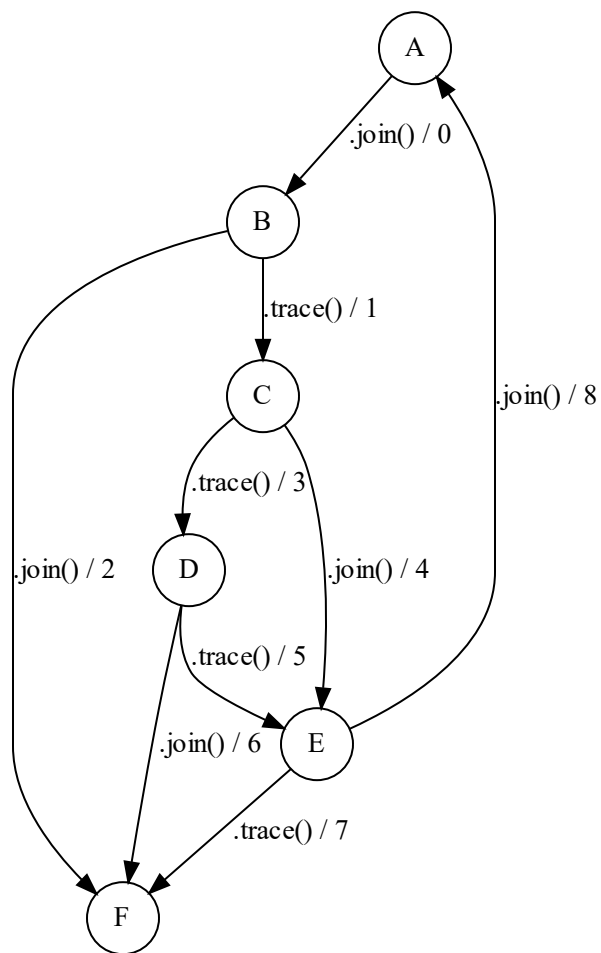
```
(b'QNFK\x06\xfb\x10\xc8\xb3\x00\x00\x005\x00\x00\x00BVv\x12\xf4!C\t'
b'R\x0e\x00\x00\x00\x04\x00\x00\x000npikbsl\x00\x00\x005\xc4(\xac\xb0\x80A\x99'
b"a\xbd\xca\xc4'\xbf\xdf[\x15\xb6\xd6\xe8\xde?6\xb2\xd2?\xd3h\x122\xe9"
b'\xcc,\xdayqxd\xc4"j\x9f^\xbf\xc8\xb6\xcc\x9c\x1a\xcaH\x056\xcf\xb6'
b'\xfa\xe4\xf4\xe5|')

```

Результат разбора:

```
{'A1': 4212181171,
 'A2': [{'B1': -0.09900694340467453, 'B2': -0.48993437629145076, 'B3': 222},
        {'B1': 0.7136660814285278, 'B2': 0.3032269952011031, 'B3': 218}],
 'A3': {'C1': 6230187974253021522, 'C2': 14, 'C3': 'yqxd'},
 'A4': 'npikbsl',
 'A5': {'D1': -15326,
      'D2': 106,
      'D3': [159, 94],
      'D4': -0.1930785906057102,
      'D5': 5,
      'D6': 3949576586926548348},
 'A6': -4312006769141245599}
```

Задача 3.2. Реализовать конечный автомат Мили в виде класса C32. Начальным состоянием автомата является А. Методы возвращают числовые значения. Если вызываемый метод не реализован для некоторого состояния, необходимо вызвать исключение RuntimeError.



1. Пример использования класса C32:

```

o = C32()
o.join()           0
o.trace()          1
o.join()           4
o.join()           8
o.join()           0
o.trace()          1
o.trace()          3
o.trace()          5
o.trace()          7

```

2. Пример использования класса C32:

o = C32()	
o.join()	0
o.trace()	1
o.join()	4
o.join()	8
o.join()	0
o.trace()	1
o.trace()	3
o.trace()	5
o.join()	8
o.join()	0
o.join()	2

Вариант №30

Задача 3.1. Реализовать разбор двоичного формата данных (в духе формата WAD игры Doom или графического формата PNG). Данные начинаются с сигнатуры 0x57 0x56 0x4c 0x57 0x3f, за которой следует структура A. Порядок байт: от младшего к старшему. Адреса указаны в виде смещений от начала данных. В решении разрешено использовать модуль struct.

Структура A:	1	uint64
	2	int16
	3	Адрес (uint32) структуры B
	4	Размер (uint32) и адрес (uint32) массива char
	5	Размер (uint32) и адрес (uint32) массива uint32
	6	Адрес (uint16) структуры D

Структура B:	1	Размер (uint32) и адрес (uint16) массива char
	2	int32
	3	Массив структур C, размер 2
	4	uint16
	5	int8
	6	uint8

Структура C:	1	uint8
	2	int8
	3	int32

Структура D:	1	int64
	2	int64
	3	uint16
	4	float
	5	uint64
	6	int16
	7	int32

Примеры разбора двоичного формата с помощью функции f31:

1. Двоичные данные:

```
(b'WVLW?\x96q\x93\x8b\xaa\xa4\x1bw\x0c\xb2)\x00\x00\x00\x03\x00\x00\x00C'
b'\x00\x00\x00\x02\x00\x00\x00F\x00\x00\x00N\x00ycak\x04\x00\x00\x00%\x00,'
b':\x05#\xf2Q&\xb4piMY\xecG\xfbF\x97\xb4\xb6\qsgC\xd2\x0f\xf0^xedy\xbe\xb30'
b'\x14\x8c\x7f\xc7\xe1\xd5\x85#\x91}e\x9ez\xc6\xb8I\xec\xceS\xbf3\x80\xb8\xe0'
b'r\xaead\xa1)\xba\x92\x832')
```

Результат разбора:

```
{'A1': 8582634567277703574,
```

```

'A2': -19956,
'A3': {'B1': 'ycak',
      'B2': 587545132,
      'B3': [{'C1': 242, 'C2': 81, 'C3': 1768993830},
              {'C1': 77, 'C2': 89, 'C3': 1190873068}],
      'B4': 46231,
      'B5': -74,
      'B6': 92},
'A4': 'qsg',
'A5': [4027568707, 3195661662],
'A6': {'D1': -3034925323246088269,
      'D2': -4144826348305570939,
      'D3': 18872,
      'D4': -0.827376127243042,
      'D5': 7233254284953419827,
      'D6': 10657,
      'D7': 847483578}}

```

2. Двоичные данные:

```

(b'WVLW?\xb2j;\xf3\r\x8eI^.(\\x00\\x00\\x00\\x08\\x00\\x00\\x00B\\x00\\x00\\x00\\x02'
 b'\\x00\\x00\\x00J\\x00\\x00\\x00R\\x00vzw\\x03\\x00\\x00\\x00%\\x00\\xf3\\xe3\\x9c\\xa7<\\x91'
 b"<\\xd3^&\\xb9N;\\xf3\\x9d\\x0e&?\\xb4\\x93xhacuerj\\xfd'5\\xd8Y\\xea\\x97\\xfa.\\xca"
 b'l\\xe4\\xbb\\xad\\x9c\\xc3\\xe9w\\x0e\\x9atF\\xd5=&-\\x97\\xbdK\\x19\\x93\\x8a2Z$\\xdf'
 b'_\\xfc\\xe3\\xa4\\x9c\\xa3')

```

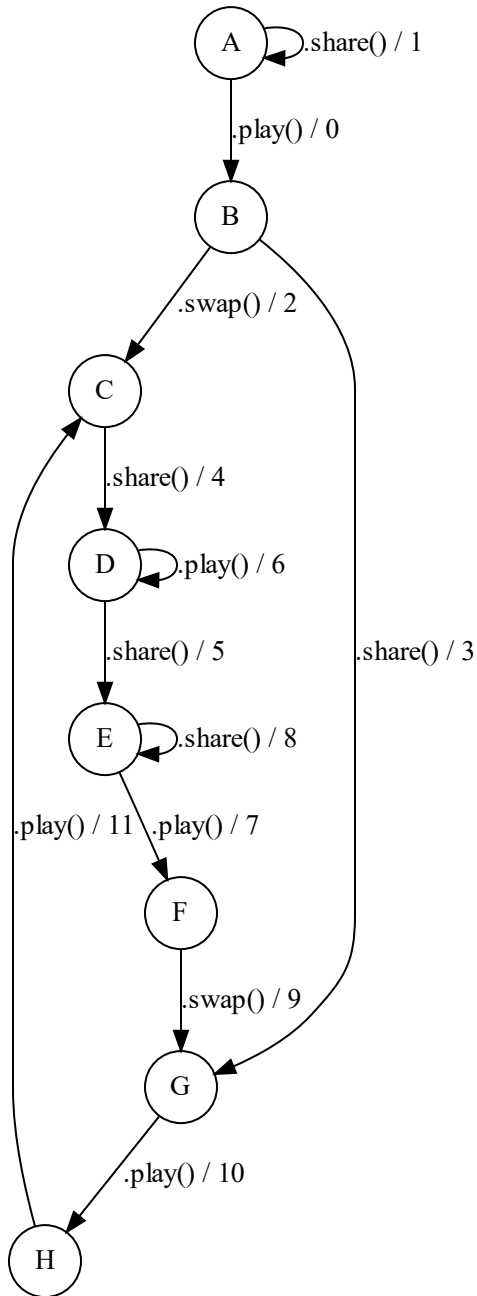
Результат разбора:

```

{'A1': 5300189149786714759,
 'A2': 11870,
 'A3': {'B1': 'vzw',
      'B2': -1482890253,
      'B3': [{'C1': 60, 'C2': -111, 'C3': 643748668},
              {'C1': 185, 'C2': 78, 'C3': 245232443}],
      'B4': 16166,
      'B5': -76,
      'B6': 147},
 'A4': 'xhacuerj',
 'A5': [3627362301, 4204259929],
 'A6': {'D1': 8175913196040538670,
      'D2': 5076852066885419932,
      'D3': 15829,
      'D4': -0.07381658256053925,
      'D5': 16079075742739274059,
      'D6': -929,
      'D7': -1550015261}}

```

Задача 3.2. Реализовать конечный автомат Мили в виде класса C32. Начальным состоянием автомата является A. Методы возвращают числовые значения. Если вызываемый метод не реализован для некоторого состояния, необходимо вызвать исключение RuntimeError.



1. Пример использования класса C32:

o = C32()	
o.share()	1
o.play()	0
o.swap()	2
o.share()	4
o.share()	5
o.play()	7
o.swap()	9
o.play()	10
o.play()	11
o.share()	4
o.play()	6
o.swap()	RuntimeError
o.share()	5
o.swap()	RuntimeError
o.share()	8
o.share()	8
o.swap()	RuntimeError
o.share()	8

2. Пример использования класса C32:

o = C32()	
o.share()	1
o.swap()	RuntimeError
o.share()	1
o.swap()	RuntimeError
o.share()	1
o.play()	0
o.swap()	2
o.share()	4
o.share()	5
o.share()	8
o.play()	7
o.swap()	9
o.share()	RuntimeError
o.play()	10
o.play()	11
o.share()	4
o.play()	6
o.share()	5

Вариант №31

Задача 3.1. Реализовать разбор двоичного формата данных (в духе формата WAD игры Doom или графического формата PNG). Данные начинаются с сигнатуры 0x73 0x4f 0x58 0x4b 0x59, за которой следует структура A. Порядок байт: от младшего к старшему. Адреса указаны в виде смещений от начала данных. В решении разрешено использовать модуль struct.

Структура A:	1	uint8
	2	Массив структур B, размер 3
	3	int64
	4	Структура D
	5	float

Структура B:	1	uint16
	2	uint8
	3	Массив адресов (uint16) структур C, размер 2

Структура C:	1	uint32
	2	Размер (uint16) и адрес (uint32) массива uint16

Структура D:	1	uint64
	2	Размер (uint32) и адрес (uint16) массива float
	3	int64

Примеры разбора двоичного формата с помощью функции f31:

1. Двоичные данные:

```
(b's0XKY\xd5=K\x87A\x000\x00@\xd1\xa6]\x00k\x00j\xd6\xddy\x00\x87\x00`'  
b'T\xb1M\xbf\x8bSD\xcf9\xfo\x82Y,\xcf\xbd\x02\x00\x00\x00\x91\x00\xcd\xc2X'  
b'\xe1\xe7\x82'\x99\xa1\xaeD\xbf\xae\xaf\x0c\x4U\xcdz\xe5\x02\x00='  
b'\x00\x00\x00\x80\xab\xd0z\xcf5\x1f\x10\xd9\x02\x00K\x00\x00\x00\x1d\x8d\xf6'  
b'0 \xa8\x94\x93\x02\x00Y\x00\x00\x00\xbf\xc9*;d\xf2\xd5\x1a\x02\x00g\x00\x00'  
b"\x00.\xa7\x1b%\xa8\xd5\xea\xe4\x02\x00u\x00\x00\x00\xd9]T\x9f\xa2q'\x81\x02"  
b'\x00\x83\x00\x00\x00.\xc8\xeb>\xf7oo\xbf')
```

Результат разбора:

```
{'A1': 213,  
 'A2': [{'B1': 19261,  
         'B2': 135,  
         'B3': [{'C1': 3850030421, 'C2': [44974, 62476]},  
                 {'C1': 3641712629, 'C2': [43904, 31440]}]},  
 {'B1': 53568,  
  'B2': 166,  
  'B3': [{'C1': 2475993120, 'C2': [36125, 12534]},  
          {'C1': 450228836, 'C2': [51647, 15146]}]},
```

```
{'B1': 54890,
  'B2': 221,
  'B3': [{'C1': 3840595368, 'C2': [42798, 9499]},
         {'C1': 2166845858, 'C2': [24025, 40788]}]},
'A3': 4923432471404565600,
'A4': {'D1': 13677199356299571449,
      'D2': [0.46051162481307983, -0.935302197933197],
      'D3': -7394766655712607539},
'A5': -0.7682896256446838}
```

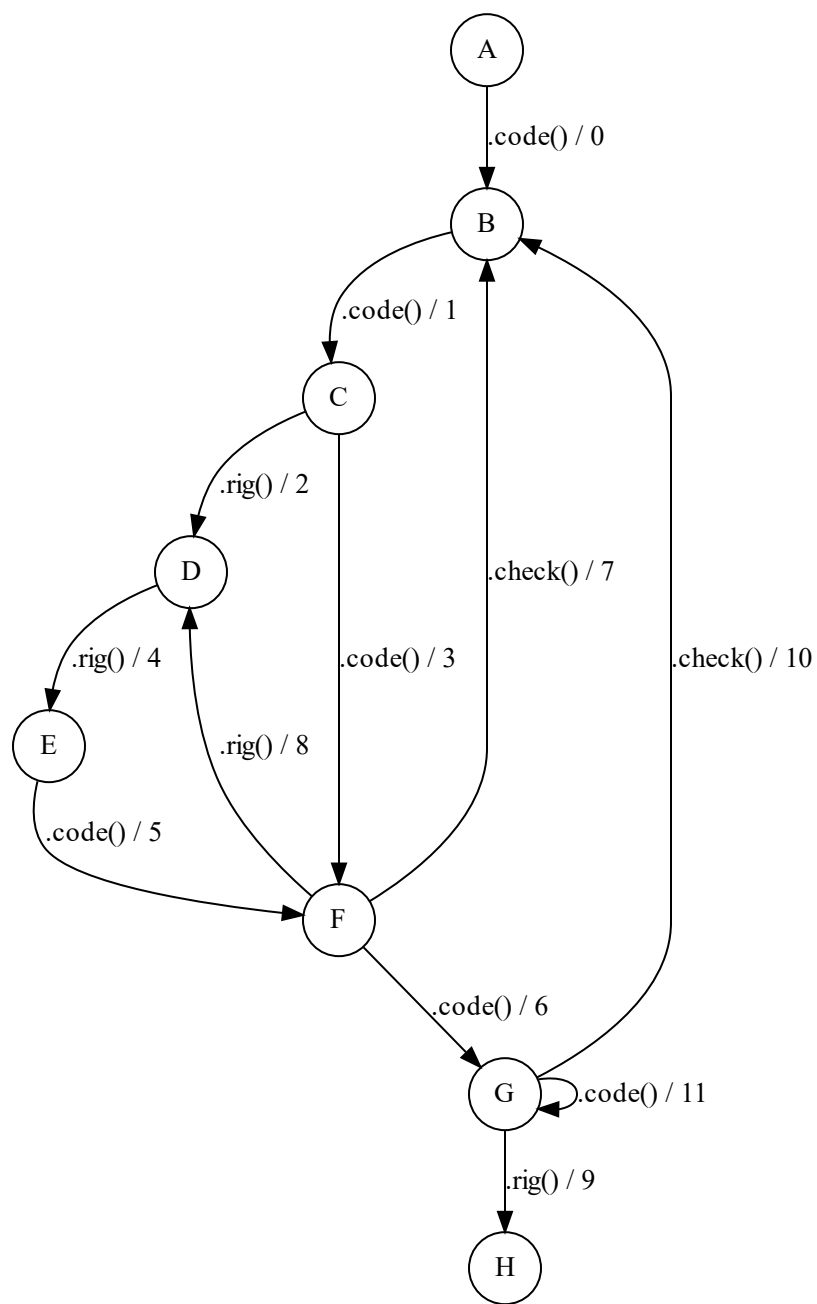
2. Двоичные данные:

```
(b's0XKY\x12\x0e\x17_A\x000\x00\x87\xd2}\x00k\x00%\xd4My\x00\x87\x00k'
b'r\x7f\xeat\xd1\xfcml\l\x93\xd0V\x88\x05s\x04\x00\x00\x00\x91\x00!4\x88'
b'? \x0b@c3u\x97Z\xbf\xc5\x13\xa2\xb9S\xba2\x86\x02\x00=\x00\x00\x00xil\xc8\x9b'
b'\xc4%\x02\x00K\x00\x00\x00:\xe7\xe5P\xc5\xfb\xfb\x02\x00Y\x00\x00\x00X'
b'J\xb6\x95\x1be#\x05\x02\x00g\x00\x00\x00I7#\xb2\x16\x02\xc5\xb5\x02\x00u'
b'\x00\x00\x005\xb1\x1e\xb5\xc3Q\xe7s\x02\x00\x83\x00\x00\x00\x8eg?'
b'? \xf3\x05\x03>!\x1e&>\x03\xb9\xf6\xbd')
```

Результат разбора:

```
{'A1': 18,
  'A2': [{'B1': 5902,
        'B2': 95,
        'B3': [{'C1': 2251471443, 'C2': [5061, 47522]},
                 {'C1': 633626779, 'C2': [27000, 51308]}]},
        {'B1': 53895,
        'B2': 125,
        'B3': [{'C1': 4093363536, 'C2': [23610, 58855]},
                 {'C1': 86205723, 'C2': [19032, 38326]}]},
        {'B1': 54309,
        'B2': 77,
        'B3': [{'C1': 3049587222, 'C2': [14153, 45603]},
                 {'C1': 1944539587, 'C2': [45365, 46366]}]}],
'A3': 7925439744345862763,
'A4': {'D1': 8288180595693153372,
      'D2': [0.7476738691329956,
              0.12795238196849823,
              0.16222430765628815,
              -0.1204700693488121],
      'D3': 3702873735683388449},
'A5': -0.8538735508918762}
```

Задача 3.2. Реализовать конечный автомат Мили в виде класса C32. Начальным состоянием автомата является А. Методы возвращают числовые значения. Если вызываемый метод не реализован для некоторого состояния, необходимо вызвать исключение RuntimeError.



1. Пример использования класса C32:

<code>o = C32()</code>	
<code>o.rig()</code>	<code>RuntimeError</code>
<code>o.code()</code>	0
<code>o.code()</code>	1
<code>o.code()</code>	3
<code>o.check()</code>	7
<code>o.code()</code>	1
<code>o.code()</code>	3
<code>o.rig()</code>	8
<code>o.rig()</code>	4
<code>o.code()</code>	5
<code>o.code()</code>	6
<code>o.code()</code>	11
<code>o.code()</code>	11
<code>o.check()</code>	10
<code>o.code()</code>	1
<code>o.rig()</code>	2
<code>o.rig()</code>	4

2. Пример использования класса C32:

<code>o = C32()</code>	
<code>o.code()</code>	0
<code>o.code()</code>	1
<code>o.rig()</code>	2
<code>o.rig()</code>	4
<code>o.code()</code>	5
<code>o.rig()</code>	8
<code>o.rig()</code>	4
<code>o.code()</code>	5
<code>o.check()</code>	7
<code>o.check()</code>	<code>RuntimeError</code>
<code>o.code()</code>	1
<code>o.check()</code>	<code>RuntimeError</code>
<code>o.code()</code>	3
<code>o.code()</code>	6
<code>o.code()</code>	11
<code>o.rig()</code>	9

Вариант №32

Задача 3.1. Реализовать разбор двоичного формата данных (в духе формата WAD игры Doom или графического формата PNG). Данные начинаются с сигнатуры 0x4b 0x54 0x47 0x48, за которой следует структура А. Порядок байт: от старшего к младшему. Адреса указаны в виде смещений от начала данных. В решении разрешено использовать модуль struct.

Структура А:	1	Размер (uint32) и адрес (uint32) массива char
	2	Структура В
	3	uint32
	4	Массив адресов (uint16) структур С, размер 2
	5	Массив int32, размер 5
	6	Размер (uint16) и адрес (uint16) массива int8
	7	uint32

Структура В:	1	int32
	2	int8
	3	int8

Структура С:	1	Структура D
	2	Размер (uint32) и адрес (uint16) массива uint8

Структура D:	1	uint32
	2	int16

Примеры разбора двоичного формата с помощью функции f31:

1. Двоичные данные:

```
(b'KTGH\x00\x00\x00\x04\x00\x00\x006\xa0\x12\x9f\xedN\x8c\x1c\xce7#\x00>'
b'\x00N\xf8\xd8@\xf1>\xc5\xbcpxe8\xae\x03\x9a\x9c\x81*\n\xb7\x9a\\=\x00\x02'
b'\x00Z\xa9\x95X\xf6jncz\x14\x90\xeb\xca\x0f,\xd4(\x00\x00\x00\x04\x00:.\x02'
b'\xfbl\xa6\xb1Xxv\xc7\x00\x00\x00\x04\x00J\xe72')
```

Результат разбора:

```
{'A1': 'jncz',
'A2': {'B1': -1609392147, 'B2': 78, 'B3': -116},
'A3': 483276579,
'A4': [{'C1': {'D1': 3396013868, 'D2': -11224}, 'C2': [20, 144, 235, 115]},
        {'C1': {'D1': 2796640376, 'D2': 30407}, 'C2': [46, 2, 251, 108]}],
'A5': [-120045327, 1053146224, -391248998, -1669256694, -1214620611],
'A6': [-25, 50],
'A7': 2845137142}
```

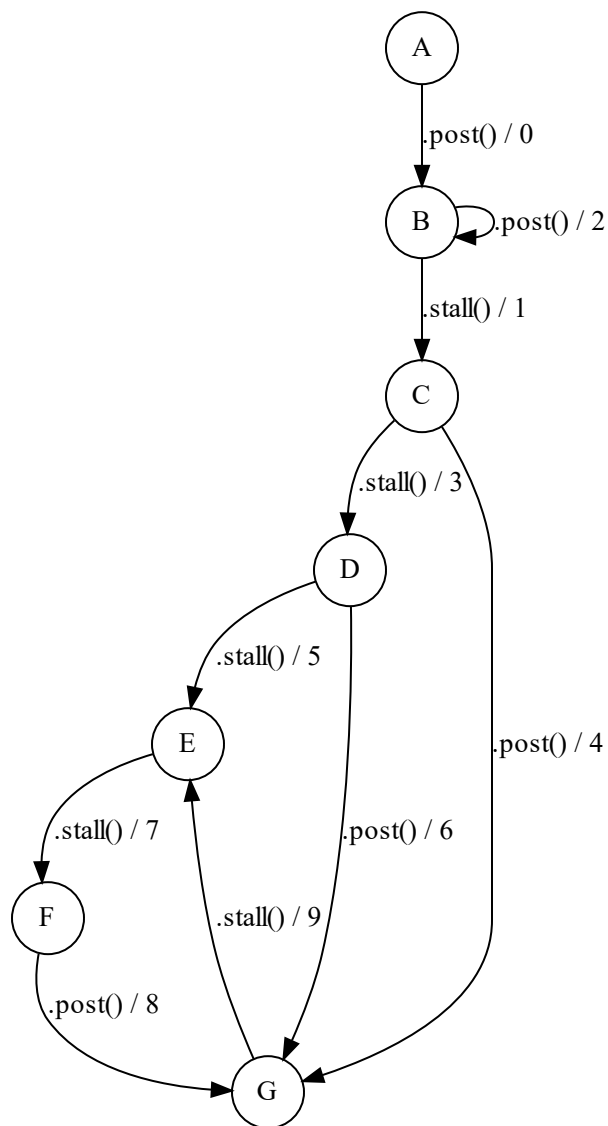
2. Двоичные данные:

```
(b'KTGH\x00\x00\x00\x04\x00\x00\x006\xe6\x10\xc4\x11\x7f\xaeZY1\xf2\x00<'
b'\x00L\xf3\xa6>\x83\xbb~\x17Y,\x92\xf5U\x91F\xda\xb8<It\xda\x00\x02'
b'\x00X\x14\xcc\xa\xebufpw\x984\xb5\xf7\xc5\xe6\xf5\xe4\x00\x00\x00\x02\x00:'
b'\x1a\xf5\x96E>R\x9c8\xc0+\x00\x00\x00\x04\x00H9\xc3')
```

Результат разбора:

```
{'A1': 'ufpw',
'A2': {'B1': -435108847, 'B2': 127, 'B3': -82},
'A3': 1515810034,
'A4': [{'C1': {'D1': 3052914150, 'D2': -2588}, 'C2': [152, 52]},
        {'C1': {'D1': 1045601336, 'D2': -16341}, 'C2': [26, 245, 150, 69]}],
'A5': [-207208829, -1149364391, 747828565, -1857627464, 1011447002],
'A6': [57, -61],
'A7': 348953323}
```

Задача 3.2. Реализовать конечный автомат Мили в виде класса C32. Начальным состоянием автомата является А. Методы возвращают числовые значения. Если вызываемый метод не реализован для некоторого состояния, необходимо вызвать исключение RuntimeError.



1. Пример использования класса C32:

<code>o = C32()</code>	
<code>o.post()</code>	0
<code>o.post()</code>	2
<code>o.stall()</code>	1
<code>o.stall()</code>	3
<code>o.stall()</code>	5
<code>o.stall()</code>	7
<code>o.post()</code>	8
<code>o.stall()</code>	9
<code>o.stall()</code>	7
<code>o.post()</code>	8
<code>o.stall()</code>	9

2. Пример использования класса C32:

<code>o = C32()</code>	
<code>o.stall()</code>	<code>RuntimeError</code>
<code>o.post()</code>	<code>0</code>
<code>o.post()</code>	<code>2</code>
<code>o.post()</code>	<code>2</code>
<code>o.stall()</code>	<code>1</code>
<code>o.stall()</code>	<code>3</code>
<code>o.post()</code>	<code>6</code>
<code>o.post()</code>	<code>RuntimeError</code>
<code>o.stall()</code>	<code>9</code>
<code>o.stall()</code>	<code>7</code>
<code>o.stall()</code>	<code>RuntimeError</code>
<code>o.post()</code>	<code>8</code>
<code>o.stall()</code>	<code>9</code>
<code>o.post()</code>	<code>RuntimeError</code>
<code>o.stall()</code>	<code>7</code>

Вариант №33

Задача 3.1. Реализовать разбор двоичного формата данных (в духе формата WAD игры Doom или графического формата PNG). Данные начинаются с сигнатуры 0x41 0x45 0x57 0x55 0xbb, за которой следует структура A. Порядок байт: от старшего к младшему. Адреса указаны в виде смещений от начала данных. В решении разрешено использовать модуль struct.

Структура A:	1	uint64
	2	uint16
	3	float
	4	int8
	5	int16
	6	Адрес (uint32) структуры B

Структура B:	1	double
	2	double
	3	int8
	4	int8
	5	Структура C
	6	Массив структур D, размер 5
	7	int64

Структура C:	1	uint64
	2	Размер (uint16) и адрес (uint32) массива char

Структура D:	1	float
	2	Массив int8, размер 7
	3	int16
	4	int16
	5	Массив int8, размер 3

Примеры разбора двоичного формата с помощью функции f31:

1. Двоичные данные:

```
(b'AEWU\xbb\xb3\xad\x17\xb1\x0b9B\xb7\xe5\x00\xbeM\x0fk\xa9\xc7h\x00\x00'
b'\x00\x1cxc?\xd3w\xd2\xa2\xde\xe2\xa0\xbf\xc1i\xc7@\xad\xb6\xf8\x9d7\x02\xcd'
b'CQ\x8c\xf0-\xfa\x00\x02\x00\x00\x00\x1a\xbf*" \xf4\xe3Y\xbf\x97'
b'\x03\xbf\xf9\xb7*}* \xa7\xf1\x17>\x84\xf9V\xac\xe2j\xed\xe8X\xad\x1c\xccP'
b'\xf0\xa1UY\xbf gBHY\xf9\xdd#\xce\x08\x12\xcc:&D\xc9I\x12?0\xc1\xf7I\x97'
b'\x11-"e\xcd0\x9e\xe9}{u` \xbe?p\x00\xcd\x91\xad\xcb\xa9\x87\xad!\x87\x0f\xa20'
b'#$\xbb\xc8\x11\xcf\xcf\xadG\xae')
```

Результат разбора:

```
{ 'A1': 12947030552924996279,
```

```

'A2': 58624,
'A3': -0.20025412738323212,
'A4': -87,
'A5': -14488,
'A6': {'B1': 0.30418840318047025,
      'B2': -0.13604059841184557,
      'B3': -99,
      'B4': 55,
      'B5': {'C1': 201891575837523706, 'C2': 'xc'},
      'B6': [{'D1': -0.6645958423614502,
              'D2': [-29, 89, -65, -105, 3, -65, -7],
              'D3': -18646,
              'D4': 32042,
              'D5': [-89, -15, 23]},
             {'D1': 0.2597147822380066,
              'D2': [-84, -30, 106, -19, -24, 88, -83],
              'D3': 7372,
              'D4': 20720,
              'D5': [-95, 85, 89]},
             {'D1': -0.9033551216125488,
              'D2': [89, -7, -35, 35, -50, 8, 18],
              'D3': -13254,
              'D4': 9796,
              'D5': [-55, 73, 18]},
             {'D1': 0.811553418636322,
              'D2': [73, -105, 17, 45, 34, 101, -51],
              'D3': 28574,
              'D4': -5763,
              'D5': [123, 117, 96]},
             {'D1': -0.18695354461669922,
              'D2': [-51, -111, -83, -53, -87, -121, -83],
              'D3': 8583,
              'D4': 4002,
              'D5': [48, 35, 36]}],
      'B7': -4915659409034754130}}

```

2. Двоичные данные:

```

(b'AEWU\xbb\x9a\x01T}\|\}|Qr\xd3\xbe\n<\xf5.\xceS\x00\x00\x00\x1cno?\xc9\xe6%'
b'tm\x03X\xbf\xc9\xf5?\x16\xd1\xd6\xf0\xe5\xeeU\x0f\xc3\xc6\x97\xb1'
b"\x13\xb1\x00\x02\x00\x00\x00\x1a?\x17\xf6'\xfb\xcc*8'\xb\x04m\x9c"
b'\x91\xa9\x04r<\xdf\xbf\xe0\x86<\x07\xa3"v\xcd\x7+\xe67\x82\xc3p\xbf;S\x7'
b'K\x80T\xae\x94$\xf3\xbcgN\xab\x05\xb9\xbf{\x99\x8esq\x8at[#\x9b\xc3AP'
b'\xbf\x1eh\x9f\xbf?\x18\x050~\xa1\x8f}/\t\xad\x1e\x82\xd2\xff\xc27%\x04'
b'\xfe\xf72\x11\x14\x04')

```

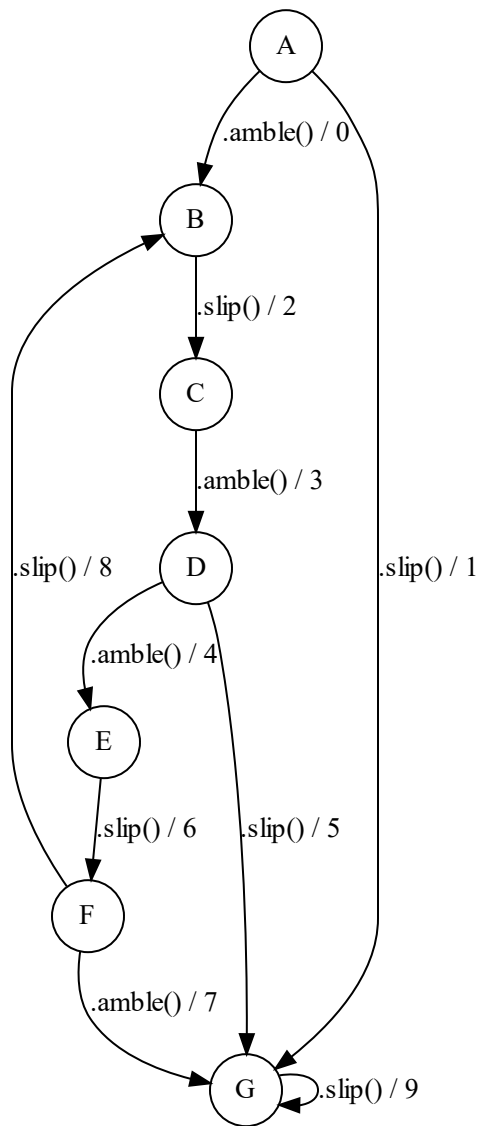
Результат разбора:

```

{'A1': 11097243854216985681,
 'A2': 29395,
 'A3': -0.13499815762043,
 'A4': 46,
 'A5': -12717,
 'A6': {'B1': 0.2023360079274663,
        'B2': -0.20279682745211103,
        'B3': -27,
        'B4': -18,
        'B5': {'C1': 6129332875590439857, 'C2': 'no'},
        'B6': [{'D1': 0.5935997366905212,
                  'D2': [-5, -52, 42, 56, -5, 4, 109],
                  'D3': -25455,
                  'D4': -22268,
                  'D5': [114, 60, -33]},
                {'D1': -0.8901447057723999,
                  'D2': [60, 7, -93, 34, 118, -57, 100],
                  'D3': -14549,
                  'D4': -6601,
                  'D5': [-126, -61, 112]},
                {'D1': -0.7317470908164978,
                  'D2': [75, -128, 84, -82, -108, 36, 120],
                  'D3': -3140,
                  'D4': 26446,
                  'D5': [-85, 5, -71]},
                {'D1': -0.9828118085861206,
                  'D2': [115, 113, -118, 116, 91, 35, -101],
                  'D3': -15551,
                  'D4': 20671,
                  'D5': [30, 104, -97]},
                {'D1': -0.7464602589607239,
                  'D2': [48, 126, -95, -113, 125, 47, 9],
                  'D3': -21218,
                  'D4': -32046,
                  'D5': [-1, -62, 55]}],
        'B7': 2667537216960533508}}

```

Задача 3.2. Реализовать конечный автомат Мили в виде класса C32. Начальным состоянием автомата является A. Методы возвращают числовые значения. Если вызываемый метод не реализован для некоторого состояния, необходимо вызвать исключение `RuntimeError`.



1. Пример использования класса C32:

```

o = C32()
o.amble()          0
o.slip()           2
o.slip()           RuntimeError
o.amble()           3
o.amble()           4
o.slip()            6
o.slip()            8
o.slip()            2
o.slip()           RuntimeError
o.amble()           3
o.slip()            5
o.slip()            9

```

2. Пример использования класса C32:

```

o = C32()
o.amble()          0
o.slip()           2
o.slip()           RuntimeError
o.amble()           3
o.amble()           4
o.slip()            6
o.slip()            8
o.amble()           RuntimeError
o.slip()            2
o.amble()           3
o.amble()           4
o.slip()            6
o.amble()           7
o.slip()            9

```

Вариант №34

Задача 3.1. Реализовать разбор двоичного формата данных (в духе формата WAD игры Doom или графического формата PNG). Данные начинаются с сигнатуры 0xd7 0x55 0x4c 0x55, за которой следует структура А. Порядок байт: от старшего к младшему. Адреса указаны в виде смещений от начала данных. В решении разрешено использовать модуль struct.

Структура А:	1	float
	2	Адрес (uint16) структуры В
	3	int8
	4	int64

Структура В:	1	Массив char, размер 3
	2	Массив структур С, размер 5
	3	Структура D
	4	int32
	5	Адрес (uint32) структуры Е
	6	int64
	7	double

Структура С:	1	double
	2	int8
	3	uint32
	4	Размер (uint16) и адрес (uint16) массива int8
	5	uint8

Структура D:	1	double
	2	Массив int8, размер 8
	3	Размер (uint16) и адрес (uint16) массива uint8
	4	int16
	5	uint64
	6	int32

Структура Е:	1	uint16
	2	int8

Примеры разбора двоичного формата с помощью функции f31:

1. Двоичные данные:

```
(b'\xd7ULU\xbe\x02F$\x00"\xa8\x86\xf6\xf1}\x1c\xda\xff\x0bC\x8de\x03\x07'  
b'+\x0b\xef\xaf\xa0\xa1\t\xc8C/qmh?\xbb*\xd7\xb9c\x80\x00\xdc\xa5\xe2'  
b'\xb0"\x00\x02\x00\x13\x94?\x990\x0b\xd0\xad\xe8@\xc0\xd5*Jv\x00\x02\x00\x15'  
b'\x9d\xbf\xc6\xb9\x85\xc1\x05\xd1X$\xdc\xe4\x10K\x00\x02\x00\x17U?\xd2q\xe7a'  
b'\xa50\x0c\xe4I\xd6\x94\xcb\x00\x02\x00\x19\x07\xbf\xe9\xb47\x97\xc6<2\xe0@3'  
b'\x8c:\x00\x02\x00\x1b\xb6\xbf\xe8b\xae\xc4\x02\x08\x16Z\xd0N\xd50'  
b'<\x81\xa7\x00\x02\x00\x1d|\xf3z\xab\xe0\xce\x99i\xbf\x94\xcc\xd4'
```

b'=\x90\x8f\xaa#\x00\x00\x00\x1f/S\x92u\xea\xe4\xdd\x03?\xd4\x0b\x89\x9a\xb9M'
b'|')

Результат разбора:

```
{'A1': -0.12722069025039673,  
'A2': {'B1': 'qmh',  
      'B2': [{'C1': 0.10612247730341551,  
              'C2': -36,  
              'C3': 2783096866,  
              'C4': [67, -115],  
              'C5': 148},  
             {'C1': 0.024597344028258528,  
              'C2': -64,  
              'C3': 3576318582,  
              'C4': [101, 3],  
              'C5': 157},  
             {'C1': -0.17753669667184613,  
              'C2': 36,  
              'C3': 3705933899,  
              'C4': [7, 43],  
              'C5': 85},  
             {'C1': 0.288202138287702,  
              'C2': -28,  
              'C3': 1238799563,  
              'C4': [11, -17],  
              'C5': 7},  
             {'C1': -0.8032491649954692,  
              'C2': -32,  
              'C3': 1077120058,  
              'C4': [-81, -96],  
              'C5': 182}],  
      'B3': {'D1': -0.7620462253837583,  
            'D2': [90, -48, 78, -43, 48, 60, -127, -89],  
            'D3': [161, 9],  
            'D4': 31987,  
            'D5': 8839405417212242367,  
            'D6': -1798515651},  
      'B4': -1869632989,  
      'B5': {'E1': 51267, 'E2': 47},  
      'B6': 3410230377999293699,  
      'B7': 0.3132041941618906},  
'A3': -88,  
'A4': -8721518108699066613}
```

2. Двоичные данные:

```
(b'\xd7ULU\xbe\xb0\xdc\x12\x00"\xe7(@Y\xf21\xb0\x8e\x16pGn\xc5v\x00;\xeb\x8b'
b'N\xc1\xdf \x9b\xfc hsc?\xa58y\xa2\xa7\xb1\xa0%\xeaA1\xf0\x00\x02\x00\x13-?'
b'\xde\xf0\xc0\xb5\x903\x8c\x9e\x04E\xb6J\x00\x02\x00\x15\xb8?\xe7D'
b'7\xf9\x80\xec.\xae\x01a\xb7\x0b\x00\x02\x00\x17\x18\xbf\xde\xa0xca\xea'
b'$H\x08\xa8\x93\xbc0\xb6\x00\x02\x00\x19\x94?\xb47T\xf0\x89\xdc`Em\x0c'
b'J_\x00\x02\x00\x1b}\xbf\xd8\xb9\xa4\xcb\xbc$ \x15~u\xf9\xee\xfc\x8c\x99\x00'
b'\x02\x00\x1d>\xc5\xb6\x10#t\xc5\rq\xa9\x83<rWf\xb3\xec:\x00\x00\x00'
b'\x1f\xc0s\xfc\xa8\x89\x16\xfaq?\xe7\t\xea\x0e\xba\x8e>')
```

Результат разбора:

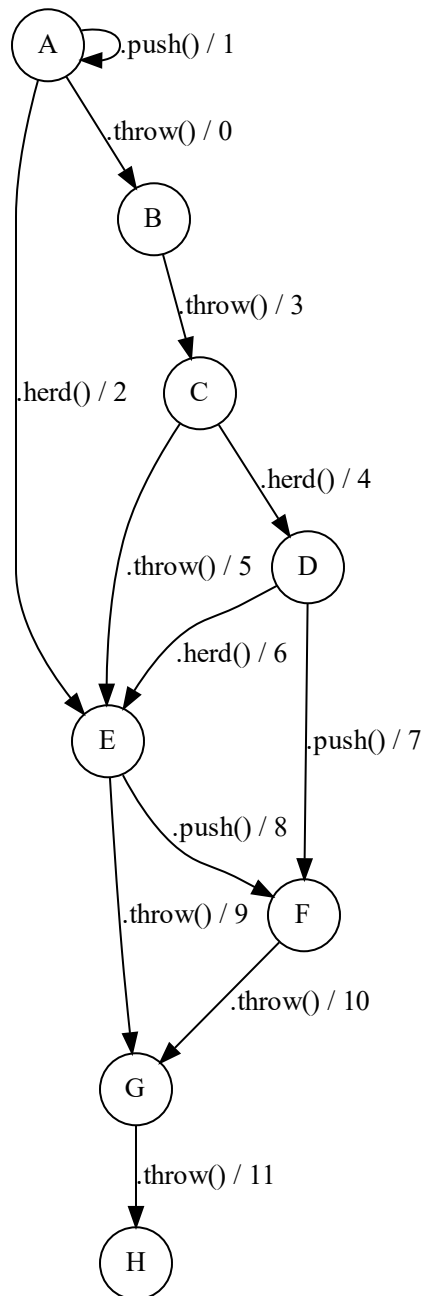
```
{'A1': -0.345429003238678,
'A2': {'B1': 'hsc',
'B2': [{'C1': 0.04144649611029938,
'C2': 37,
'C3': 3930141168,
'C4': [112, 71],
'C5': 45},
{'C1': 0.483444382961175,
'C2': -98,
'C3': 71677514,
'C4': [110, -59],
'C5': 184},
{'C1': 0.7270774720303785,
'C2': -82,
'C3': 23181067,
'C4': [118, 0],
'C5': 24},
{'C1': -0.47856400361445184,
'C2': -88,
'C3': 2478591926,
'C4': [59, -21],
'C5': 148},
{'C1': 0.07896929619333681,
'C2': 69,
'C3': 1829522015,
'C4': [-117, 78],
'C5': 125}],
'B3': {'D1': -0.3863307942970826,
'D2': [21, 126, 117, -7, -18, -4, -116, -103],
'D3': [193, 223],
'D4': 16069,
'D5': 13119024698959425961,
'D6': -2093190569},
'B4': 1723067450,
'B5': {'E1': 8347, 'E2': -4},
'B6': -4579038595320972687,
```

```

    'B7': 0.7199602401419225},
    'A3': -25,
    'A4': 2900417057767067158}

```

Задача 3.2. Реализовать конечный автомат Мили в виде класса C32. Начальным состоянием автомата является A. Методы возвращают числовые значения. Если вызываемый метод не реализован для некоторого состояния, необходимо вызвать исключение RuntimeError.



1. Пример использования класса C32:

<code>o = C32()</code>	
<code>o.push()</code>	1
<code>o.throw()</code>	0
<code>o.throw()</code>	3
<code>o.herd()</code>	4
<code>o.herd()</code>	6
<code>o.herd()</code>	RuntimeError
<code>o.push()</code>	8
<code>o.throw()</code>	10
<code>o.throw()</code>	11

2. Пример использования класса C32:

<code>o = C32()</code>	
<code>o.push()</code>	1
<code>o.throw()</code>	0
<code>o.throw()</code>	3
<code>o.push()</code>	RuntimeError
<code>o.herd()</code>	4
<code>o.herd()</code>	6
<code>o.push()</code>	8
<code>o.herd()</code>	RuntimeError
<code>o.throw()</code>	10
<code>o.push()</code>	RuntimeError
<code>o.throw()</code>	11

Вариант №35

Задача 3.1. Реализовать разбор двоичного формата данных (в духе формата WAD игры Doom или графического формата PNG). Данные начинаются с сигнатуры 0x58 0x4d 0x59 0xf7, за которой следует структура A. Порядок байт: от старшего к младшему. Адреса указаны в виде смещений от начала данных. В решении разрешено использовать модуль struct.

Структура A:	1	int32
	2	Структура B

Структура B:	1	int16
	2	Размер (uint32) и адрес (uint16) массива адресов (uint16) структур C
	3	Размер (uint32) и адрес (uint32) массива char
	4	double
	5	Адрес (uint32) структуры D
	6	int16
	7	Массив uint32, размер 6

Структура C:	1	uint32
	2	int8

Структура D:	1	Массив float, размер 7
	2	float
	3	float

Примеры разбора двоичного формата с помощью функции f31:

1. Двоичные данные:

```
(b'XMY\xf7H\xce-I\xa9\x83\x00\x00\x00\x02\x00H\x00\x00\x00\x02\x00\x00\x00L'
b'\xbf\xe7\xcf.\xcePz\x06\x00\x00\x00N\xc7\xea\xf13\xf6\x902\xb9'
b'\xf0\x8b\xfa\x97\x9e\xe7\x9d\xd2\xfc\xc7\x93\xd0+\xae\x19#I\x8c\x00\xa6'
b'b5\xb1\x8f\xb5\xa0\xf5\xb0\x00>\x00Chk=a\xdd\xfc>\xab\xe5\x00\xbf\x02'
b'\x9d\xa7\xbf>\xcd\xbf\xbf\x1fv\xea>mtF\xbf\xcd7\xae>T\xbd\xc7\xbf`A:')
```

Результат разбора:

```
{ 'A1': 1221471561,
  'A2': { 'B1': -22141,
          'B2': [{ 'C1': 10904117, 'C2': -79}, { 'C1': 2411045109, 'C2': -80}],
          'B3': 'hk',
          'B4': -0.7440408734872228,
          'B5': { 'D1': [0.055143341422080994,
                        0.33573150634765625,
                        -0.51021808385849,
                        -0.7453269362449646,
                        -0.6229082345962524,
```



```

0.23188886046409607,
-0.8978222608566284],
'D2': 0.2077551931142807,
'D3': -0.8759952783584595},
'B6': -14358,
'B7': [4046714512,
851046539,
4204240615,
2647850183,
2479893422,
421742988]}}

```

2. Двоичные данные:

```

(b'XMY\xf7\xe_o\x9c\xc6\x96\x00\x00\x00\x06\x00\\\x00\x00\x00\x02\x00\x00\x00h'
b'?\xbee\t*"\'\x90\x00\x00\x00j\x92\xe2\xab\xbcT\x81\xfd5\x92@\xa4\x11'
b'\xd27\x85\xfdI\xa1\r\xbf\xaaQp\xc9\x8c\xc9\x88\x96\xf0\xbf\x9e9\n{m\x8c'
b'n\xff*\x144`3\xe7\xb3\xc8\x04G\x94\xecV\xdbo\xcd&\x00>\x00C\x00H\x00M'
b'\x00R\x00Wxd\xbf\x08\xa6\xfa\xbe\xb0\x01+=h\xd0\x99\xbd\x93^\xfc\xbe\x91'
b'(j;\xb0m\xd3\xbf\x77?GC:>)\xe4\x11')

```

Результат разбора:

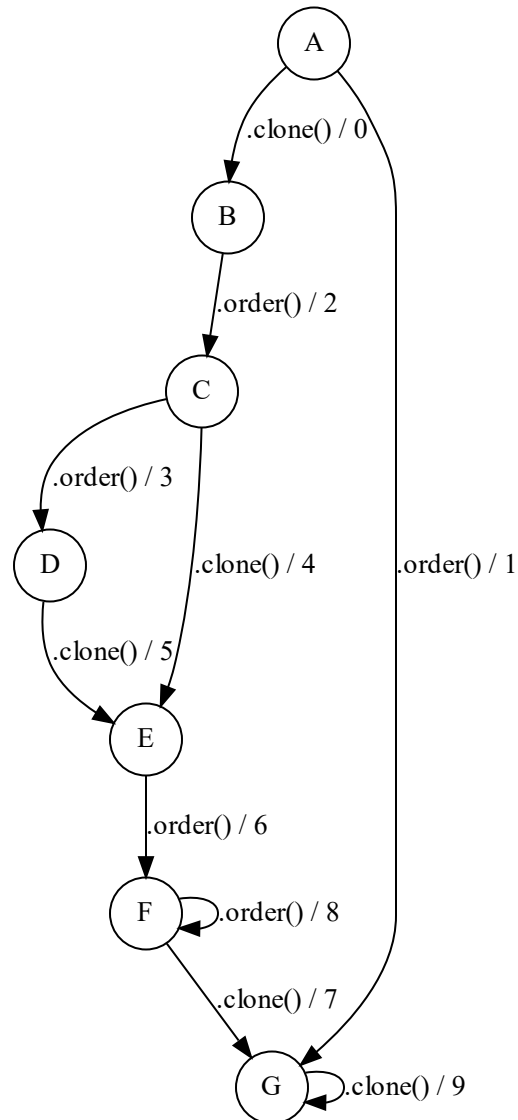
```

{'A1': -1637912676,
'A2': {'B1': -14698,
'B2': [{'C1': 2291593407, 'C2': 120},
{'C1': 3909778285, 'C2': -116},
{'C1': 1862216212, 'C2': 52},
{'C1': 1614014387, 'C2': -56},
{'C1': 71800044, 'C2': 86},
{'C1': 3681537318, 'C2': 97}],
'B3': 'xd',
'B4': 0.11872918394706589,
'B5': {'D1': [-0.5337978601455688,
-0.3437589108943939,
0.056839559227228165,
-0.0719585120677948,
-0.2835114598274231,
0.005384185817092657,
-0.8826826214790344],
'D2': 0.7783695459365845,
'D3': 0.1659090667963028},
'B6': -27934,
'B7': [2881246337,
4248146496,
2752631351,

```

2247969185,
230664785,
1892256969]]}

Задача 3.2. Реализовать конечный автомат Мили в виде класса C32. Начальным состоянием автомата является A. Методы возвращают числовые значения. Если вызываемый метод не реализован для некоторого состояния, необходимо вызвать исключение RuntimeError.



1. Пример использования класса C32:

o = C32()	
o.clone()	0
o.order()	2
o.order()	3
o.order()	RuntimeError
o.clone()	5
o.order()	6
o.order()	8
o.order()	8
o.order()	8
o.order()	8
o.clone()	7
o.clone()	9

2. Пример использования класса C32:

o = C32()	
o.clone()	0
o.order()	2
o.order()	3
o.order()	RuntimeError
o.clone()	5
o.clone()	RuntimeError
o.order()	6
o.order()	8
o.clone()	7
o.order()	RuntimeError
o.clone()	9
o.order()	RuntimeError
o.clone()	9
o.clone()	9
o.order()	RuntimeError
o.clone()	9

Вариант №36

Задача 3.1. Реализовать разбор двоичного формата данных (в духе формата WAD игры Doom или графического формата PNG). Данные начинаются с сигнатуры 0x4e 0x50 0x52, за которой следует структура А. Порядок байт: от младшего к старшему. Адреса указаны в виде смещений от начала данных. В решении разрешено использовать модуль struct.

Структура А:	1	Адрес (uint32) структуры В
	2	int8
	3	uint8
	4	Массив структур С, размер 4
	5	Адрес (uint16) структуры Е

Структура В:	1	int8
	2	int8

Структура С:	1	Массив char, размер 8
	2	int8
	3	Структура D
	4	int8
	5	int16
	6	uint8

Структура D:	1	int8
	2	float
	3	uint16

Структура Е:	1	Массив uint32, размер 4
	2	int64
	3	uint32
	4	int16
	5	Размер (uint32) и адрес (uint16) массива int64

Примеры разбора двоичного формата с помощью функции f31:

1. Двоичные данные:

```
(b'NPR[\x00\x00\x00!\x99slimguex\x8e)\xcd\xe0>\xcf\x8e\x94u\xc3vvabxvhb'  
b'p\xa7\x1d\xaf\x95V?\x97\xbc\xbd!Culeemhzgr\xcd\xdd@\xd2>\x96\\e-(\xd4lpi'  
b'rnicm\xfb\xd8\x8b\xf3s\xbf\x89&\x88tnGu\x00Y\x19\x06\xe9P\xecX\xbf\x94'  
b"\x8e\xd9\xcc\xa7\xaf\x8c\xf6K\xbb\x94\x915'\xc3>?<\x92\xbb\xa4%\xc9\xb2a"  
b'td\xfc\xce0u\xfdR\xe8I\xca\x99\xa5\x0b\x19\x97\x9c4\x03?\xfe\xe9^\x03'  
b'\x00\x00\x00]\x00')
```

Результат разбора:

```
{'A1': {'B1': 89, 'B2': 25},
'A2': 33,
'A3': 153,
'A4': [{'C1': 'slimgue',
'C2': 120,
'C3': {'D1': -114, 'D2': 0.43906524777412415, 'D3': 36559},
'C4': -108,
'C5': -15499,
'C6': 118},
{'C1': 'vabxvhbp',
'C2': -89,
'C3': {'D1': 29, 'D2': 0.8382214903831482, 'D3': 48279},
'C4': -67,
'C5': 8550,
'C6': 67},
{'C1': 'uleemhgz',
'C2': 114,
'C3': {'D1': -51, 'D2': 0.4106511175632477, 'D3': 23702},
'C4': 101,
'C5': 10285,
'C6': 212},
{'C1': 'lpirnicm',
'C2': -5,
'C3': {'D1': -40, 'D2': -0.9529349207878113, 'D3': 9865},
'C4': -120,
'C5': 28276,
'C6': 71}],
'A5': {'E1': [631552914, 1952559817, 818871396, 3897752949],
'E2': -7163229144495830455,
'E3': 4265542452,
'E4': 24297,
'E5': [-8172697035127854842,
-4950592281260864295,
4341257573705814420]}}
```

2. Двоичные данные:

```
(b'NPR[\x00\x00\x00+\x8a1rjnhtns\xc8\xaa\x1b\xc30?\xe5\x1d\x99t\x08.yqwkpkz'
b'e\xb5$\xd0t\x8c\xbe\xd93\xae\xe8\xf9+vwrxwmiz(\x10m\xaa\x1c?\xf2yq\xb2mulqw'
b'fnbgm$\xff\xfa\x95r\xbf\xd7h+z\xcb\x9cm\x00\x8e\xb6\x9b\xc3\xa6Zm\x93'
b'3\xe3\x1d&\x84)&\x92\xed\xba\xdf\x17z\x1d1^\xea\x81\x9d\xd62*\xb3:/2`\x10\xdf'
b'\xc7V\xfdZ[\xd9\xf6Z\xf6\x87\xa1\x02\x00\x00\x00]\x00')
```

Результат разбора:

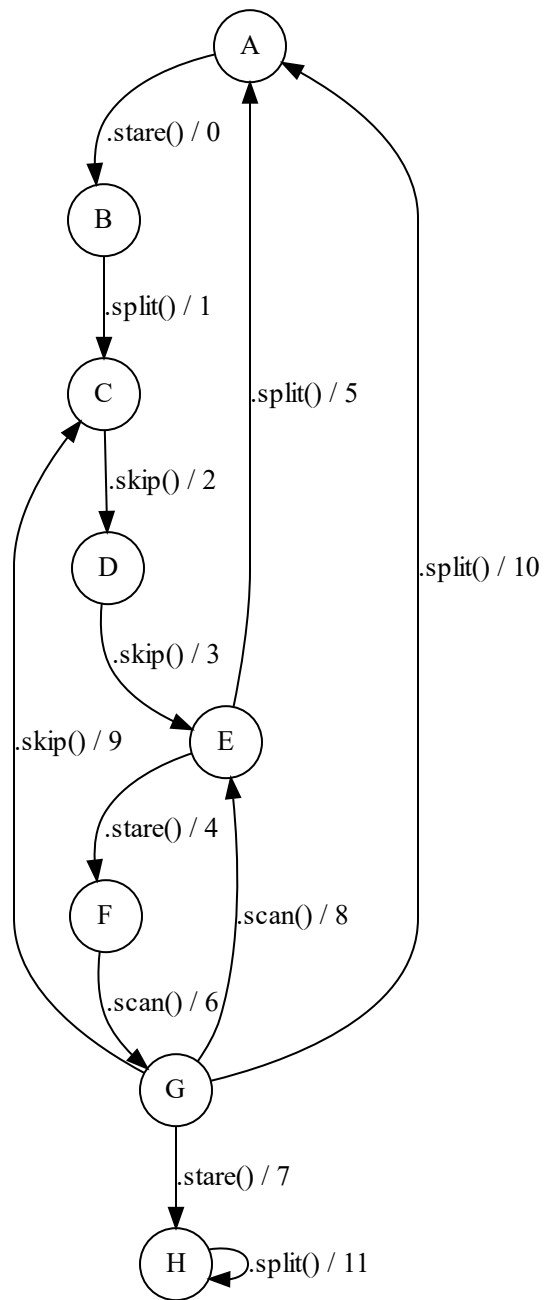
```
{'A1': {'B1': -114, 'B2': -74},
```

```

'A2': 43,
'A3': 138,
'A4': [{ 'C1': 'lrjnhnts',
          'C2': -56,
          'C3': { 'D1': -86, 'D2': 0.8115708231925964, 'D3': 7653},
          'C4': -103,
          'C5': 2164,
          'C6': 46},
        { 'C1': 'yqwkpzke',
          'C2': -75,
          'C3': { 'D1': 36, 'D2': -0.27432870864868164, 'D3': 13273},
          'C4': -82,
          'C5': -1560,
          'C6': 43},
        { 'C1': 'vwrxwmiz',
          'C2': 40,
          'C3': { 'D1': 16, 'D2': 0.6119754910469055, 'D3': 31218},
          'C4': 113,
          'C5': 28082,
          'C6': 117},
        { 'C1': 'lqwfnbgm',
          'C2': 36,
          'C3': { 'D1': -1, 'D2': -0.9476009607315063, 'D3': 26839},
          'C4': 43,
          'C5': -13446,
          'C6': 156}],
'A5': { 'E1': [2048384954, 2179620561, 707974813, 841956019],
        'E2': 6582852354517700704,
        'E3': 4133156569,
        'E4': -24185,
        'E5': [3716434353623112568, -1327956980368269853]}

```

Задача 3.2. Реализовать конечный автомат Мили в виде класса C32. Начальным состоянием автомата является А. Методы возвращают числовые значения. Если вызываемый метод не реализован для некоторого состояния, необходимо вызвать исключение `RuntimeError`.



1. Пример использования класса C32:

o = C32()	
o.skip()	RuntimeError
o.stare()	0
o.split()	1
o.skip()	2
o.skip()	3
o.stare()	4
o.scan()	6
o.scan()	8
o.split()	5
o.stare()	0
o.split()	1
o.skip()	2
o.skip()	3
o.stare()	4
o.scan()	6
o.stare()	7
o.split()	11

2. Пример использования класса C32:

o = C32()	
o.stare()	0
o.split()	1
o.skip()	2
o.skip()	3
o.stare()	4
o.scan()	6
o.scan()	8
o.split()	5
o.stare()	0
o.split()	1
o.stare()	RuntimeError
o.skip()	2
o.skip()	3
o.stare()	4
o.scan()	6
o.stare()	7
o.split()	11

Вариант №37

Задача 3.1. Реализовать разбор двоичного формата данных (в духе формата WAD игры Doom или графического формата PNG). Данные начинаются с сигнатуры 0x50 0x4b 0x47, за которой следует структура A. Порядок байт: от младшего к старшему. Адреса указаны в виде смещений от начала данных. В решении разрешено использовать модуль struct.

Структура A:	1	Размер (uint16) и адрес (uint16) массива char
	2	Массив структур B, размер 2
	3	Адрес (uint32) структуры D

Структура B:	1	Размер (uint32) и адрес (uint16) массива uint8
	2	int16
	3	int64
	4	double
	5	Адрес (uint32) структуры C
	6	double
	7	uint32

Структура C:	1	int64
	2	double
	3	uint8

Структура D:	1	Размер (uint16) и адрес (uint32) массива float
	2	int64
	3	uint16

Примеры разбора двоичного формата с помощью функции f31:

1. Двоичные данные:

```
(b'PKG\x03\x00[\x00\x02\x00\x00\x00^\x00\xb1\x12\xf9\xf6\x8b\xb4\x8a\xd5\xc7,b'
b'0\xb9g\xcc(\xeb\xbf`\x00\x00\x00\x90\xe2\x0f\xba$B\xdf?\x99\x89R\xeb\x04'
b'\x00\x00\x00q\x00\x98\t\xa6\x1f\xe8\tt\x1ah\x11<!\x0b\xb7\xa4l\xd5?u'
b'\x00\x00\x00\x00\xa1\xf4\x8ep]\xcf\xbf\xda\x1c\xf1\xd2\x92\x00\x00\x00jri-c'
b'\x11}n\x1e\xb1F\x05k,\xd2\xe7!\x9c\x96\xeb\xbf{\xea\x9dR\xf8\xd5\xbb!'
b'\x90\xeeQ\xc7E\x10\xa3 \xaf\xf6*\xb5?\x8b\xa3\xdd$\xbe\xa4\xff[?\x1e@'
b'^?\x03\x00\x86\x00\x00\x00k\xe3\xd2\xe9R\xc9\x17\x19M\x17')
```

Результат разбора:

```
{'A1': 'jri',
 'A2': [{'B1': [45, 99],
          'B2': 4785,
          'B3': 3226782449745524473,
          'B4': -0.8487302804801511,
```

```

'B5': {'C1': 7711647663475555601,
       'C2': -0.8621349966256893,
       'C3': 123},
'B6': 0.48841207666513586,
'B7': 3948054937},
{'B1': [234, 157, 82, 248],
 'B2': 2456,
 'B3': 1254281581907419046,
 'B4': 0.33475606799449387,
 'B5': {'C1': 5028077594044644309,
       'C2': 0.08268682265828553,
       'C3': 139},
 'B6': -0.24503905277493487,
 'B7': 3539016922}],
'A3': {'D1': [-0.16100172698497772, 0.8593695163726807, 0.8681658506393433],
       'D2': 1808135133359956843,
       'D3': 5965}}

```

2. Двоичные данные:

```

(b'PKG\x03\x00[\x00\x02\x00\x00\x00^\x00\xf4\x8f&\x19jX\xa1At\x02pi\x81og'
 b'\x01\xdd\xbf`\x00\x00\x00D\xda\xba_\x9a\xe8\xed\xbfX\xcb;\xc7\x03'
 b'\x00\x00\x00q\x00\xed\x98\xfa\xba\xe4\xdbn`\xfe%\x8c7Q\xfb\xc9R\xee?t'
 b'\x00\x00\x00\xa0\xa1c-\xe3M\xe0?45Y\xa4\x91\x00\x00\x00cdh\xd6k'
 b'\xab\xc3\x90\xc1\x9c\xa9\x88\xd4Lr\xfe\x7f\xe8\x13\xee?v\x96L\x94h\xe4\t\x05'
 b'\xc1\xf4\xd1\xec\x1aU\xfd:\x8dp\xe9\xbf\xea\x1a\xfe\x02>+8\xa3'
 b'\xbd\x16\xbf\x07?\x03\x00\x85\x00\x00\x00L\x12C\xce\x91])\x93\x05X')

```

Результат разбора:

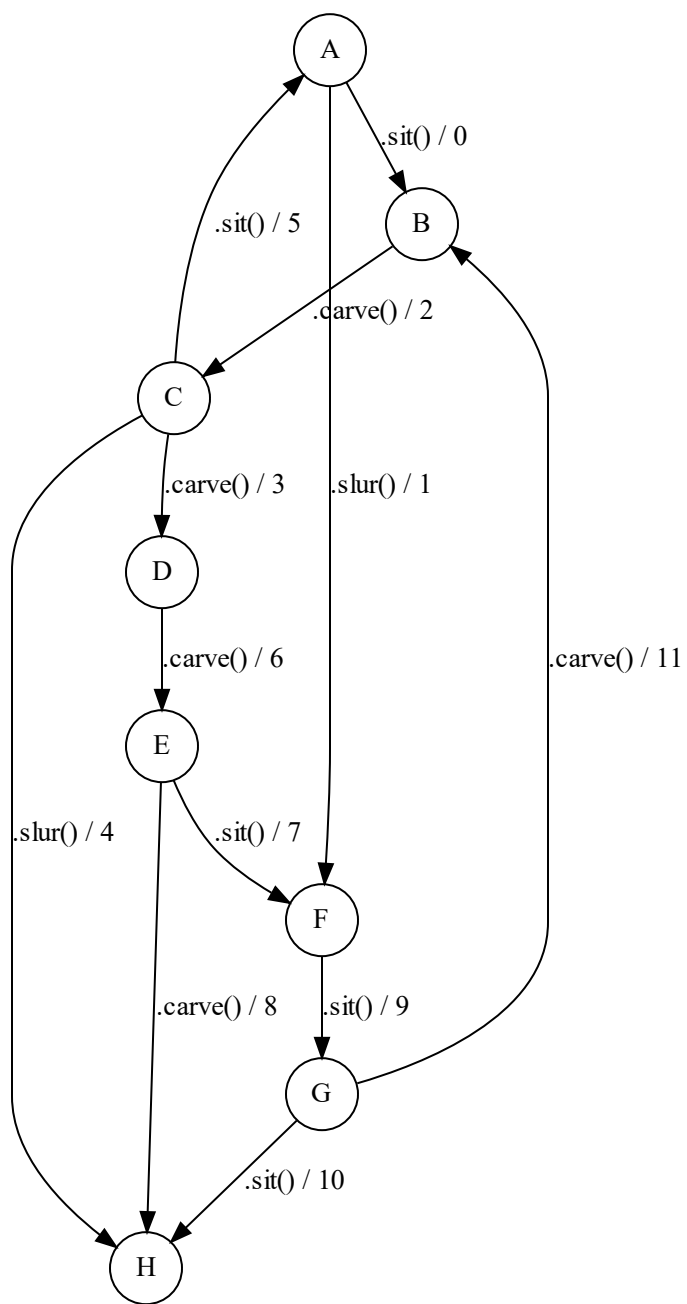
```

{'A1': 'cdh',
 'A2': [{'B1': [214, 107],
        'B2': -28684,
        'B3': 176838446603180326,
        'B4': -0.45321069611750087,
        'B5': {'C1': -3132067050108697685,
              'C2': 0.9399302005654788,
              'C3': 118},
        'B6': -0.9346439236895283,
        'B7': 3342584664},
       {'B1': [150, 76, 148],
        'B2': -26387,
        'B3': 2737731652739707898,
        'B4': 0.9476060780086626,
        'B5': {'C1': -1382054500775631768,
              'C2': -0.7949892189159613,

```

```
        'C3': 234},  
        'B6': 0.5095077406413573,  
        'B7': 2757309748}],  
    'A3': {'D1': [0.1279224455356598, -0.07969697564840317, 0.5302594900131226],  
          'D2': -7842634395276864948,  
          'D3': 22533}}
```

Задача 3.2. Реализовать конечный автомат Мили в виде класса C32. Начальным состоянием автомата является A. Методы возвращают числовые значения. Если вызываемый метод не реализован для некоторого состояния, необходимо вызвать исключение RuntimeError.



1. Пример использования класса C32:

```

o = C32()
o.sit()           0
o.carve()         2
o.carve()         3
o.carve()         6
o.sit()           7
o.sit()           9
o.slur()          RuntimeError
o.carve()         11
o.carve()         2
o.sit()           5
o.slur()          1
o.sit()           9
o.sit()           10

```

2. Пример использования класса C32:

```

o = C32()
o.sit()           0
o.carve()         2
o.sit()           5
o.slur()          1
o.sit()           9
o.carve()         11
o.carve()         2
o.carve()         3
o.carve()         6
o.sit()           7
o.sit()           9
o.sit()           10

```

Вариант №38

Задача 3.1. Реализовать разбор двоичного формата данных (в духе формата WAD игры Doom или графического формата PNG). Данные начинаются с сигнатуры 0xad 0x55 0x5a 0x4f 0x51, за которой следует структура A. Порядок байт: от старшего к младшему. Адреса указаны в виде смещений от начала данных. В решении разрешено использовать модуль struct.

Структура A:	1	Адрес (uint16) структуры B
	2	double
	3	int8
	4	Размер (uint16) и адрес (uint16) массива адресов (uint32) структур D
	5	Размер (uint32) и адрес (uint32) массива int32

Структура B:	1	Структура C
	2	int32
	3	Массив char, размер 5

Структура C:	1	int64
	2	int16
	3	int16
	4	int8
	5	uint16

Структура D:	1	Массив int8, размер 5
	2	uint32

Примеры разбора двоичного формата с помощью функции f31:

1. Двоичные данные:

```
(b'\xadUZ0Q\x00\x1c\xbf\xde\xed\xef8\xf0\xadt\t\xdb\x00\x04\x00X\x00\x00\x00\x04'
b'\x00\x00\x00h\xe5:%\x93;\x00S\xa6\xf8\xdf\xa6X\x83\xa5E\x9d\xf0\xfc\x88h1tp1'
b"m#KE\xf4\xda\xbd\x09[\xf4?'\x86|\x83s,\xce\xde\xcc_\xa0e\xd0\xeb\x08\xb4\xad"
b'1^U\x9b9\xbd\x06\xa6\x00\x00\x004\x00\x00\x00=\x00\x00\x00F\x00\x00\x000'
b'\x88(\x9e\x8f\xfaUN&\x86dJ\x0f\x8d\xb5\x92e')
```

Результат разбора:

```
{ 'A1': { 'B1': { 'C1': -1929188176094538842,
                'C2': -1825,
                'C3': -22952,
                'C4': -125,
                'C5': 42309},
  'B2': -1653605240,
  'B3': 'h1tp1'},
  'A2': -0.4832746839873103,
```

```
'A3': -37,
'A4': [{'D1': [109, 35, 75, 69, -12], 'D2': 3669023067},
      {'D1': [-12, 63, 39, -122, 124], 'D2': 2205363406},
      {'D1': [-34, -52, 95, -96, 101], 'D2': 3505064116},
      {'D1': [-83, 49, 94, 85, -101], 'D2': 968689318}],
'A5': [-2010603889, -92975578, -2040247793, -1917480347]}
```

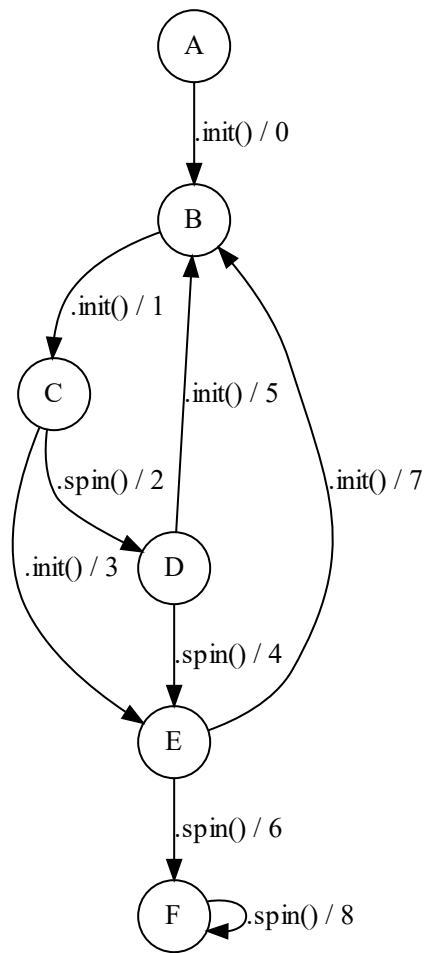
2. Двоичные данные:

```
(b'\xadUZ0Q\x00\x1c\xbf\xe4\x9c\x03!\x91\x01\x88\x81\x00\x03\x00'
b'\x00\x00\x00\x02\x00\x00\x00[\x92Ib\x10\xaa\x8a\xf6\xbf\xe0\xf6\ta'
b'\xaf\x96h\xb3\x91\xcb\xf6laekt=\x95\x81\x83\xe7\xc1\xc8\xaf|\x07\xd0\xd7'
b'\x970.2\x11f\xb2\x1cT\x9a\x98\x1d\x91\xb4\xf5\x00\x00\x004\x00\x00\x00=\x00'
b'\x00\x00F\xbcFt\x04\x17T\x05)')
```

Результат разбора:

```
{'A1': {'B1': {'C1': -7905679847151962433,
              'C2': -7946,
              'C3': 2401,
              'C4': -81,
              'C5': 38504},
        'B2': -1282290698,
        'B3': 'laekt'},
'A2': -0.6440444617838219,
'A3': -127,
'A4': [{'D1': [61, -107, -127, -125, -25], 'D2': 3251154812},
      {'D1': [7, -48, -41, -105, 48], 'D2': 775033190},
      {'D1': [-78, 28, 84, -102, -104], 'D2': 496088309}],
'A5': [-1136233468, 391382313]}
```

Задача 3.2. Реализовать конечный автомат Мили в виде класса C32. Начальным состоянием автомата является A. Методы возвращают числовые значения. Если вызываемый метод не реализован для некоторого состояния, необходимо вызвать исключение RuntimeError.



1. Пример использования класса C32:

```

o = C32()
o.init()           0
o.spin()          RuntimeError
o.init()           1
o.init()           3
o.init()           7
o.init()           1
o.spin()           2
o.init()           5
o.init()           1
o.spin()           2
o.spin()           4
o.spin()           6
o.spin()           8
  
```

2. Пример использования класса C32:

o = C32()	
o.init()	0
o.init()	1
o.spin()	2
o.init()	5
o.init()	1
o.spin()	2
o.spin()	4
o.init()	7
o.init()	1
o.init()	3
o.spin()	6
o.spin()	8

Вариант №39

Задача 3.1. Реализовать разбор двоичного формата данных (в духе формата WAD игры Doom или графического формата PNG). Данные начинаются с сигнатуры 0x48 0x5a 0x4f 0x4a 0x80, за которой следует структура А. Порядок байт: от старшего к младшему. Адреса указаны в виде смещений от начала данных. В решении разрешено использовать модуль struct.

Структура А:	1	uint8
	2	Массив char, размер 3
	3	Структура В
	4	uint64
	5	Адрес (uint16) структуры Е
	6	float

Структура В:	1	Размер (uint32) и адрес (uint16) массива структур С
	2	Массив float, размер 3
	3	Структура D
	4	uint64
	5	uint8
	6	uint64
	7	int16

Структура С:	1	int8
	2	int16

Структура D:	1	uint64
	2	int64
	3	Массив float, размер 2
	4	Размер (uint16) и адрес (uint16) массива int16
	5	int8
	6	uint32

Структура Е:	1	int32
	2	uint8
	3	int16
	4	int16

Примеры разбора двоичного формата с помощью функции f31:

1. Двоичные данные:

```
(b'HZ0J\x80\xbbbn\x00\x00\x00\x04\x00]?Rbg\xbf$\x87\xc9=,\x17\x1eD'
b'\xb0\xdf \xdd\x8f\x080@\xa37\x0e,u(X\xbf*\xab@?\x13b\xfb\x00\x05\x00i\xfd'
b"v\xdd\xe7\x87'\xed\xc4@\x03-b6\x1e<\x01\xc2'\ns\x17\xee\x08\x16/"
b"? \xdc\x1b\xbeW\n\x0e\x00s\xbf|lWdh\xb7 ' '\xea0\xcb\x86\xaf\x0e7\xb7\xd8"
b'\xc8\x91\xc5\xa3\xfa\xbd\xc1\xb4\xfb\xb4\x7f\x16\x8c\x96g\x04')
```

Результат разбора:

```
{'A1': 187,
'A2': 'bnc',
'A3': {'B1': [{'C1': 68, 'C2': 26807},
              {'C1': 32, 'C2': 10023},
              {'C1': -22, 'C2': 12491},
              {'C1': -122, 'C2': -20722}]},
'B2': [0.8218140006065369, -0.6426969170570374, 0.0420142337679863],
'B3': {'D1': 4949701322729261104,
'D2': 4657626973646170200,
'D3': [-0.6666755676269531, 0.5757290720939636],
'D4': [14263, -10040, -28219, -23558, -16959],
'D5': -3,
'D6': 1994254215},
'B4': 6984454376337072694,
'B5': 30,
'B6': 4323950590187214830,
'B7': 2070},
'A4': 3404681855030987278,
'A5': {'E1': -1258572673, 'E2': 22, 'E3': -29546, 'E4': 26372},
'A6': -0.9860281348228455}
```

2. Двоичные данные:

```
(b'HZ0J\x80\xadhxx\x00\x00\x00\x05\x00]\xbf8\xd1[>\x9d[ \xbfa\x8d\x7f\x01'
b'\xcd*\Y\x1c\x9f\x927\xa8\x1c\x94\xaf\xfc\xd6,?$He\xbe/\x80d\x00'
b'\x02\x00l\x0c\x05c\x9e\x9f}g\x1X[\x1fSs\x1b\xb4\xa8\x030\xea2Q\x03E\xf8U'
b'\x14\xfa\xb28\xf5\xf2\xc9\x00p\xbfy\xcf\t\b1\xa6]\xdf\xc44U\x88\x05'
b"r-\xa3'\x81T\xf2\xb5\x85]\x99\t\x8f\xfe\xb7\x7f\xb2")
```

Результат разбора:

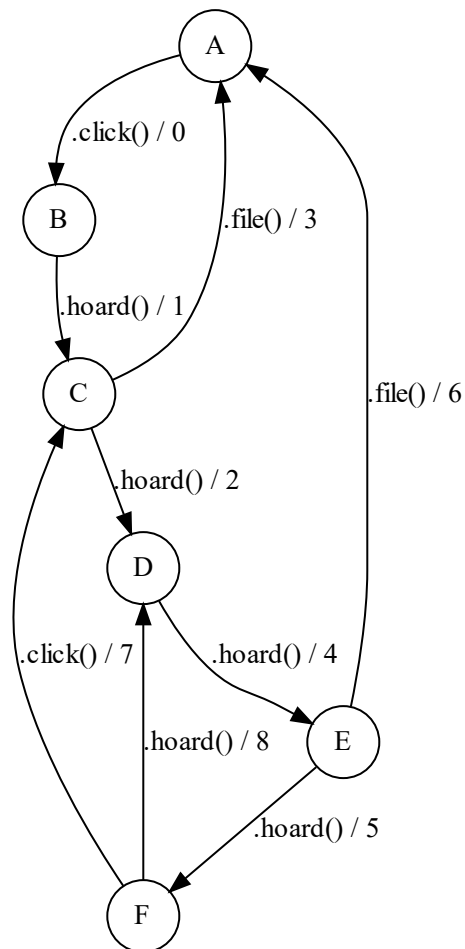
```
{'A1': 173,
'A2': 'hxx',
'A3': {'B1': [{'C1': 9, 'C2': -20058},
              {'C1': 93, 'C2': 24799},
              {'C1': -60, 'C2': 13397},
              {'C1': -120, 'C2': 1394},
              {'C1': 45, 'C2': -23769}]},
'B2': [-0.7219445109367371, 0.30733585357666016, -0.881065309047699],
'B3': {'D1': 129806540384041362,
'D2': 4010486893106746924,
'D3': [0.6417296528816223, -0.17138820886611938],
'D4': [-32428, -3403],
'D5': 12,
```

```

        'D6': 90414751},
    'B4': 9036456539130385267,
    'B5': 27,
    'B6': 13017658364680884483,
    'B7': 17912},
    'A4': 6130800636124852937,
    'A5': {'E1': -2057463543, 'E2': 143, 'E3': -329, 'E4': 32690},
    'A6': -0.9758207201957703}

```

Задача 3.2. Реализовать конечный автомат Мили в виде класса C32. Начальным состоянием автомата является A. Методы возвращают числовые значения. Если вызываемый метод не реализован для некоторого состояния, необходимо вызвать исключение RuntimeError.



1. Пример использования класса C32:

o = C32()	
o.click()	0
o.click()	RuntimeError
o.hoard()	1
o.file()	3
o.click()	0
o.hoard()	1
o.hoard()	2
o.hoard()	4
o.hoard()	5
o.hoard()	8
o.hoard()	4
o.file()	6
o.click()	0

2. Пример использования класса C32:

o = C32()	
o.click()	0
o.hoard()	1
o.hoard()	2
o.hoard()	4
o.hoard()	5
o.hoard()	8
o.hoard()	4
o.hoard()	5
o.click()	7
o.hoard()	2
o.hoard()	4
o.file()	6

Вариант №40

Задача 3.1. Реализовать разбор двоичного формата данных (в духе формата WAD игры Doom или графического формата PNG). Данные начинаются с сигнатуры 0x3 0x5a 0x53 0x41, за которой следует структура A. Порядок байт: от младшего к старшему. Адреса указаны в виде смещений от начала данных. В решении разрешено использовать модуль struct.

Структура A:	1	Адрес (uint16) структуры B
	2	uint32
	3	uint64
	4	uint32

Структура B:	1	uint16
	2	Структура C
	3	int16

Структура C:	1	float
	2	Размер (uint32) и адрес (uint32) массива структур D
	3	uint16
	4	int64
	5	int16
	6	Структура E
	7	Адрес (uint16) структуры F
	8	uint16

Структура D:	1	uint32
	2	double

Структура E:	1	uint64
	2	int64
	3	double
	4	Массив uint16, размер 7
	5	Массив uint64, размер 2
	6	uint16

Структура F:	1	Массив int8, размер 8
	2	Массив uint64, размер 3
	3	uint16
	4	int64

Примеры разбора двоичного формата с помощью функции f31:

1. Двоичные данные:

```
(b'\x03ZSAX\x00\x0b\x00\x1b\x99\xc7\x01<JQ\xe6\xa7ia\rz\xf3\x8ce'  
b'\xf9\xf6\x00\x18\x13d\xe2!\x82?\x9dG#a\xa8\xf7\x8d1?\x16\xc8?\xf8r'
```

```

b'\xfbE\x0f\x95\xd5wG)\n9,\x7f7\xb5\xc5\x162\xac~\xdd}w4;\xef\x003@'
b'\xd5\xbe\xd1\x9cp+S\xe4e\xe0\x93+\x98\x12\xd10%\xbf\x02\x00\x00\x16\x00'
b'\x00\x00\xda\xdcW\xc1\x195Z\x87\t&\x8aF\xc5k1\xda\xda\xd5\xf5\x8e\xed\x8d'
b'\xddk\xe9\xdbg\0\xad2\xf7\xb9\x8c\xbb\xbf\xd0\xe8 \xcc;\xc0\x97\xa7\x80\x00'
b'\xea;G9\xf0\xb9E:\x16 )\x92\r\xa6\x89\x91\x02\xdd\x9a\xb0\x01\xca.\x00'
b')\x00\xa4\xc5')

```

Результат разбора:

```

{'A1': {'B1': 4760,
        'B2': {'C1': -0.6452761292457581,
                'C2': [{'D1': 4143539596, 'D2': 0.008853691748651471},
                       {'D1': 1629702045, 'D2': 0.188178919979632}],
                'C3': 56538,
                'C4': 2740870669739344215,
                'C5': 18058,
                'C6': {'E1': 10301374858620136389,
                       'E2': 6658532369674112493,
                       'E3': -0.10761606488329245,
                       'E4': [59600, 52256, 49211, 42903, 128, 15338, 14663],
                       'E5': [10531984483421633008, 12725726700190148109],
                       'E6': 51713},
                'C7': {'F1': [-8, 114, -5, 69, 15, -107, -43, 119],
                       'F2': [13058045472496298311,
                               8610281598726837957,
                               13750967625048865588],
                       'F3': 40145,
                       'F4': 3140100093434342256},
                'C8': 41},
        'B3': -14940},
'A2': 2568683531,
'A3': 7613306932224065991,
'A4': 4084862305}

```

2. Двоичные данные:

```

(b'\x03ZSAX\x005\xc9\x97SMe\x80\xe5=\x0c\x9e\x16\xa1\xfc\xebw"W\xac\x95\x98\xc3'
b'{\xa4\xa9\xc7\xc4\xbfu{i=B-\xa8G1X\xeb?\x87\xdb\xfb\xaf\x90}\xb9i\x08C'
b'\xc8\x0bH\xb0_OK}\x07\x9c \xf4\xc1\x8e\xde\x1aJ\xb6\xc0{\x89^\\\x96'
b'>-\xb9\xdd$L\x1b\x9b\xd0\xfc#\xcfb\xfb\x02\x00\x00\x00\x16\x00'
b'\x00\x00\xc2\x7f-r\xfb4&\xce#\xc4q\xd9\x0e\xb4ua\xbdD\xfb2\xc1\x99\xb3\x01'
b'\x90R\x03\x85\xec*d\x7d\xeb,\x0f\x1b\xd0\xfb\x08\xc6\x0c\x8aW\xac'
b'\x1d\xb4\xbdCq\x1b\xb01\x89\xc3\x95\xa9:\xa9a\x06\xa8\xb0\xdaHY'
b'\xe5\x94\x84\xb2\xed\xc4.\x00\xc0\xe843')

```

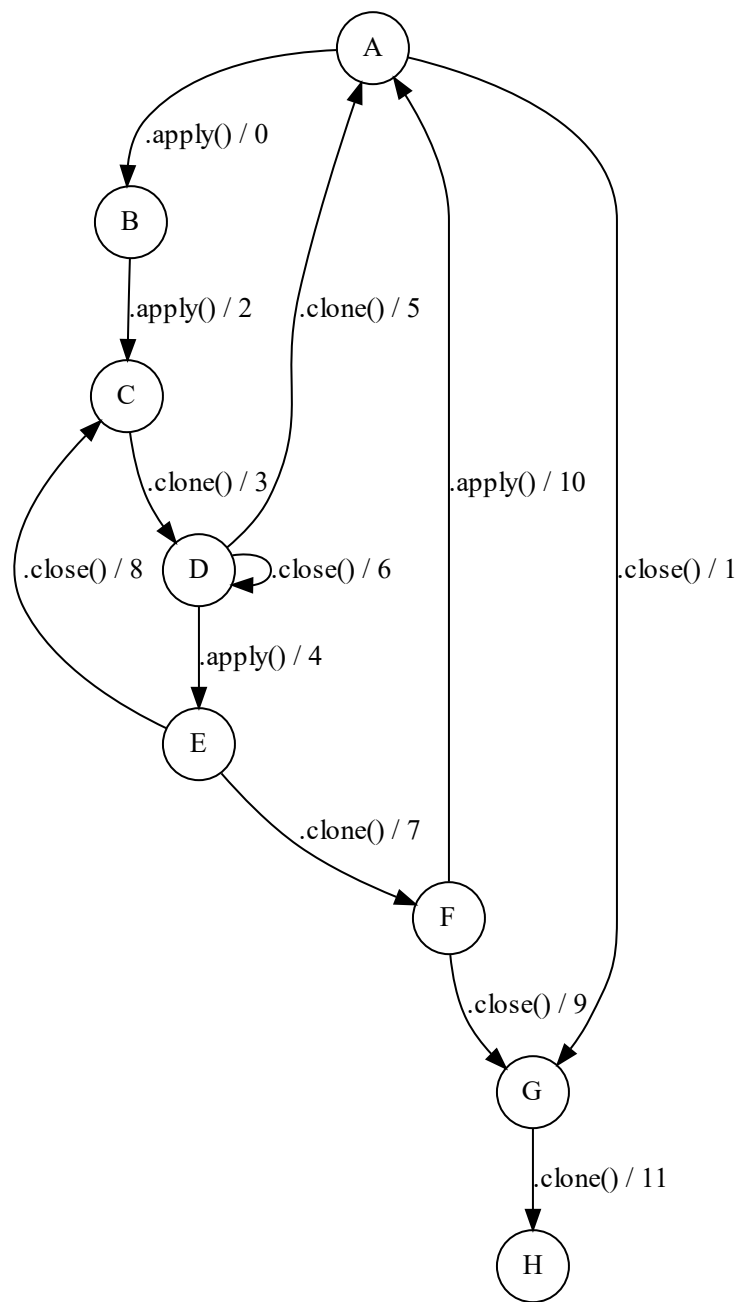
Результат разбора:

```

{'A1': {'B1': 64720,
        'B2': {'C1': -0.8859731554985046,
                'C2': [{'D1': 2511099682, 'D2': -0.16234322101038567},
                       {'D1': 1030323061, 'D2': 0.8545156859922047}],
                'C3': 32706,
                'C4': 8197716590044934701,
                'C5': 3801,
                'C6': {'E1': 11079402935357109684,
                       'E2': 3092993293413253555,
                       'E3': -0.2516515673336157,
                       'E4': [50696, 35340, 44119, 46109, 17341, 7025, 27824],
                       'E5': [12107534224976102281, 12863570148443347632],
                       'E6': 50413},
                'C7': {'F1': [-121, -37, -10, -81, -112, 125, -71, 105],
                       'F2': [5719483875265626888,
                              10286771444784790859,
                              6812111978996832990],
                       'F3': 38492,
                       'F4': -7270133452237099714},
                'C8': 59584},
        'B3': 13108},
{'A2': 1402456373,
{'A3': 1629753575137633613,
{'A4': 2011954337}

```

Задача 3.2. Реализовать конечный автомат Мили в виде класса C32. Начальным состоянием автомата является A. Методы возвращают числовые значения. Если вызываемый метод не реализован для некоторого состояния, необходимо вызвать исключение RuntimeError.



1. Пример использования класса C32:

o = C32()	
o.clone()	RuntimeError
o.apply()	0
o.apply()	2
o.clone()	3
o.apply()	4
o.apply()	RuntimeError
o.close()	8
o.clone()	3
o.clone()	5
o.apply()	0
o.clone()	RuntimeError
o.apply()	2
o.clone()	3
o.apply()	4
o.clone()	7
o.apply()	10
o.close()	1
o.clone()	11

2. Пример использования класса C32:

o = C32()	
o.apply()	0
o.apply()	2
o.apply()	RuntimeError
o.clone()	3
o.close()	6
o.close()	6
o.close()	6
o.close()	6
o.apply()	4
o.clone()	7
o.apply()	10
o.clone()	RuntimeError
o.apply()	0
o.apply()	2
o.clone()	3
o.clone()	5
o.clone()	RuntimeError
o.close()	1
o.clone()	11