

# Группа К1

## Вариант №1

**Задача 3.1.** Реализовать разбор двоичного формата данных (в духе формата WAD игры Doom или графического формата PNG). Данные начинаются с сигнатуры 0x45 0x4a 0x46 0x50 0x46, 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	uint16
	3	Массив структур D, размер 2
	4	Структура E
	5	float
	6	int8

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

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

Структура D:	1	uint64
	2	float

Структура E:	1	int32
	2	Размер (uint16) и адрес (uint32) массива int32
	3	int8
	4	uint16
	5	int64

Примеры разбора двоичного формата с помощью функции f31:

### 1. Двоичные данные:

```
(b' EJFPFm\xa0\xae\x94\xf5\xca\x00\x00\x00\x02\x00\\\x00\x00\x00\x02\x00\x00\x00'  
b'^8\xe7\xf4\x07\xc9\x8c\xf56c\xb4\xc2\xa6\xb0\x96\xfd\t\x13\x1a\xd6'  
b'\xa0\x1b_\xaf\xf4\x01?M\x14\x02R\xba\xb7\xf1V\xba\x9e\x1d>vCTf\x86'  
b'\xb2\xb3\x00\x05\x00\x00\x00nL\xd1\x92u\x92\t\xa6\xc5\xf0\xf9+\xbf]\xe0\xa5X'  
b"ah\xb7^\n\x1e}\x19\xf4\xc6'\xff\x98\x01\xcfZ\xe3\xbb\\\xb0d\xea&\x97"  
b'\x11\xd0\xf5^\xe51\x8d\x84\x0fF\xb8\xd9\xaa\xce')
```

Результат разбора:

```
{'A1': {'B1': 28064,
        'B2': -1365969462,
        'B3': 'ah',
        'B4': {'C1': [-5233734590893787962, 2882189420081636283],
                'C2': 14567,
                'C3': -862499196371573836},
        'B5': -62,
        'B6': -1498376451},
'A2': 2323,
'A3': [{'D1': 1933909129432134657, 'D2': 0.8010865449905396},
        {'D1': 5961279303948344861, 'D2': 0.24049121141433716}],
'A4': {'E1': 1720103603,
        'E2': [1555064042, 647434704, -178330319, -1920725178, -1193694514],
        'E3': 76,
        'E4': 53650,
        'E5': 8471844460927449387},
'A5': -0.8667090535163879,
'A6': 88}
```

## 2. Двоичные данные:

```
(b'EJFPF\b6\x18Z\x19\x01\xd3\x00\x00\x00\x02\x00\\\x00\x00\x00\x06\x00\x00\x00'
b'^\xd0\xcf\x88\xf8_\xff\xaat.\xdb8\x84\xa6aJ\xda3\xbc\xa59N\xb7\xe1\xe3^=\x87'
b'\xfb\xcb\x7f.MT\xe9\xa1I\x8b?k\xd4\x98\xf7\x03 #\x00\x04\x00\x00\x00\x8e'
b'e\xa9\xf6\xfb\x86\xad\xfe\xb6}D\xda?n\n\xfecme%N\xe8\xd\x15N"^O>'
b'\xf6\x87Q\xdf&\r\xeb\x8e\xc0\x06\xdf=\\\xf3\xc2J\xd0\xac\x97\x86\xae=A\xdb'
b'o\xb3\x1e\xd5\xd42\x91>${\t\x8e\xf2\x01G\xb32\x8c\xec\xe0\xcc\x1f\xc3\x90'
b'+\xa8%\xdf5\xa8')
```

Результат разбора:

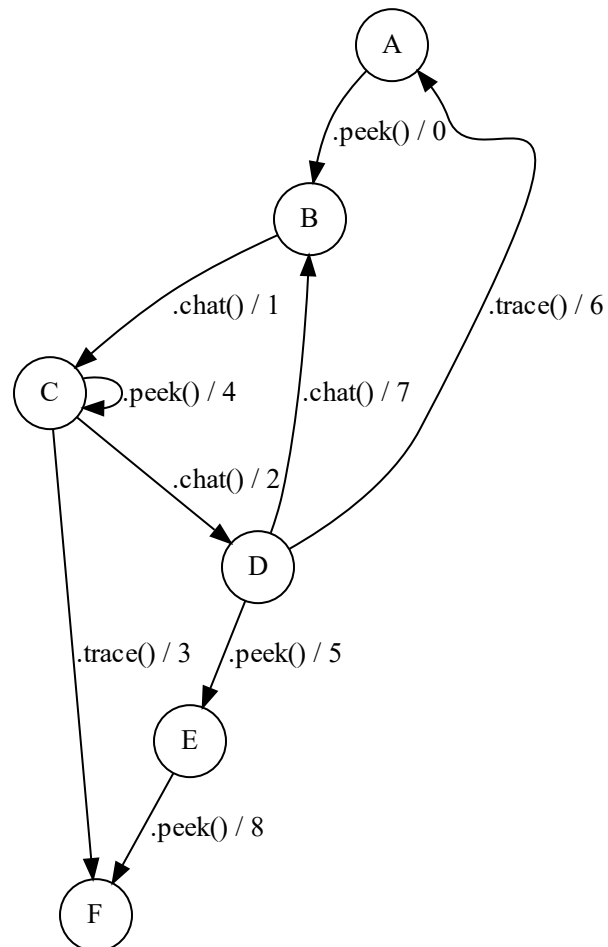
```
{'A1': {'B1': -18920,
        'B2': 1511588307,
        'B3': 'me',
        'B4': {'C1': [2688341720232239710,
                        5710272438606964237,
                        -1473028892355896077,
                        -4446512242379805123,
                        4745509447472108594,
                        -7980901378793999871],
                'C2': 53455,
                'C3': -8576999938646069541},
        'B5': 56,
        'B6': -2069470902},
```

```

'A2': 55859,
'A3': [{'D1': 13593334060542976862, 'D2': 0.06639822572469711},
        {'D1': 9164347318837791115, 'D2': 0.9212126731872559}],
'A4': {'E1': -150790109,
        'E2': [1202926220, -320812001, -1013961816, 635385256],
        'E3': 101,
        'E4': 43510,
        'E5': -322379013815450406},
'A5': 0.929855227470398,
'A6': 99}

```

**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.peek()</code>	0
<code>o.chat()</code>	1
<code>o.peek()</code>	4
<code>o.chat()</code>	2
<code>o.trace()</code>	6
<code>o.trace()</code>	RuntimeError
<code>o.peek()</code>	0
<code>o.chat()</code>	1
<code>o.chat()</code>	2
<code>o.peek()</code>	5
<code>o.trace()</code>	RuntimeError
<code>o.peek()</code>	8

## 2. Пример использования класса C32:

<code>o = C32()</code>	
<code>o.peek()</code>	0
<code>o.chat()</code>	1
<code>o.peek()</code>	4
<code>o.chat()</code>	2
<code>o.trace()</code>	6
<code>o.peek()</code>	0
<code>o.chat()</code>	1
<code>o.chat()</code>	2
<code>o.chat()</code>	7
<code>o.chat()</code>	1
<code>o.peek()</code>	4
<code>o.trace()</code>	3

## Вариант №2

**Задача 3.1.** Реализовать разбор двоичного формата данных (в духе формата WAD игры Doom или графического формата PNG). Данные начинаются с сигнатуры 0x5a 0x4f 0x50 0x53 0x1b, 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	uint8
	3	uint8
	4	Размер (uint16) и адрес (uint16) массива char
	5	int64
	6	Структура C
	7	Размер (uint16) и адрес (uint16) массива int8

Структура B:	1	uint8
	2	double

Структура C:	1	int16
	2	Размер (uint32) и адрес (uint32) массива структур D
	3	uint8
	4	int64

Структура D:	1	Массив int8, размер 5
	2	uint16
	3	uint8
	4	int32
	5	int16
	6	float

Примеры разбора двоичного формата с помощью функции f31:

1. Двоичные данные:

```
(b'ZOPS\x1b\x00,\xd4\x97\x00\x02\x005-\xe4\xc0>\xa8\xbe\xe0\x8d,\r\x00'
b"\x00\x00\x03\x00\x00\x007zK\x83\xe6d\xbb'\xc0S\x00\x04\x00m\xd6?\xe0\xb3"
b'\\\xd6\x0bi\x10ziJ\x99u\xafC\xeb96\xf3b\xc6\xb5\xe7\x1f<\xd6G\x872\xb11'
b'\xe4\xb9\x0bS\x06\x13V\xdc\xdf\x92\xb5\xbf:z\no\xdes\x8e\x98\xebV\x8av'
b"? \xcd\xb1\xd2\xc9\xbf\x18r\xbd\xb5'4\xc6")
```

Результат разбора:

```
{ 'A1': { 'B1': 214, 'B2': 0.5218948536451382 },
  'A2': 212,
  'A3': 151,
  'A4': 'zi',
```

```

'A5': 3306979401748373645,
'A6': {'C1': 11277,
      'C2': [{'D1': [74, -103, 117, -81, 67],
                'D2': 60217,
                'D3': 54,
                'D4': -211630411,
                'D5': -6369,
                'D6': 0.026157153770327568},
              {'D1': [50, -79, 49, -28, -71],
                'D2': 2899,
                'D3': 6,
                'D4': 324459743,
                'D5': -27979,
                'D6': -0.7284246683120728},
              {'D1': [111, -34, 115, -114, -104],
                'D2': 60246,
                'D3': 138,
                'D4': 1983892913,
                'D5': -11575,
                'D6': -0.5955007672309875}],
      'C3': 122,
      'C4': 5441446095104753747},
'A7': [-75, 39, 52, -58]}

```

## 2. Двоичные данные:

```

(b'ZOPS\x1b\x00,Qy\x00\x04\x005\x0c/\xa2\xe3\xf7\x86-\xb1\x86\xb5\x00'
 b'\x00\x00\x03\x00\x00\x009s^\xe4v[\xea\x86\x08\xb3\x00\x06\x00o\xce?\xc1\xc7'
 b'n\xac\xab\x96h\vpt\xb44f\x1eZ\xfaM\xc1<-\xa6\x06\xcd\xd1>c\xb3\xdc1\r\x8c/J'
 b'\x8c\x01\xf1f\x94ijI\x8c\xf2\xbe\xc2\xbe\xd7\x828\xe2rz\x17\xce\x8d\xb8W\x87'
 b'\xe8\x0b3\xbf\x19\x04<\xf8\x08\xca\xaf\xd2\x0b')

```

## Результат разбора:

```

{'A1': {'B1': 206, 'B2': 0.1388986914745829},
 'A2': 81,
 'A3': 121,
 'A4': 'zvpt',
 'A5': 878099552354577841,
 'A6': {'C1': -31051,
      'C2': [{'D1': [-76, 52, 102, 30, 90],
                'D2': 64077,
                'D3': 193,
                'D4': 1009624582,
                'D5': -12847,
                'D6': 0.22236579656600952},

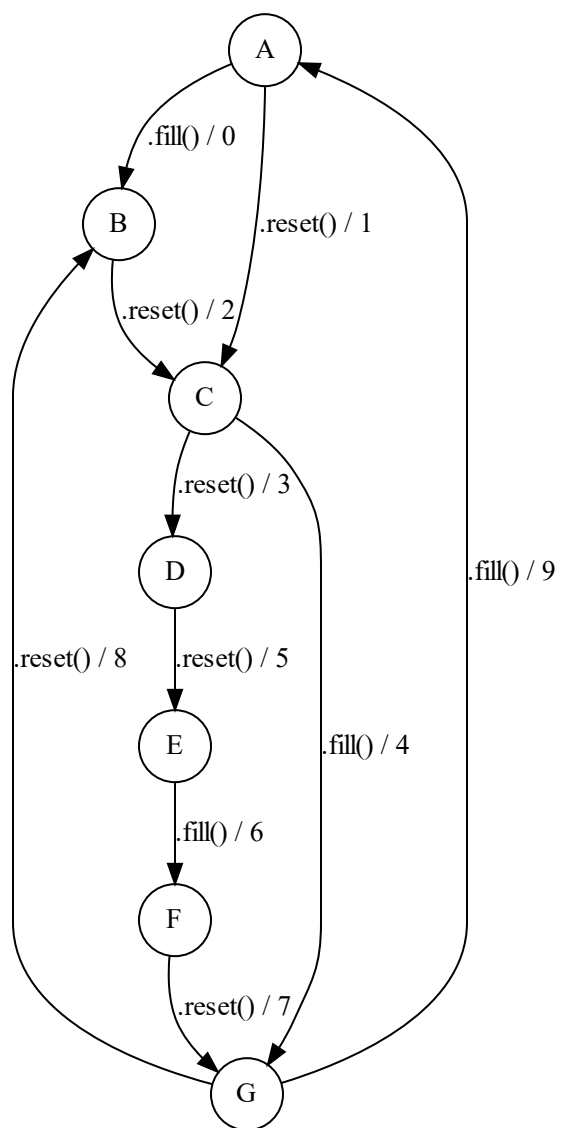
```

```

        {'D1': [49, 13, -116, 47, 74],
         'D2': 35841,
         'D3': 31,
         'D4': -1805030839,
         'D5': -29454,
         'D6': -0.38036224246025085},
        {'D1': [-126, 56, -30, 114, 122],
         'D2': 6094,
         'D3': 141,
         'D4': -1202223128,
         'D5': 2867,
         'D6': -0.5977208614349365}],
        'C3': 115,
        'C4': 6837720271404009651},
        'A7': [-8, 8, -54, -81, -46, 11]}

```

**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()	1
o.fill()	4
o.fill()	9
o.fill()	0
o.reset()	2
o.reset()	3
o.fill()	RuntimeError
o.reset()	5
o.reset()	RuntimeError
o.fill()	6
o.reset()	7
o.reset()	8
o.reset()	2
o.fill()	4
o.reset()	8

## 2. Пример использования класса C32:

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

## Вариант №3

**Задача 3.1.** Реализовать разбор двоичного формата данных (в духе формата WAD игры Doom или графического формата PNG). Данные начинаются с сигнатуры 0x96 0x4a 0x53 0x52, 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	Структура C
	3	Структура D
	4	int64
	5	uint32
	6	Размер (uint32) и адрес (uint32) массива int8

Структура B:	1	double
	2	int8

Структура C:	1	Массив char, размер 6
	2	uint16

Структура D:	1	int64
	2	uint16
	3	int16
	4	Массив структур E, размер 3
	5	Адрес (uint16) структуры F
	6	int64
	7	int8
	8	float

Структура E:	1	uint16
	2	double
	3	Массив int8, размер 8

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

Примеры разбора двоичного формата с помощью функции f31:

### 1. Двоичные данные:

```
(b'\x96JSR?\xba\x9e\xc6R\x1bZ\x80+yiluqy4\xa2`\xa1b\xd40W\x1a\xd3"\xe7\x99'
b'J\xb7\xb6?\xe8\x99\xf9\xb5\t}b\x08\xbfJY!z:[/\xff?\xedL\xf5\xafAe'
b'\xfe\x0c\x8f"\x7f%o>\xe9\x19q\xbf\xe6\x0c\xf1\xe6z\xc10\t\x06\xbbg\x86'
```

```

b'\xf5H\x94\x00\x86\xd8\x9e\xees\xcd\xbb\xf8\xe1}\xbfd?\x06\xc9\xea'
b'\xfd\xf8\xff\xd4\xcf\xa2\xb7\xcf\r\x0b\x00\x00\x02\x00\x00\x98?', '
b'ia\xbe\xa9?g\xbfM\x96\xa7>pg\x9d\xb4\xee&;\x00\x03\x00z\xff\xc1^\xf0\x1fL'
b'\xd2\xbc')

```

Результат разбора:

```

{'A1': {'B1': 0.10398520950853118, 'B2': 43},
'A2': {'C1': 'yiluqy', 'C2': 13474},
'A3': {'D1': 6962955162895194835,
'D2': 8935,
'D3': -26294,
'D4': [{'E1': 47030,
'E2': 0.7687958274727118,
'E3': [8, -65, 74, 89, 33, 122, 58, 91]},
{'E1': 12287,
'E2': 0.9156444952924969,
'E3': [12, -113, 34, 127, 37, 111, 62, -23]},
{'E1': 6513,
'E2': -0.6890801908066688,
'E3': [9, 6, -69, 103, -122, -11, 72, -108]}]},
'D5': {'F1': 0.2347702533006668,
'F2': -1259461061,
'F3': [0.6734829545021057,
-0.3305618464946747,
-0.8030800223350525],
'F4': 4290862832,
'F5': 8012},
'D6': -2837568534056535839,
'D7': 125,
'D8': -0.8915866613388062},
'A4': -3897023281649889374,
'A5': 3083799819,
'A6': [-46, -68]}

```

## 2. Двоичные данные:

```

(b'\x96JSR\xbf\xeb\xf9\xb8<L\xd9\xe8[qxsjks\xd5!\x15\xcc\x1e\x19\x88/m'
b'\x06\xce2\x0e\x97u\xa4\xbf\xd7\x86\xb6\x8eB\xd44/\xee\xda\xf5{k\x17\xa4\xed'
b'\x9e\xbf\xe3\x9a\xc6!\x080\x0c\xfe\xddt\xb3\x89\xa1}\xcd\xdf\xbf'
b'\x91\x1e\xf9\x85A\xde\x80\xce\nKa\t>2\xc6\x00\x86\xda\x98\xc0\x8f\x11\xc3X'
b'F\xf5?`\x1f\xf9\xa1\xcc\x8a+\xdeNCK\x18\xa5\x883\x00\x00\x02\x00\x00'
b'\x00\x98\xbd\x90\xd7(\xbfsmm\xbe\xe7#v=\xea\x99\\\xe3J\x9d\xa9\x00\x03'
b'\x00z\r\xb5W\xce\xd2\x95\xa8')

```

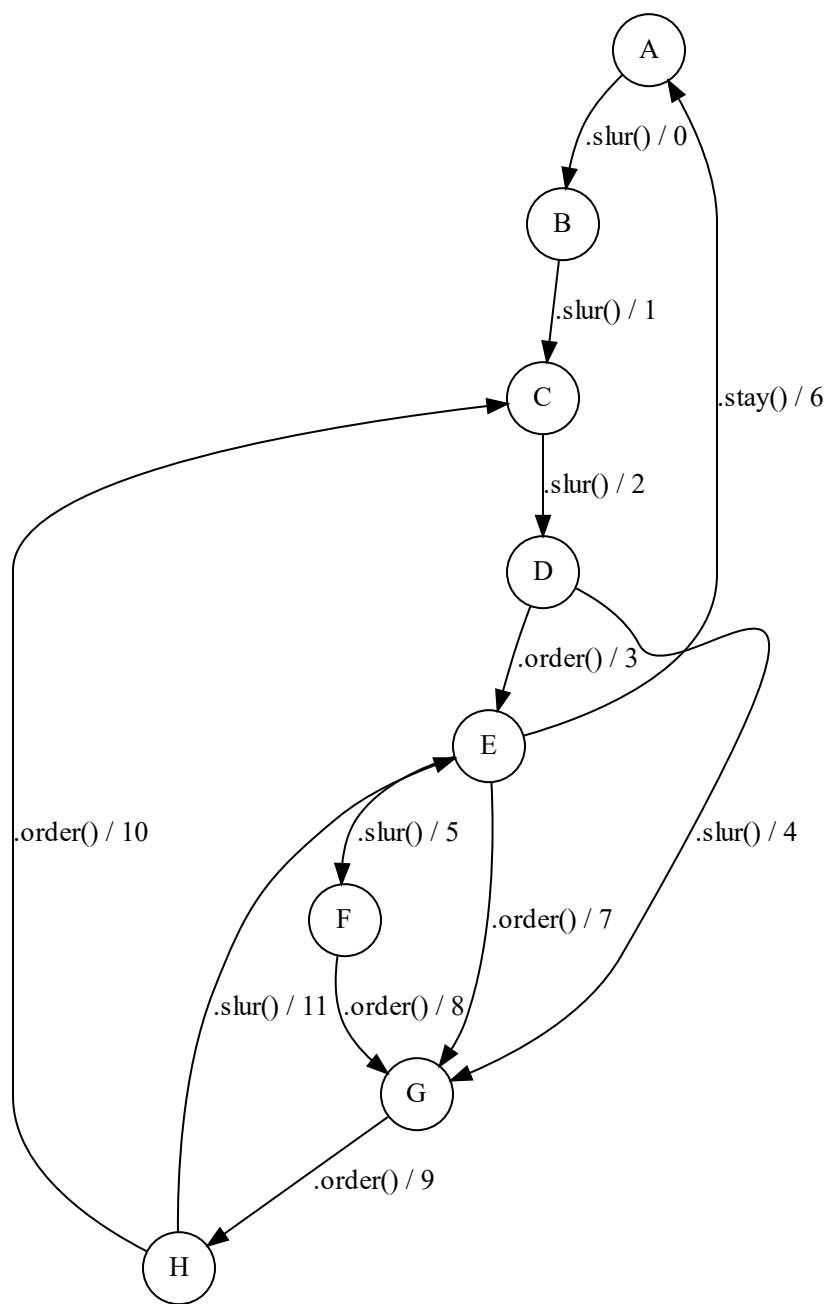
Результат разбора:

```

{'A1': {'B1': -0.8742333581674844, 'B2': 91},
'A2': {'C1': 'qxsjks', 'C2': 54561},
'A3': {'D1': 1570663465053285638,
'D2': 52786,
'D3': 3735,
'D4': [{'E1': 30116,
'E2': -0.36759723560982427,
'E3': [47, -18, -38, -11, 123, 107, 23, -92]},
{'E1': 60830,
'E2': -0.6084722394664421,
'E3': [12, -2, -35, 116, -77, -119, -95, 125]},
{'E1': 52703,
'E2': -0.016719721563701828,
'E3': [-50, 10, 75, 97, 9, 62, 50, -58]}],
'D5': {'F1': 0.11455032229423523,
'F2': -481649239,
'F3': [-0.07072287797927856,
-0.950888454914093,
-0.4514424204826355],
'F4': 229988302,
'F5': 53810},
'D6': -2695192656270370746,
'D7': -11,
'D8': 0.8754878640174866},
'A4': -6787898617336282293,
'A5': 413501491,
'A6': [-107, -88]]

```

**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.order()	RuntimeError
o.slur()	0
o.slur()	1
o.slur()	2
o.stay()	RuntimeError
o.slur()	4
o.order()	9
o.order()	10
o.slur()	2
o.order()	3
o.order()	7
o.order()	9
o.stay()	RuntimeError
o.slur()	11
o.slur()	5
o.slur()	RuntimeError
o.order()	8
o.order()	9
o.slur()	11
o.stay()	6

## 2. Пример использования класса C32:

o = C32()	
o.slur()	0
o.slur()	1
o.slur()	2
o.order()	3
o.slur()	5
o.order()	8
o.order()	9
o.slur()	11
o.order()	7
o.order()	9
o.order()	10
o.slur()	2
o.slur()	4
o.stay()	RuntimeError
o.order()	9
o.slur()	11
o.stay()	6

## Вариант №4

**Задача 3.1.** Реализовать разбор двоичного формата данных (в духе формата WAD игры Doom или графического формата PNG). Данные начинаются с сигнатуры 0x5d 0x52 0x4d 0x56, 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	Адрес (uint16) структуры B

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

Структура C:	1	uint8
	2	int16
	3	uint64
	4	double

Структура D:	1	Массив uint8, размер 8
	2	Массив int32, размер 3
	3	int16
	4	uint32
	5	uint32
	6	Массив int64, размер 4
	7	int16

Примеры разбора двоичного формата с помощью функции f31:

### 1. Двоичные данные:

```
(b'JRMV&\xa9\x90\xa99\x04\x16\xf0t\x00\xa0c\x7f\x00QQ5j\xa5\xd0\x9d\xd8\xb1\t'
b'p\xdb!\xd6?,\xd1\xba\T\x13!G\x11\xb1\x9c\x1eF\n\x82\x92\x80\xec?'
b'\xb1\x08\xf0\x07\xf5Tv\xdc\x15\nHA\xfcE\xc3\xb6\x19:A\x17Be\xf1\xc4'
b'\xd8b<\xa7\xc0%D\x08\x03\x0c\x1d\x83\xd5\xdf\x17\xb9V2@\xea_\xe1\xeb\x95'
b"S\xc7\xd9\xd7\x9b\x1e\x07*'d\xe3\x9c\xab\x1a.\xed\x0e\x00\x00\x00"
b'!\x00\x00\x004\x00\xe8\xe9\x8aZ')
```

Результат разбора:

```
{'A1': 17300019664294816038,
'A2': {'B1': [{'C1': 160,
                'C2': 32611,
                'C3': 11371770934690140416,
                'C4': 0.3458164781684707}],
```

```

{'C1': 44,
 'C2': -17711,
 'C3': 11290824737989809244,
 'C4': 0.8906948604437195}},
'B2': {'D1': [177, 8, 240, 7, 245, 84, 118, 220],
 'D2': [1095240213, -1228716548, 390150681],
 'D3': 25922,
 'D4': 1658373361,
 'D5': 633382716,
 'D6': [-2317802272433436604,
        -2206787730944902889,
        2205593772880008683,
        1921802166455446023],
 'D7': -4818},
'B3': -24,
'B4': 1519053112}}

```

## 2. Двоичные данные:

```

(b' ]RMV\xd4\xde\xb2d\xf3\x9co\xebt\x00\xa8\xb7\xaaN(\x1c\x97\x96\x80\xf3'
 b'\xb4\xd0\xaf\x82#\x06\xb3\xbd\xbf\xc5\xc3_\xb6\x98q9\x98\xa2\xb0\x1a'
 b"\xc0C\xeb{\xeee\x80\xc9?\x9fE'\x9d6\x11\x15W(\x0b\xfd$b\xf7\xa0\x83"
 b'_\xc2\xdc\xb4g\xa6@^5A~B\xde\xd6"\xaf\xceXB/\xd2\xc4\xdb\xbd\xe3\x0et\x9e'
 b'\x127X \x15\x82\xb9n\xaem~\x82\xf8\xcdXN\xb3B|\xec\x0e\x00\x00\x00'
 b'!\x00\x00\x004\x00\x89\xba\xa4\x08\x17')

```

## Результат разбора:

```

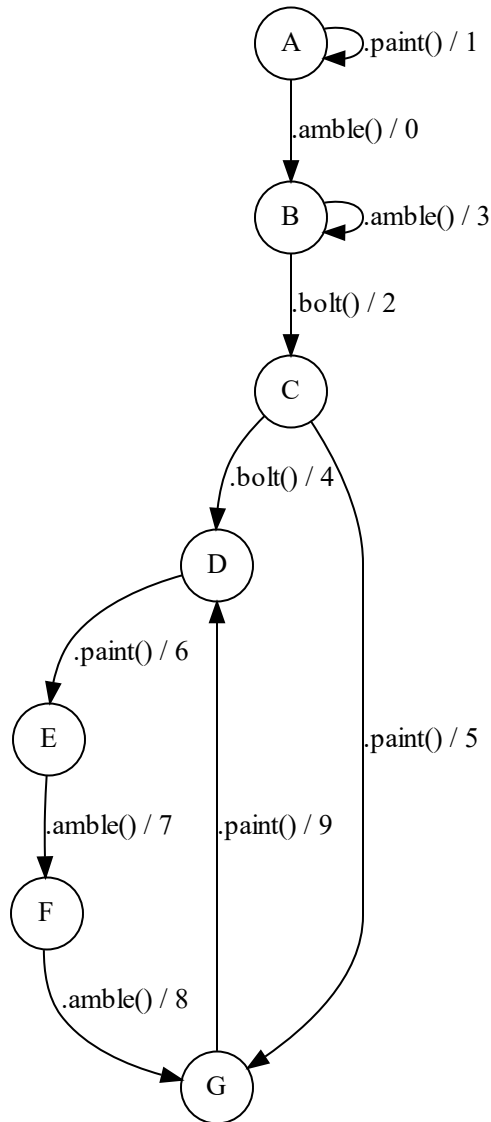
{'A1': 16964950890508377812,
 'A2': {'B1': [{'C1': 168,
 'C2': -21833,
 'C3': 13038906730436372558,
 'C4': -0.1160129391379876},
 {'C1': 197,
 'C2': 24515,
 'C3': 1923215815569676470,
 'C4': 0.19924717951519533}],
 'B2': {'D1': [159, 69, 39, 157, 54, 17, 21, 87],
 'D2': [620563240, -2086602910, -1260600737],
 'D3': -22937,
 'D4': 1094016576,
 'D5': 3604890238,
 'D6': [-4264293935162151134,
        3968408442970029531,
        7903376139112226904,
        4806271370654155390],

```



```
'D7': -4996},  
'B3': -119,  
'B4': 386442426}}
```

**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.bolt()	RuntimeError
o.paint()	1
o.bolt()	RuntimeError
o.paint()	1
o.amble()	0
o.amble()	3
o.bolt()	2
o.paint()	5
o.paint()	9
o.paint()	6
o.amble()	7
o.amble()	8
o.paint()	9

## 2. Пример использования класса C32:

o = C32()	
o.paint()	1
o.amble()	0
o.amble()	3
o.bolt()	2
o.bolt()	4
o.paint()	6
o.amble()	7
o.amble()	8
o.paint()	9
o.paint()	6
o.amble()	7

## Вариант №5

**Задача 3.1.** Реализовать разбор двоичного формата данных (в духе формата WAD игры Doom или графического формата PNG). Данные начинаются с сигнатуры 0x4d 0x4a 0x49, 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 (uint32) и адрес (uint16) массива char
	2	Размер (uint32) и адрес (uint32) массива адресов (uint16) структур B
	3	float
	4	int32
	5	uint8
	6	Структура C

Структура B:	1	int8
	2	uint32

Структура C:	1	Адрес (uint16) структуры D
	2	int32
	3	Массив int32, размер 7

Структура D:	1	uint16
	2	Массив uint8, размер 5
	3	int8
	4	int16

Примеры разбора двоичного формата с помощью функции f31:

### 1. Двоичные данные:

```
(b'MJI\x06\x00\x00\x00<\x00\x02\x00\x00\x00L\x00\x00\x00w\xc3\x80=. \x0ej'  
b'~\xb3P\x00\x0fri\x0e\xee\xe8\xa8L\x8fs@{m\x83m\x16})\x11\xaa8 gz\x1b'  
b"\x05\x11\x14t\x8d'==dkgemp\xcc\xc5\x91tW\x9cR\xda \x96B\x00G\x00"  
b'\x97\x8e\xcf.{\x0c\xf5DQZ')
```

Результат разбора:

```
{ 'A1': 'dkgemp',  
  'A2': [{ 'B1': -52, 'B2': 1467257285}, { 'B1': -100, 'B2': 2518735442}],  
  'A3': 0.06287281960248947,  
  'A4': 2120879662,  
  'A5': 179,  
  'A6': { 'C1': { 'D1': 36503,  
                  'D2': [207, 46, 123, 12, 245],  
                  'D3': 68,  
                  'D4': 23121},
```

```
'C2': 241791503,
'C3': [1286138094,
       2067821455,
       376275821,
       950669609,
       461006624,
       1947472133,
       1027417997]]}
```

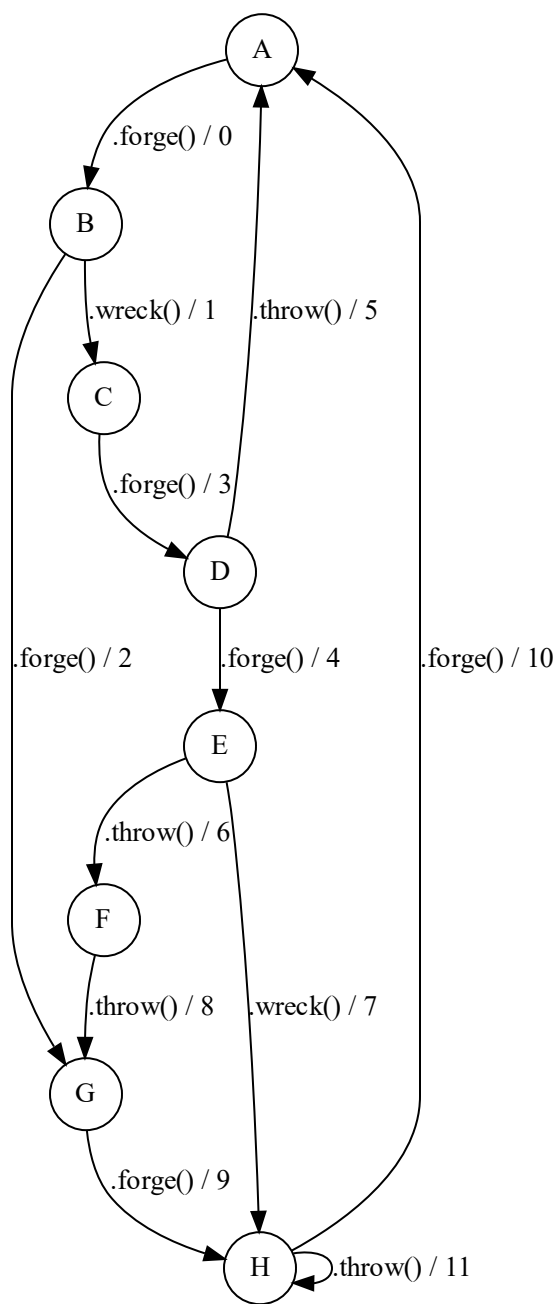
## 2. Двоичные данные:

```
(b'MJI\x05\x00\x00<\x00\x02\x00\x00\x00K\x00\x00\x00\xea\x18\xbf#\x03\xbd'
 b'Q@O\x00M\xe9\x14\xf7\xc8\xc2\x0c\xe5\x02\xb5\xd3?\xea\x1d\x97\xf8\xeeqR'
 b'3\x8e\xec\xa1S\x96\xeaX\x19\xf8bpgomy#\t\x0e\x05\xe8\xf9\xa0\xaa\x10A'
 b'\x00F\x00\xad\xf9\x82+\xba^B\xf9\xd5\x0c')
```

## Результат разбора:

```
{'A1': 'pgomy',
 'A2': [{'B1': 35, 'B2': 3892645385}, {'B1': 48, 'B2': 279617785}],
 'A3': -0.5952440500259399,
 'A4': 1371341603,
 'A5': 64,
 'A6': {'C1': {'D1': 63917, 'D2': [130, 43, 186, 94, 66], 'D3': -7, 'D4': 3285},
        'C2': 339142989,
        'C3': [214092023,
               -743111963,
               -1759647169,
               1383198456,
               -328298957,
               -359246943,
               1660426584]}}
```

**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.wreck()	1
o.forge()	3
o.wreck()	RuntimeError
o.forge()	4
o.throw()	6
o.throw()	8
o.forge()	9
o.throw()	11
o.throw()	11
o.wreck()	RuntimeError
o.forge()	10
o.throw()	RuntimeError
o.forge()	0
o.wreck()	1
o.forge()	3
o.throw()	5
o.forge()	0
o.forge()	2

## 2. Пример использования класса C32:

o = C32()	
o.forge()	0
o.wreck()	1
o.forge()	3
o.throw()	5
o.forge()	0
o.forge()	2
o.throw()	RuntimeError
o.forge()	9
o.throw()	11
o.forge()	10
o.forge()	0
o.throw()	RuntimeError
o.wreck()	1
o.forge()	3
o.forge()	4
o.throw()	6
o.wreck()	RuntimeError
o.throw()	8

## Вариант №6

**Задача 3.1.** Реализовать разбор двоичного формата данных (в духе формата WAD игры Doom или графического формата PNG). Данные начинаются с сигнатуры 0x49 0x4f 0x58, 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	Адрес (uint16) структуры C
	3	Массив uint8, размер 5
	4	int32
	5	Размер (uint32) и адрес (uint16) массива int64

Структура B:	1	int8
	2	uint16

Структура C:	1	int16
	2	Адрес (uint16) структуры D
	3	Адрес (uint32) структуры E
	4	uint64
	5	int64
	6	Размер (uint16) и адрес (uint16) массива int8

Структура D:	1	float
	2	uint64

Структура E:	1	Массив структур F, размер 3
	2	uint64

Структура F:	1	int16
	2	uint64
	3	Размер (uint16) и адрес (uint16) массива int8
	4	int32
	5	uint8

Примеры разбора двоичного формата с помощью функции f31:

### 1. Двоичные данные:

```
(b' I0X\xab\xa0Eu\x00_\x9c \xb4\xee\x16\xc1G_\x03\x00\x00\x00\x91\x006\x08p=\xa4'
b'L\xdf\x84;$\x82\xb1\x8f\x0e\xfd\x00i#\x9b\x88\x8d\xc7\x07\xf2s\x84\xd9\xdem'
b"\x95\xe5'\x03\x00#\x00\xa4\xe4?m\xc6\xbc5\x00\x16\x14\xd5\\\xc3\x03~\x03\x00"
b'&\x00\x91\\\x95\x11\xc5c\x96\xd4\x95H\n\x08\x82\xa9\x04\x00)\x00\x1b9\x07'
b'f5\xb5d\xb3\xa0\x1b\xd9\xa7\xc8\x9d\xa4b\xf4\x9a\xe09\xa2\x8d\x17'
b"\x00-\x00\x00\x00\xd6|t\x98\x90'\x98\xca\xf9X\xb7\xa9%WT\xaa\x07\x00n"
b'\x00\xff\t=T\xd3\xf8\xa7\xae\xd2\xa6\x99\x9d=\x80\xcf\xd7\x04\x82\xde'
b'\xd0\x08Ba`')`
```

Результат разбора:

```
{'A1': {'B1': -85, 'B2': 17824},
'A2': {'C1': -29278,
'C2': {'D1': 0.05860158056020737, 'D2': 12790825729736527012},
'C3': {'E1': [{'F1': -3577,
'F2': 2874868236268700787,
'F3': [-113, 14, -3],
'F4': 1832903844,
'F5': 198},
{'F1': 13756,
'F2': 9080316077188322816,
'F3': [0, 105, 35],
'F4': 295001233,
'F5': 197},
{'F1': 25381,
'F2': 12214333979642680470,
'F3': [-101, -120, -115, -57],
'F4': 1711749403,
'F5': 53}],
'E2': 14458763841379722421},
'C4': 14598461694108007638,
'C5': -6173213369945335559,
'C6': [-99, -92, 98, -12, -102, -32, 57]},
'A3': [95, 156, 32, 180, 238],
'A4': 1598538006,
'A5': [-5861442803464336897, -2895955033250486574, 6944904706013430276]}
```

## 2. Двоичные данные:

```
(b'I0X\xf2\x0b\xe1q\x00\x9f\x8e\xad\xb3\xf7\x9e\xdb>\x9f\x04\x00\x00'
b'\x00\x8d\x00,\xe9\x00\xbe\xcd\x8c\x1aD\xaf\x1e\x05\xf2\xe3ZEz'
b'\x9c\xf2\xcf\x92\x0eM\xed6\x91E\xe6\xd5@\x17\x87\x06\x00#\x00e'
b'\x99\xbc\xc6\xe6\xcd\xc3kx3\xdd(\xd3F\x02\x00)\x00\xf0>\x13\x82r\xc1'
b'+\x11"\xc3\xba\xebv\xfd\xc6\x02\x00+\x00\x18IkG\xfcK\xe1\x85oF2\x82\x93\x9e'
b'\xfe<\xfd\x17\x00-\x00\x00\x00\xe3\x99\x9d\x90)\xce\xfe\xc8\xd1\x82\xa1'
b""\xafl\xfei\x03\x00n\x00]\xc1V\x19\x0e\xaa \xd3\xbb&\x8d\xeb\xde\x1e="
b"\xf0\xbe\x07\r1\xa54\x12\xf8'\t\xac@\xe3\xa4w\xf4")
```

Результат разбора:

```
{'A1': {'B1': -14, 'B2': 57611},
'A2': {'C1': -708,
'C2': {'D1': -0.12588948011398315, 'D2': 368924926527638733},
'C3': {'E1': [{'F1': -4787,
'F2': 9734320407020146998,
```

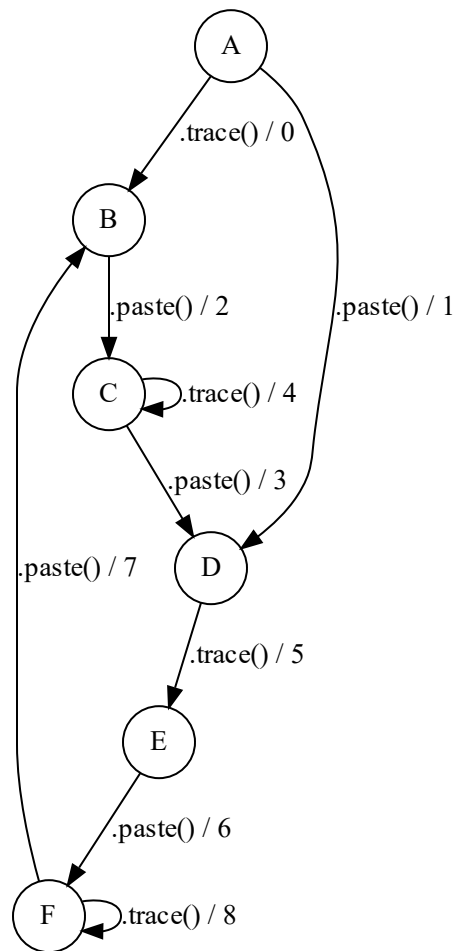


```

        'F3': [-14, -29, 90, 69, 122, -100],
        'F4': -960718491,
        'F5': 230},
    {'F1': -15411,
     'F2': 5103467733257317240,
     'F3': [-14, -49],
     'F4': -2112667920,
     'F5': 114},
    {'F1': 11201,
     'F2': 14338747543440269841,
     'F3': [-110, 14],
     'F4': 1198213400,
     'F5': 252}],
    'E2': 9381638418567613295},
    'C4': 14483240129585322467,
    'C5': 7637661517606912721,
    'C6': [-109, -98, -2]},
    'A3': [159, 142, 173, 179, 247],
    'A4': -1623270498,
    'A5': [-3233397554920636067,
            -1135717588244027717,
            -571336318625445954,
            -831014310277019353]}

```

**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.trace()           0
o.trace()           RuntimeError
o.paste()           2
o.paste()           3
o.trace()           5
o.paste()           6
o.trace()           8
o.paste()           7
o.paste()           2
o.trace()           4
o.trace()           4
o.trace()           4

```

2. Пример использования класса C32:

o = C32()	
o.paste()	1
o.trace()	5
o.paste()	6
o.paste()	7
o.paste()	2
o.trace()	4
o.paste()	3
o.trace()	5
o.paste()	6
o.trace()	8
o.paste()	7
o.trace()	RuntimeError
o.paste()	2

## Вариант №7

**Задача 3.1.** Реализовать разбор двоичного формата данных (в духе формата WAD игры Doom или графического формата PNG). Данные начинаются с сигнатуры 0x69 0x52 0x4c 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	int32
	2	Размер (uint32) и адрес (uint32) массива адресов (uint16) структур B
	3	Адрес (uint32) структуры C

Структура B:	1	Массив char, размер 7
	2	uint16

Структура C:	1	Структура D
	2	int32
	3	uint16
	4	uint8
	5	int16

Структура D:	1	uint32
	2	Массив int8, размер 7
	3	Размер (uint32) и адрес (uint32) массива int32

Примеры разбора двоичного формата с помощью функции f31:

1. Двоичные данные:

```
(b'iRLP\xdf\xa1\\\x93\x02\x00\x00\x00&\x00\x00\x00:\x00\x00\x00awokxsu\xa8Iraa'  
b'eyez\x06\xa5\x14\x00\x1d\x00\xd3\xeb\xa1\xd9m\xf8\xbf\x16\x01e`\x08\x82'  
b'\xb5V\xfcY\xb9F\x9f\x83\x1b\xe3\x82\xbf\xba\x04\x00\x00\x00*\x00\x00'  
b'\x00o7\x05\xd5.\xfaR\xf16')
```

Результат разбора:

```
{'A1': -1822645793,  
'A2': [{'B1': 'awokxsu', 'B2': 18856}, {'B1': 'raaeyez', 'B2': 42246}],  
'A3': {'C1': {'D1': 1186552316,  
              'D2': [-97, -125, 27, -29, -126, -65, -70],  
              'D3': [-643699757, 1941960813, 1617232150, 1454735880]},  
       'C2': -721078417,  
       'C3': 64046,  
       'C4': 82,  
       'C5': 14065}}
```

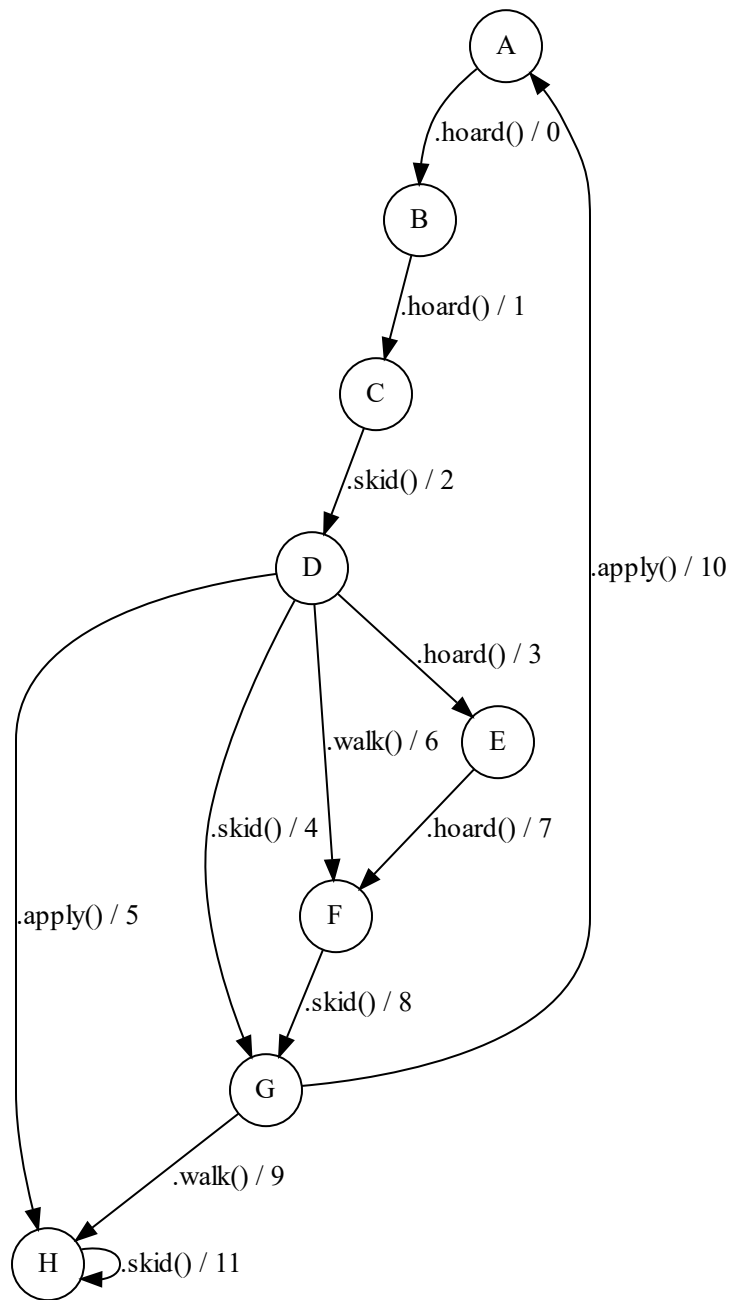
2. Двоичные данные:

```
(b'iRLP\t\x99\xcc\xa8\x04\x00\x00\x008\x00\x00H\x00\x00vzzyuazu\x0bigo'
b'gncs>\xa4dxomkre\x91\xderzlnedq\xf6\xdb\x14\x00\x1d\x00&\x00/\x00'
b'\xa6\x13\x0c\x8d\xd86\xfc\xcd)\xe2g/\x1d\xff\nwF\xee\x9f\x02\x00\x00\x00@'
b'\x00\x00\x00{\r\x13\xd8\xa0\xf0\x01<X')
```

Результат разбора:

```
{'A1': -1462986487,
'A2': [{'B1': 'vzzyuaz', 'B2': 2933},
        {'B1': 'igogncs', 'B2': 42046},
        {'B1': 'dxomkre', 'B2': 56977},
        {'B1': 'rzlnedq', 'B2': 56310}],
'A3': {'C1': {'D1': 795337257,
               'D2': [29, -1, 10, 119, 70, -18, -97],
               'D3': [-1928588378, -839108904]},
        'C2': -669840005,
        'C3': 61600,
        'C4': 1,
        'C5': 22588}}
```

**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.hoard()	0
o.hoard()	1
o.skid()	2
o.walk()	6
o.skid()	8
o.apply()	10
o.hoard()	0
o.hoard()	1
o.skid()	2
o.hoard()	3
o.hoard()	7
o.skid()	8
o.walk()	9
o.hoard()	RuntimeError
o.skid()	11
o.skid()	11

## 2. Пример использования класса C32:

o = C32()	
o.hoard()	0
o.skid()	RuntimeError
o.hoard()	1
o.skid()	2
o.hoard()	3
o.walk()	RuntimeError
o.hoard()	7
o.skid()	8
o.apply()	10
o.hoard()	0
o.hoard()	1
o.skid()	2
o.skid()	4
o.walk()	9
o.apply()	RuntimeError
o.skid()	11
o.skid()	11
o.skid()	11

## Вариант №8

**Задача 3.1.** Реализовать разбор двоичного формата данных (в духе формата WAD игры Doom или графического формата PNG). Данные начинаются с сигнатуры 0x57 0x4d 0x58, 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	int8
	2	uint16
	3	uint32
	4	Структура В
	5	Размер (uint32) и адрес (uint32) массива char
	6	double
	7	int64
	8	Структура D

Структура В:	1	uint8
	2	uint16
	3	float
	4	Размер (uint32) и адрес (uint16) массива адресов (uint16) структур С
	5	uint32
	6	int64
	7	Массив char, размер 7
	8	double

Структура С:	1	uint16
	2	float

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

Примеры разбора двоичного формата с помощью функции f31:

### 1. Двоичные данные:

```
(b'WMXYTM&N\b6n0\b7\b3>\xceF\xaf\x00\x00\x00\x02\x00r\xe8;[!\xfd'
b'\xb6\b86r\xc6I\x8a]cdz mhjz?\xea\x02p\xfdNP\x0c\x00\x00\x00\x03\x00\x00'
b'\x00v?\xdc6\xc2\xdf&C\xdc;D\xdd\tB$\xb2.;9\x9b\x13D\x8f\xb1_\x00\x00'
b'\x00\x05\x00\x00\x00y\x81A\xed\xe6\xd7\xa8?^\xed\x00\xf7\xbd\x0b\x84'
b'\xbe\xe4\xe7\x81\xc4n?M6X\x00f\x00liwq\xaddi\xa2\xbcW\t\xac\n3 %\xc6Rb'
b'\xe1\xe7>\xbe\xc2\x16\xa2K\xbesEI\xdb+\x04Ln\xddW\x93\x83\x0cjDb' )
```

Результат разбора:



```
{'A1': 89,
'A2': 21581,
'A3': 642692718,
'A4': {'B1': 48,
      'B2': 47027,
      'B3': 0.40288302302360535,
      'B4': [{'C1': 2948, 'C2': -0.44707873463630676},
              {'C1': 50286, 'C2': 0.8016104698181152}],
      'B5': 3896204065,
      'B6': -164796508841342371,
      'B7': 'cdznhjz',
      'B8': 0.8127980182057413},
'A5': 'iwq',
'A6': 0.4408423594659874,
'A7': 4270781378488218158,
'A8': {'D1': 4267612628967207263,
      'D2': [-5952516659811513940,
              734966510804689633,
              -1783778661579469890,
              8306126293423115374,
              -2497365277295360926],
      'D3': -32447,
      'D4': 3991328680,
      'D5': 0.8708038330078125,
      'D6': 63421}}
```

## 2. Двоичные данные:

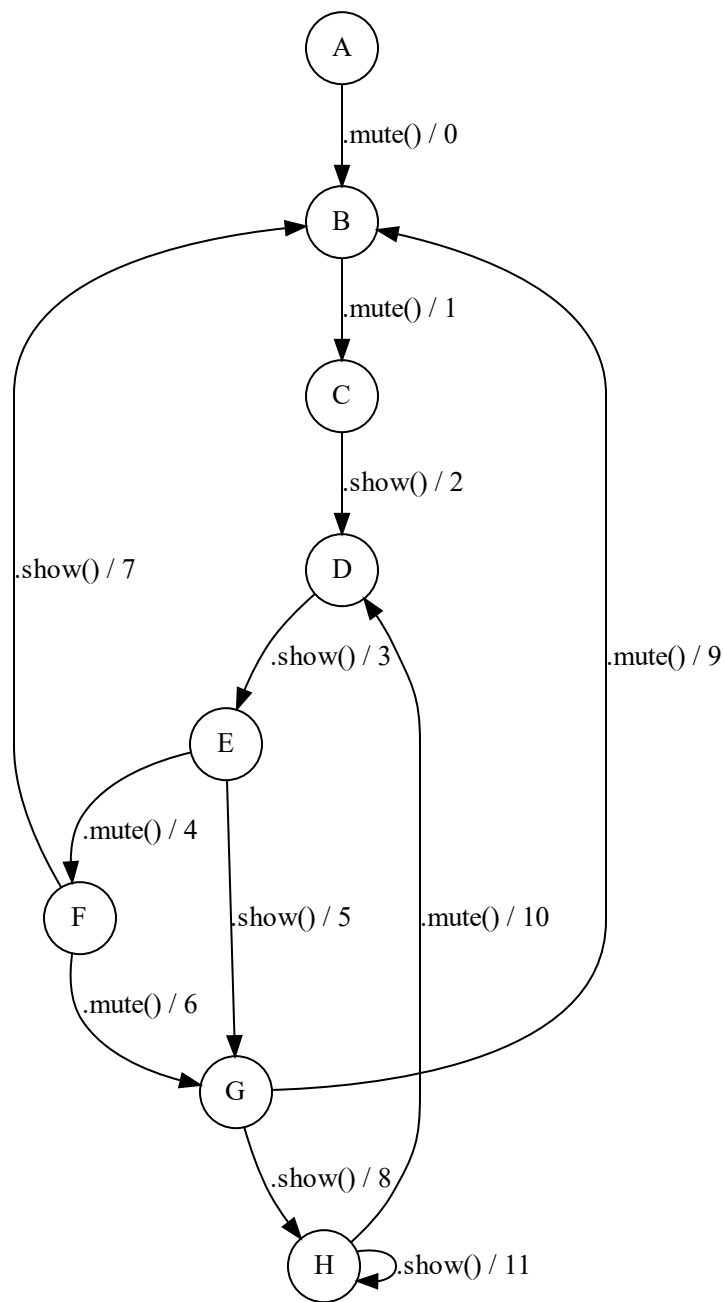
```
(b'WMX\xe2\xc4\xab\xeam;\xa4\xcb\xfc\xc5?! \xda\xd7\x00\x00\x00\x02\x00rv'
b'\x88\xebF8\xd0\xc0S5p-cr sxjxi\xbf\xdb"M\xb9\xcc\xd8\xc8\x00\x00'
b'\x00\x04\x00\x00\x00v\xbf\xe1\xf1\x13N\xe2\xd5\x1e\x1c\x89\xec\xd3I_'
b'\xba\x9a\xbf~\x8f\xdd%\xf6\xfc\x00\x00\x00\x04\x00\x00\x00z\x80\xb0'
b'9\x81V\x93=-\xf8\xb4;\x02A8\xbeD\x8d\x11\xaa\xa5\xbe\xe8{\xb4\x00f\x00lwg'
b'ah~\x07[\x91\x13\xb5W\xc1kg\xd6g\x059Z\xab\x918\x15\xbd\xba\xc6v\xe5\x06b'
b'\x9e\x9d\xc1\xed\xe3\xb9')
```

## Результат разбора:

```
{'A1': -30,
'A2': 50347,
'A3': 3933027236,
'A4': {'B1': 203,
      'B2': 64709,
      'B3': 0.6322454810142517,
      'B4': [{'C1': 16696, 'C2': -0.19194437563419342},
              {'C1': 43685, 'C2': -0.45406877994537354}],
      'B5': 3896204065,
      'B6': -164796508841342371,
      'B7': 'cdznhjz',
      'B8': 0.8127980182057413},
'A5': 'iwq',
'A6': 0.4408423594659874,
'A7': 4270781378488218158,
'A8': {'D1': 4267612628967207263,
      'D2': [-5952516659811513940,
              734966510804689633,
              -1783778661579469890,
              8306126293423115374,
              -2497365277295360926],
      'D3': -32447,
      'D4': 3991328680,
      'D5': 0.8708038330078125,
      'D6': 63421}}
```

```
'B5': 1988638955,  
'B6': 5060023705799061549,  
'B7': 'crsxjxi',  
'B8': -0.42396872658299545},  
'A5': 'wgah',  
'A6': -0.5606781521552511,  
'A7': 2056435097084607130,  
'A8': {'D1': 5674393211865790204,  
        'D2': [9081327852274931649,  
                7739390222617107115,  
                -7982606434887633179,  
                460104512345859001]},  
'D3': -32592,  
'D4': 964777619,  
'D5': 0.04247350990772247,  
'D6': 15106}}
```

**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.mute()           0
o.mute()           1
o.show()            2
o.show()            3
o.mute()            4
o.show()            7
o.mute()            1
o.show()            2
o.show()            3
o.mute()            4
o.mute()            6
o.show()            8
o.mute()            10
o.show()            3
o.show()            5
o.mute()            9

```

## 2. Пример использования класса C32:

```

o = C32()
o.mute()           0
o.mute()           1
o.show()            2
o.show()            3
o.mute()            4
o.mute()            6
o.show()            8
o.show()            11
o.mute()            10
o.show()            3
o.show()            5
o.show()            8
o.mute()            10
o.show()            3
o.show()            5
o.mute()            9

```



```

        {'C1': 0.9622386693954468, 'C2': -0.48759850713151787}],
        'B2': 'ae'},
    'A2': 8111,
    'A3': [35, 97],
    'A4': 2591225017,
    'A5': {'D1': 0.7797323695602669,
           'D2': 124,
           'D3': 15035919146454129143,
           'D4': -2330734801136487483,
           'D5': 0.0966031476855278,
           'D6': -644178389,
           'D7': 114},
    'A6': 6,
    'A7': [72, 254, 207, 81, 76, 60, 106],
    'A8': 0.4059494963015424}

```

## 2. Двоичные данные:

```

(b'\x82WCRP\x03\x00\x00\x00m\x00\x02\x00y\x00\xa7\xb9\x071)\x12\\\xa3\xe0'
b"\x0c\xdd'\xa3\x8e\xc2\xbf\xed\x8e\xc3g\xe4\xbf=\xafM\xb28\x98\x96\xb9!\xb0V"
b'\xeb\xd85\xbf\xa1\xbb\x8f\x1a\xc7y'\xd3\xfb\xedXN\xfdL3\x9d{\x8cw\xd8'
b'?j\x9e\x97\xbe\xe0_\xdc-\x03\x84\xe2?V\xa8i\xbfvJ\x82\xbcm1\xe2?\x9e\xe3i'
b'=\xb8\xa71\xb5\xc3m\xe9\xbfI\x00\x00\x00U\x00\x00\x00a\x00\x00\x00sh')

```

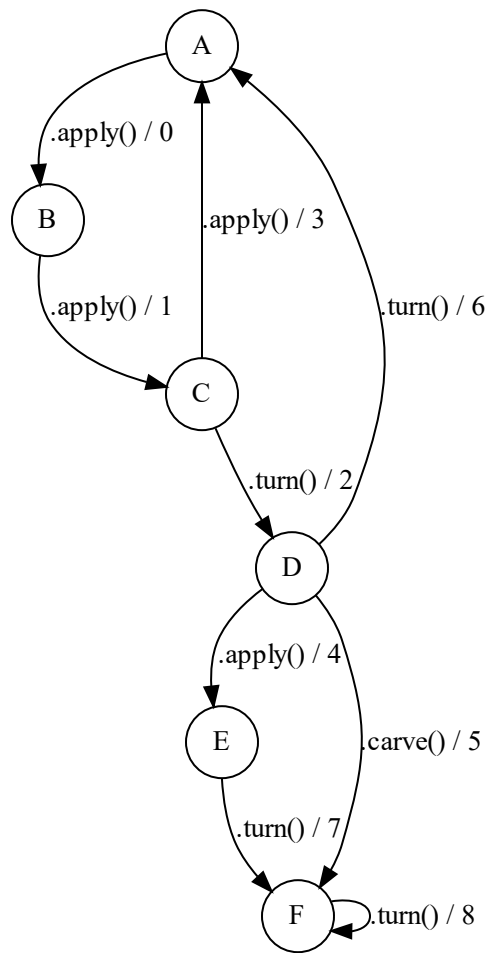
### Результат разбора:

```

{'A1': {'B1': [{'C1': -0.2961304783821106, 'C2': 0.5786147971839419},
               {'C1': -0.9127248525619507, 'C2': 0.568533771688023},
               {'C1': 0.057101838290691376, 'C2': -0.7946489848084104}],
        'B2': 'sh'},
    'A2': 47527,
    'A3': [7, 49],
    'A4': 2740720169,
    'A5': {'D1': -0.14497794577082512,
           'D2': 237,
           'D3': 5597760756224869262,
           'D4': 6246529764142102706,
           'D5': -0.7103411555290222,
           'D6': 445627297,
           'D7': 199},
    'A6': 121,
    'A7': [96, 211, 251, 237, 88, 78, 253],
    'A8': 0.38229667731939254}

```

**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.apply()           0
o.apply()           1
o.apply()           3
o.apply()           0
o.apply()           1
o.turn()            2
o.turn()            6
o.apply()           0
o.turn()            RuntimeError
o.apply()           1
o.turn()            2
o.apply()           4
o.turn()            7
  
```

### 2. Пример использования класса C32:

o = C32()	
o.apply()	0
o.apply()	1
o.apply()	3
o.apply()	0
o.turn()	RuntimeError
o.apply()	1
o.turn()	2
o.apply()	4
o.turn()	7
o.apply()	RuntimeError
o.turn()	8
o.turn()	8
o.turn()	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 0x41 0x55 0x4c 0x41 0xb1, 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	int32

Структура B:

1	int32
2	int64
3	uint64
4	Структура C
5	uint8
6	float
7	uint8
8	int32

Структура C:

1	Массив адресов (uint16) структур D, размер 2
2	Массив float, размер 8
3	uint8
4	Массив int16, размер 8

Структура D:

1	Массив uint16, размер 3
2	int32
3	uint64

Примеры разбора двоичного формата с помощью функции f31:

### 1. Двоичные данные:

```
(b'AULA\b1/\x00s\x04\xf2\x91g\xc4\xfb:\xff \xe9\x1b1\xcc\xddP\x19\xbb<\xe3'
b'\xbcr iS\x19K\xba\xae!\xbe\x920=\xba\x13\xb8\xa7\xa3sZ\xc0{?Q\xbe"\x1cK'
b'\x8d\x0f\xe3x7\xee\xcbn;\xb2H\x0b\x00\x1d\x00\x83f\x95\xbebg[\xbe\x0b'
b'\x07Z\xbe <\x06\xbf\n\xd9\xf2>\xe6 +\xbe(-)\xbf\\x06/\xbe\x9b"\x99\xba\x93'
b'\xac V\x1c\xa45\xa7C$jx\bc\x96o\xf2\x97=#\xa63\xaa\x81')
```

Результат разбора:

```
{'A1': {'B1': 1065074778,
        'B2': -2085292748715147695,
        'B3': 5238314663639136120,
        'B4': {'C1': [{'D1': [50279, 15099, 8447],
                        'D2': -865330199,
                        'D3': 13610788962306707677}],
```

```

        {'D1': [26994, 6483, 47691],
         'D2': -1833033298,
         'D3': 8332688144612343088}},
    'C2': [-0.2917977273464203,
           -0.21426156163215637,
           -0.2129174917936325,
           -0.5243549346923828,
           0.4743121266365051,
           -0.16711768507957458,
           -0.8639702796936035,
           -0.17092269659042358],
    'C3': 155,
    'C4': [-26334, -27718, 8364, 7254, 13732, 17319, 27172, -17288]},
    'B5': 150,
    'B6': 0.07419287413358688,
    'B7': 35,
    'B8': -2119552090},
    'A2': -1846410125}

```

## 2. Двоичные данные:

```

(b'AULA\xb1/\x00\xf2\xaf<\x9f9\xce\xa6\xa3\xaeb/\xe8c\xd4\x10\x87\xc5(+\xc5\xed'
 b'\xd6\xe9\tk\xae\xabp\x84\xa4r\xa4\x9cGr0\x87\xa2R\x01\xcczw&\xe6K\xcd\xc1U'
 b'+P\x98_\xdc\xf5XCx?p\x0b\x00\x1d\x00-\xdc\x15<\xa8\x00A?\xa5\x9c\x97>\xad'
 b'\xa4u\xbf\x86\x8a\x1e?1Fu\xbf\x15\xbb\?\x8e\xfe\xf4>\xc5\xd2\x9a\xe5\xe'
 b'\x18\x85\xe9\xbe\xfc\xe7\xee\xf5\xdb\x3Q9ni\x10S>v\x86\x14u\xb7')

```

## Результат разбора:

```

{'A1': {'B1': 645364428,
        'B2': -7471424134483981338,
        'B3': 8088315686431349855,
        'B4': {'C1': [{'D1': [52793, 41894, 25262],
                        'D2': -731650001,
                        'D3': 15487251482755303184},
                  {'D1': [2537, 44651, 28843],
                        'D2': -1535990652,
                        'D3': 95317244165375900}]},
        'C2': [0.009146732278168201,
                0.7539162635803223,
                0.29611697793006897,
                -0.9595440030097961,
                0.6193011999130249,
                -0.958102285861969,
                0.8622296452522278,
                0.47850459814071655],

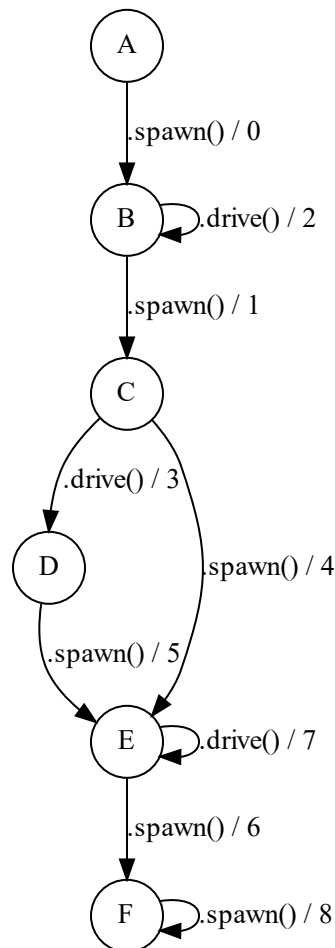
```

```

'C3': 197,
'C4': [-25902,
      -28955,
      -31464,
      -16663,
      -6148,
      -2578,
      -11301,
      14673]},
'B5': 110,
'B6': 0.20611728727817535,
'B7': 118,
'B8': -1217063802},
'A2': -1623412750}

```

**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.spawn()	0
o.drive()	2
o.drive()	2
o.drive()	2
o.spawn()	1
o.drive()	3
o.spawn()	5
o.drive()	7
o.spawn()	6
o.spawn()	8

2. Пример использования класса C32:

o = C32()	
o.drive()	RuntimeError
o.spawn()	0
o.drive()	2
o.drive()	2
o.spawn()	1
o.drive()	3
o.spawn()	5
o.drive()	7
o.spawn()	6
o.drive()	RuntimeError
o.spawn()	8

## Вариант №11

**Задача 3.1.** Реализовать разбор двоичного формата данных (в духе формата WAD игры Doom или графического формата PNG). Данные начинаются с сигнатуры 0x53 0x45 0x42 0x68, 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	Адрес (uint16) структуры В
	3	int8
	4	int8
	5	Размер (uint16) и адрес (uint32) массива структур Е

Структура В:	1	int8
	2	Адрес (uint32) структуры С
	3	int16
	4	int8

Структура С:	1	double
	2	Массив char, размер 6
	3	uint32
	4	Структура D
	5	int16

Структура D:	1	Размер (uint16) и адрес (uint16) массива uint16
	2	uint8
	3	uint16
	4	uint8

Структура Е:	1	Размер (uint32) и адрес (uint32) массива int64
	2	Массив uint8, размер 3
	3	int8
	4	int32

Примеры разбора двоичного формата с помощью функции f31:

### 1. Двоичные данные:

```
(b'SEBh\xbe\x003\xe4\x85\x00\x02\x00\x00\x00c\xae|\xd8\x08U\x9f\xf4t\xbf'  
b'\xe0\xce\xa7\x93c\xa2\xecbdf1d1\x87\x15,r\x00\x04\x00\x0f\xff\x9c\xa9'  
b'\xb8\xf0\xf5{\x00\x00\x00\x17\x13&I\xc8\xc2\xae\xb0\xce-o^Qi\xa3\xdfX'  
b'\xca\x04+\x02\xc5\xf2\x18=-C0\n\xe6\xd7"L\xe5&\x07{^\xe1d=}\xe5<\x00'  
b'\x00\x00\x02\x00\x00\x00;g+\x00`\xf6\xd3>}\x00\x00\x00\x03\x00\x00\x00K\x0c'  
b'\x00#V!\xa3\x8b#')
```

Результат разбора:

```
{'A1': 190,
'A2': {'B1': 123,
      'B2': {'C1': -0.5252263907142072,
            'C2': 'bdf1d1',
            'C3': 2266311794,
            'C4': {'D1': [44668, 55304, 21919, 62580],
                  'D2': 255,
                  'D3': 40105,
                  'D4': 184},
            'C5': -3851},
      'B3': 4902,
      'B4': 73},
'A3': -28,
'A4': -123,
'A5': [{'E1': [-3980427046245535906, 5866400169289450539],
      'E2': [103, 43, 0],
      'E3': 96,
      'E4': -153928067},
      {'E1': [199831944412676943, 785551727341676039, 8889790535120577852],
      'E2': [12, 0, 35],
      'E3': 86,
      'E4': 564366115}]}
```

## 2. Двоичные данные:

```
(b'SEBh\x92\x007\xf2\x9e\x00\x02\x00\x00\x00g\xf2\\pW\x0c\xcbBn\x8bP#C\xbf'
b'\xe9\xd7\xb8\x98\xe3nkaxbbc\xc5\xceDt\x00\x06\x00\x0fXQ:\x1c6\x93_'
b'\x00\x00\x00\x1b\x11R3\x85\x0f\xae?\xe9;\x1e\xc2\xff\x08\xc8\x99\x82'
b'\xc6\x93\xafGc\x14\x932\xb0\x0f\x83\x14\xe8\x8b\x90t7s\x86\xa5\xab\x1ck\xc5'
b'\xa6\xb4Y\x00\x00\x00\x03\x00\x00\x00?\x8e\x10\xdds\x983\xbe\xfb\x00'
b'\x00\x00\x02\x00\x00\x00W\xf7:U\xc5\xa8Rc\x82')
```

## Результат разбора:

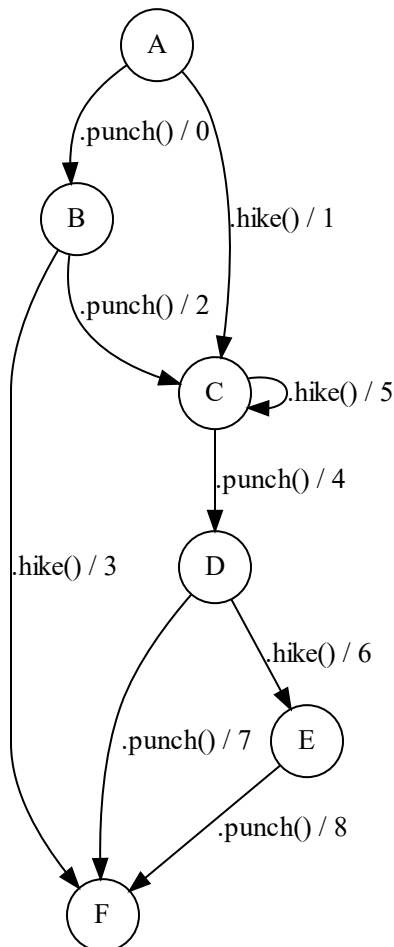
```
{'A1': 146,
'A2': {'B1': 95,
      'B2': {'C1': -0.8075255017675838,
            'C2': 'kaxbbc',
            'C3': 3318629492,
            'C4': {'D1': [62044, 28759, 3275, 17006, 35664, 9027],
                  'D2': 88,
                  'D3': 20794,
                  'D4': 28},
            'C5': 13971},
      'B3': 4434,
      'B4': 51},
```

```

'A3': -14,
'A4': -98,
'A5': [{'E1': [-8858670352495337790, -69585232574639185, 5143977821830385539],
        'E2': [142, 16, 221],
        'E3': 115,
        'E4': -1741439237},
        {'E1': [1506607527896773510, -6509077587230804903],
        'E2': [247, 58, 85],
        'E3': -59,
        'E4': -1470995582}]]

```

**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.punch()          0
o.punch()          2
o.hike()           5
o.punch()          4
o.hike()           6
o.hike()           RuntimeError
o.punch()          8
```

## 2. Пример использования класса C32:

```
o = C32()
o.punch()          0
o.punch()          2
o.hike()           5
o.punch()          4
o.hike()           6
o.punch()          8
```



## Вариант №12

**Задача 3.1.** Реализовать разбор двоичного формата данных (в духе формата WAD игры Doom или графического формата PNG). Данные начинаются с сигнатуры 0xb6 0x50 0x46 0x51 0x4d, 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	int64
	2	Массив float, размер 5
	3	uint32
	4	uint16
	5	Структура B
	6	Размер (uint32) и адрес (uint16) массива адресов (uint16) структур C
	7	Размер (uint32) и адрес (uint16) массива int64
	8	Структура D

Структура B:	1	float
	2	Массив char, размер 5

Структура C:	1	uint64
	2	double

Структура D:	1	uint8
	2	int32
	3	int64

Примеры разбора двоичного формата с помощью функции f31:

### 1. Двоичные данные:

```
(b'\xb6PFQM\xe9\xf3\xdetb\xbf\x955\xbf{\x03\x9d\xbf\c\x94\xbc\x9aJ\xa7\xbeJA'  
b'\xc5\xbfGo\x11@\xde\x1d\x95\xa3\xbe\x80\xf4\xcaugkjc\x00\x00\x00\x02'  
b'\x00i\x00\x00\x00\x02\x00m\xe2\x87^R\xcb\xd7\xa5\x18\x83\x80\xd2 \xa4_N\xd9'  
b'2\xaeF\xa9@?\xe0\x02W\xce$ap\xc0\xaa]8\x17Z=??\xe2\xfb\xc6\x18r\x11'  
b'\x98\x00I\x00Y9\x01WiJ\x9e\xbc/\xf27\x81\x99\x82T\xde\x1c')
```

Результат разбора:

```
{'A1': -1588681652077357771,  
'A2': [-0.9805238842964172,  
-0.8608944416046143,  
-0.018834425136446953,  
-0.1975165158510208,  
-0.7790384888648987],  
'A3': 1088307229,  
'A4': 38307,
```

```
'A5': {'B1': -0.2518675923347473, 'B2': 'ugkjc'},
'A6': [{'C1': 6867665293482043712, 'C2': 0.5002860094273682},
        {'C1': 13883011296814316863, 'C2': 0.593234107728267}],
'A7': [4107660445410573359, -993182696495325668],
'A8': {'D1': 226, 'D2': -2023861557, 'D3': -2907891031316750172}}
```

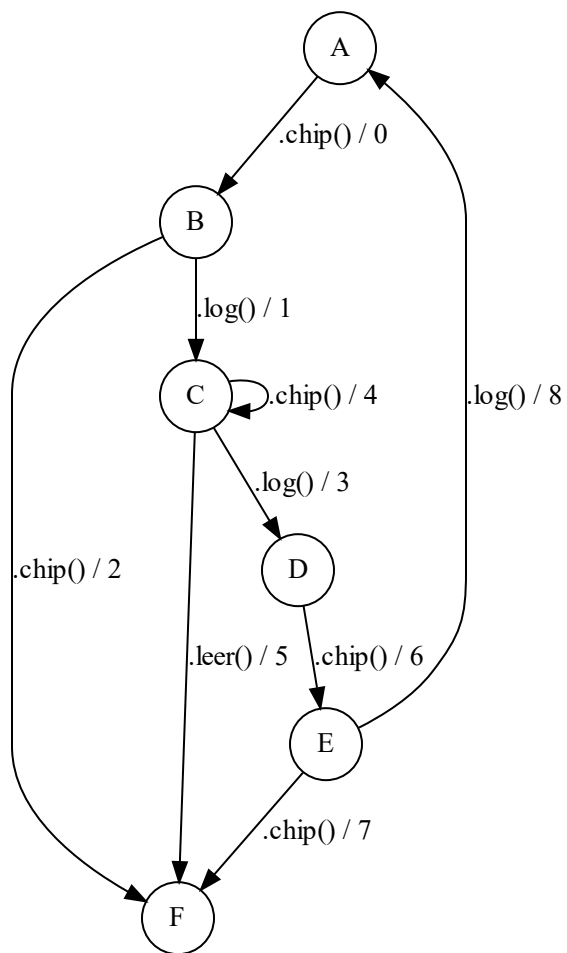
## 2. Двоичные данные:

```
(b'\xb6PFQM\x2t\xa8\xf9;\x13S?f\xd5\xad?/\x8b]\xbf\x7f\x86\x95?G\xb50?A;'
b'13\xba\x06g16\xbe\xb0c\x9ekffas\x00\x00\x00\x02\x00i\x00\x00\x00\x07\x00m'
b'\x195\xe9\x7f6=\x7f|\x07\xd4\xe6\xefV\xe3\xb4\xe1;_\xc0\xdf\xc0?\xdbz'
b'\x8e`<\x9f\x80\xf1h\x9f/B+%\xd6?\xde\xff\xca@\xe2\x85\xf8\x00I\x00Y\xa3\xcam'
b'\x12hG\x06x\x9e\x8cd\x9e\x96]\xa8t\xa1\x14\xa8\xd5)\xb7\xbf\x02\xf9\xbeV'
b'\xc4\xfe \xdb`)\x1b\rZ\x11f\x06\xea\xac\xc9^0\x0f\x91K\r\xda\xe1\x03'
b'\xa9\x8b3\xd1\xcf')
```

## Результат разбора:

```
{'A1': 3148707358570124115,
'A2': [0.9016979336738586,
        0.6857202649116516,
        -0.9981473088264465,
        0.7801084518432617,
        0.7548094391822815],
'A3': 867829351,
'A4': 12598,
'A5': {'B1': -0.344510018825531, 'B2': 'kffas'},
'A6': [{'C1': 16407986987543355328, 'C2': 0.42935523412824494},
        {'C1': 17395328586040944086, 'C2': 0.4843621858254603}],
'A7': [-6644378374405421448,
        -7022127086687115148,
        -6839656300587532542,
        -450827508613325984,
        2961975860416349930,
        -5996157735210497267,
        -2674852676961578545],
'A8': {'D1': 25, 'D2': 904494902, 'D3': 4431396931434377046}}
```

**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.chip()           0
o.log()            1
o.chip()           4
o.log()            3
o.chip()           6
o.log()            8
o.log()            RuntimeError
o.chip()           0
o.chip()           2

```

2. Пример использования класса C32:

<code>o = C32()</code>	
<code>o.log()</code>	<code>RuntimeError</code>
<code>o.chip()</code>	<code>0</code>
<code>o.log()</code>	<code>1</code>
<code>o.chip()</code>	<code>4</code>
<code>o.log()</code>	<code>3</code>
<code>o.chip()</code>	<code>6</code>
<code>o.log()</code>	<code>8</code>
<code>o.log()</code>	<code>RuntimeError</code>
<code>o.chip()</code>	<code>0</code>
<code>o.chip()</code>	<code>2</code>

## Вариант №13

**Задача 3.1.** Реализовать разбор двоичного формата данных (в духе формата WAD игры Doom или графического формата PNG). Данные начинаются с сигнатуры 0x42 0x4f 0x5a 0x57, 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	int32
	2	Размер (uint32) и адрес (uint16) массива char
	3	Массив структур B, размер 2
	4	uint16
	5	Размер (uint32) и адрес (uint16) массива uint64
	6	float
	7	Размер (uint32) и адрес (uint32) массива int16
	8	float

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

Структура C:	1	Массив int32, размер 6
	2	int32

Структура D:	1	int32
	2	uint32
	3	uint32

Примеры разбора двоичного формата с помощью функции f31:

### 1. Двоичные данные:

```
(b'BOZWA\x14\x9d\x1a\x02\x00\x00\x00Z\x00\\\x00\x00\x00\x00\xbeE\xfeM\x8e'
b'\x82\xfer\xba\xcb\xfa2|\xe0\xa0\x04\xe3\x80\x93\x04\xbf\x00\x00\x002"}W'
b'ti\xc1&B]{d\x9c\xa9\xb2\x07\xb1\xd8\xceV\xe8\xbf\xb2\xab\x02\x00\x00\x00'
b'\x94\x00$\xb2\x18?\x02\x00\x00\x00\xa4\x00\x00\x00\xf9\xac\xb3\xbcjr'
b'\xb1\x8c\x0b\x9e\xa9\x921\xd5\x89\xaf\x02\x88h\xa9\\=\x891{\xe0\xfd\xe1gu'
b'\x97\xe1w6\xce\\\xdfewF\x07\xd8\xf2\xd1\x01bL\xd1\xc0{\x072\xee\xaa'
b'a\xff\xcb:\xe3!\x9aRb\x0c\xb6\xde\xban\x12\xf4\xec\x97\xa3\xb7p\xa5\xd2\xbf'
b'\xb5J\x16N')
```

Результат разбора:

```
{'A1': 446501953,
'A2': 'jr',
'A3': [{'B1': {'C1': [-1643410255,
```

```

-718171479,
-2013089911,
1029482856,
-528797303,
1969742333],
'C2': 913826199},
'B2': -16896,
'B3': {'D1': -1905394107, 'D2': 3128098434, 'D3': 2091054283},
'B4': -0.16075144848696876},
{'B1': {'C1': [1709137102,
-670611849,
1644286450,
2076234060,
-1427230201,
986447713],
'C2': 1385832931},
'B2': 8754,
'B3': {'D1': 1769232253, 'D2': 1564616385, 'D3': 2845598843},
'B4': -0.7605966789490053}],
'A4': 43954,
'A5': [17587241243714653282, 13822292109889345516],
'A6': 0.5964682102203369,
'A7': [19125, 19990],
'A8': -0.021933065727353096}

```

## 2. Двоичные данные:

```

(b'B0ZW#\x91\x81r\x02\x00\x00\x00Z\x00\\\x00\x00\x00\x10?7SF(\xd24c\xf2'
b'\xd2\xe3_w\x16\x8dDE\xe1M\xed?x\x00\x00\x00\xe4\x9e\xe2\xd7\xf5\x07E!'
b'0\xca\xdb\x0ef38\x8d\x97Ve\x9e\xc1\xbf\x993\x05\x00\x00\x00\x94\x00\x98\xe9'
b'T\xbf\x02\x00\x00\x00\xbc\x00\x00\x00\xfb\x9G\xbel\xba\x9f:\xb6'
b'\xe8\x1c\x98t\xe4\x0e\x82m\xa3@\xe1eP\xd0\x00\xa8ow\x06\xc1\xa7\x0c1\xf4w$80'
b'\x10Y\x9eL\t\x14\xd4Jw\xb00r\xd2\xbf$\xf0\x1b,\x96\t\xcd2`\x82\x92\x1f\x90N'
b'\xb7\x94\xbc\xa5\xe5\xe4\xcb/!/c|\xc3\xbf a2 \xfe\xae\x04\xfa\xa4\xbf'
b'f\xec\x07\x90\x0b+\x04\x8c\xfb\x8f\x8a\x1e\xba~>N')

```

## Результат разбора:

```

{'A1': 1921093923,
'A2': '1x',
'A3': [{'B1': {'C1': [-1237672006,
1956125928,
1837240036,
1709260963,
-1476341680,
-1056540817],

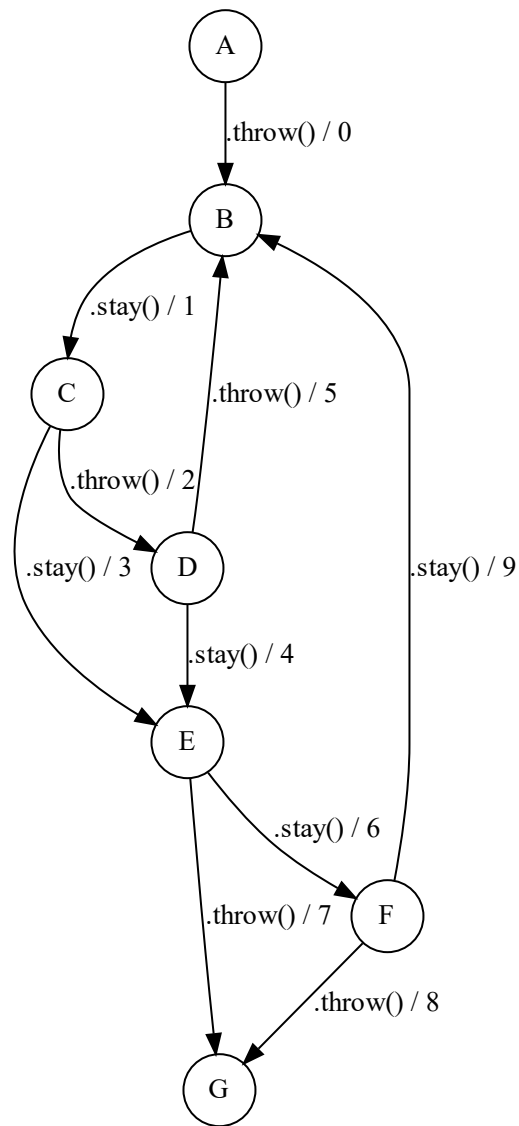
```

```

        'C2': -198112089},
    'B2': 16144,
    'B3': {'D1': 675697463, 'D2': 4066587858, 'D3': 1465902034},
    'B4': 0.915756831444386},
    {'B1': {'C1': [808985719,
                    1285445904,
                    1255412745,
                    1915793527,
                    -266027054,
                    160836635],
            'C2': -2107624755},
    'B2': -24860,
    'B3': {'D1': 133552098, 'D2': 3392151877, 'D3': 862326491},
    'B4': -0.13764635780453616}],
    'A4': 13209,
    'A5': [11942583826897706898,
           7146967621048198373,
           12609551407444378492,
           10378523792344676868,
           2200729678472555275],
    'A6': -0.8316893577575684,
    'A7': [32442, 20030],
    'A8': -0.19528953731060028}

```

**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.stay()</code>	<code>RuntimeError</code>
<code>o.throw()</code>	0
<code>o.stay()</code>	1
<code>o.throw()</code>	2
<code>o.throw()</code>	5
<code>o.stay()</code>	1
<code>o.stay()</code>	3
<code>o.stay()</code>	6
<code>o.stay()</code>	9
<code>o.stay()</code>	1
<code>o.throw()</code>	2
<code>o.stay()</code>	4
<code>o.throw()</code>	7

## 2. Пример использования класса C32:

<code>o = C32()</code>	
<code>o.stay()</code>	<code>RuntimeError</code>
<code>o.throw()</code>	0
<code>o.stay()</code>	1
<code>o.throw()</code>	2
<code>o.stay()</code>	4
<code>o.stay()</code>	6
<code>o.stay()</code>	9
<code>o.stay()</code>	1
<code>o.throw()</code>	2
<code>o.throw()</code>	5
<code>o.throw()</code>	<code>RuntimeError</code>
<code>o.stay()</code>	1
<code>o.stay()</code>	3
<code>o.throw()</code>	7

## Вариант №14

**Задача 3.1.** Реализовать разбор двоичного формата данных (в духе формата WAD игры Doom или графического формата PNG). Данные начинаются с сигнатуры 0x42 0x43 0x59 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	int8
	2	uint32
	3	Массив char, размер 3
	4	double
	5	uint64
	6	uint64
	7	Адрес (uint32) структуры В

Структура В:	1	Размер (uint16) и адрес (uint16) массива структур С
	2	uint64
	3	uint8
	4	Размер (uint16) и адрес (uint32) массива double
	5	uint16
	6	Адрес (uint16) структуры D
	7	int32

Структура С:	1	Размер (uint32) и адрес (uint16) массива uint16
	2	int32
	3	double

Структура D:	1	int16
	2	Массив uint32, размер 3
	3	int8
	4	int16

Примеры разбора двоичного формата с помощью функции f31:

### 1. Двоичные данные:

```
(b'BCYH.evi4kgh?\xeb\x02\xc3\xae\xcdL\xe4\x8d\xbf\xd8\x85A\xa7\x1f\xb3+\xc9\r#'  
b'\x1f\x07N\xba\x00\x00\x00\x93vX\xaa\x9a4\x1d\x9b0E!\x9c\xf2#\xb5\x00\x00'  
b'\x00\x05\x00(\xab\t[\x11\xbf\xb0S\x0bv\xc5\xcc@\x00\x00\x00\x02\x002\x80m'  
b'\x08\x97\xbf\xcc\xbe\xae\xaf\r\xdc`?\xde\x89\x0e\x0f)\x95\xfc?\xa7'  
b'\x01\x88\xf2v\x80\x00\xbf\xef\xb7oB\xb1\x16d?\xe4\x02\xc1\xb9\x04'  
b"\xd6\xcc?\xec\x8f%:\x83\xc2\xa8\xb7p'gN(\x1an\x9166n\x05\xef\x97\xb1\x8e\x00"  
b'\x02\x006\x97\xabuv\xf7\xc0\xed[\xbb\x00\x05\x00\x00\x00ZS\x02\x00\x82L\xc7'  
b'C\xc8')
```

Результат разбора:

```
{'A1': 46,
'A2': 1702258996,
'A3': 'kgh',
'A4': 0.8440874494648622,
'A5': 10214120546743295923,
'A6': 3155067458445332154,
'A7': {'B1': [{'C1': [30296, 43674, 13341, 39759, 17697],
'C2': -1425450223,
'C3': -0.06376716279719208},
{'C1': [40178, 9141],
'C2': -2140338025,
'C3': -0.22456916377606984}],
'B2': 10928958074567847259,
'B3': 187,
'B4': [0.4771151683874846,
0.04493358573677142,
-0.9911419203351213,
0.6253365148210945,
0.8924738066544551],
'B5': 21250,
'B6': {'D1': -18576,
'D2': [661081640, 443453750, 913180143],
'D3': -105,
'D4': -20082},
'B7': 1288127432}}
```

## 2. Двоичные данные:

```
(b'BCYH\x2f\xb1\xaa\x1doba\xbf\xe9#\x17\xfd\x2f\x81\xd4\xac\xe1\xa6X'
b'\xcb\x2f\x0j0>\xe4\xb3Xr\xa5\xa2\x00\x00\x00\x8d\xbfX\x95\x8dc0\x2f/'
b'\xa4\x05\xc0\x87\xb4@\xda\x94\xa4\x2fctg\x0f\x99\xdc\xa9\x00\x00\x00\x06'
b'\x00(~\x17\xc0\x9c\xbf\xe4X\xc49\xe30r\x00\x00\x00\x06\x04\x96[q0?\xe9\x2f1<'
b'we\x2f8\xb0\xbf\xe65Rb\x0b\xaa\xb4\xbf\xdazie\x0e\xe7X\xbf\xe4p\x9c\xb5\xa5`z'
b'\xc8`F\x14\x95\xd8=\xbf\t\xe7{I\x02\xb9\xfaYK\x00\x02\x00@^\xdd\x17'
b';\xcd\x8a\x06D^\x00\x03\x00\x00\x00d\xe5\xc0\x00|\xa9\xdd;\xc9')
```

## Результат разбора:

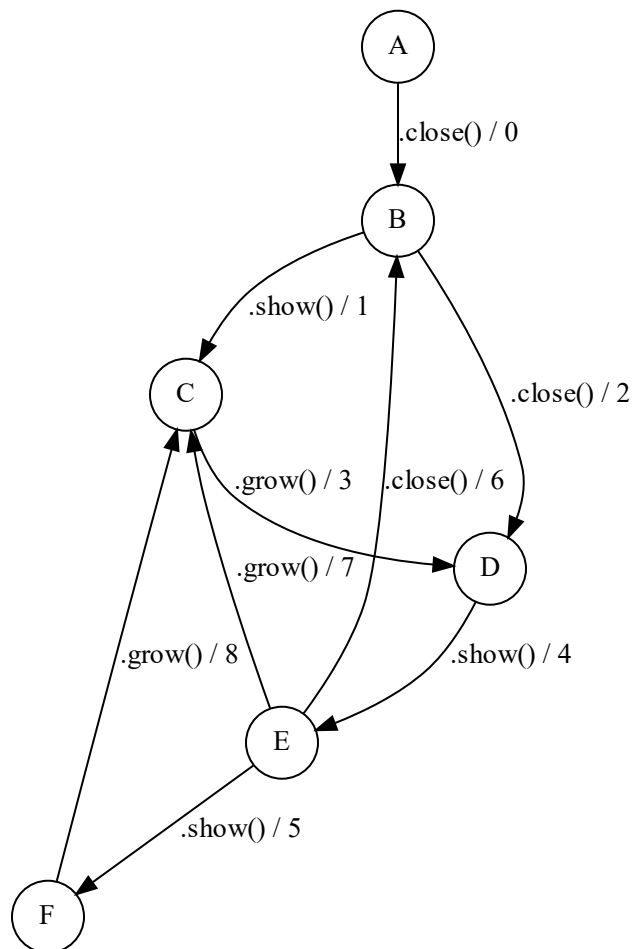
```
{'A1': -14,
'A2': 1722919453,
'A3': 'oba',
'A4': -0.7855339012349369,
'A5': 12457420944592990314,
'A6': 5710252836486555042,
'A7': {'B1': [{'C1': [48984, 38285, 25423, 61743, 41989, 49287],
'C2': 2115485852,
```

```

        'C3': -0.6358357554076817},
        {'C1': [46144, 55956, 42236, 29799, 3993, 56489],
         'C2': -1772392144,
         'C3': 0.8106977779387048}],
    'B2': 6835645355037165124,
    'B3': 94,
    'B4': [-0.6940090098335845, -0.413721417131105, -0.6387465999387267],
    'B5': 58816,
    'B6': {'D1': -14240,
           'D2': [1175754200, 1035930087, 2068382393],
           'D3': -6,
           'D4': 22859},
    'B7': -1445119031}}

```

**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.show()</code>	RuntimeError
<code>o.close()</code>	0
<code>o.close()</code>	2
<code>o.show()</code>	4
<code>o.close()</code>	6
<code>o.show()</code>	1
<code>o.grow()</code>	3
<code>o.show()</code>	4
<code>o.grow()</code>	7
<code>o.grow()</code>	3
<code>o.show()</code>	4
<code>o.show()</code>	5
<code>o.grow()</code>	8

## 2. Пример использования класса C32:

<code>o = C32()</code>	
<code>o.close()</code>	0
<code>o.close()</code>	2
<code>o.show()</code>	4
<code>o.show()</code>	5
<code>o.grow()</code>	8
<code>o.grow()</code>	3
<code>o.close()</code>	RuntimeError
<code>o.show()</code>	4
<code>o.grow()</code>	7
<code>o.close()</code>	RuntimeError
<code>o.grow()</code>	3
<code>o.show()</code>	4
<code>o.close()</code>	6
<code>o.show()</code>	1

## Вариант №15

**Задача 3.1.** Реализовать разбор двоичного формата данных (в духе формата WAD игры Doom или графического формата PNG). Данные начинаются с сигнатуры 0x22 0x44 0x5a 0x4c 0x4e, 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 (uint32) структуры B
	2	uint64
	3	uint16
	4	uint64
	5	uint64
	6	uint64
	7	double
	8	Структура C

Структура B:	1	Массив char, размер 4
	2	uint32

Структура C:	1	Массив адресов (uint32) структур D, размер 4
	2	int16
	3	Массив int32, размер 3

Структура D:	1	Размер (uint32) и адрес (uint16) массива int8
	2	uint16
	3	uint16
	4	Массив uint8, размер 8

Примеры разбора двоичного формата с помощью функции f31:

### 1. Двоичные данные:

```
(b'"DZLNQ\x00\x00\x00\xe3\x17\xc0\xf6\xe1K\x8e\xca \xf6q@~\x00\xd9|\x1at\x0e'
b'[\x87n\x84\xb4\x9b\xacU3\xa57\xa7\xb3\xe0\xf6\x84>Lr\x0eJ\xe7\xbf_'
b'\x00\x00\x00u\x00\x00\x00\x89\x00\x00\x00\x9e\x00\x00\x00(\x81\x8a\xa6s'
b'\xb9\xfb\xef1\xdcL?\xf5ferz1\x04e\re\xe2\xb9\xc2\xf7\x06\x00\x00\x00Y'
b'\x00=\xc4\x92QRWj\xc7\xba\x06[\x91\x0c\xd8Po\x04\x00\x00\x00q\x009\xa6]9|'
b'\x88K\x9a\x0b\x12^\x0ed\x02\x00\x00\x00\x87\x00\xb5\xdb\x86\x96%J\xdb\xaa\\'
b'`\xb2_\tP\x1f\x03\x00\x00\x00\x9b\x00\xf0\x86\\\xeb\xe0q_)?\x83\xe1\xa3')
```

Результат разбора:

```
{'A1': {'B1': 'ferz', 'B2': 224724076},
'A2': 14595686876233865187,
'A3': 63008,
'A4': 8366136529252139121,
```

```

'A5': 12437733276796803854,
'A6': 17789416058887942997,
'A7': -0.7277900917415114,
'A8': {'C1': [{'D1': [101, -30, 38, -71, -62, -9],
                  'D2': 50237,
                  'D3': 20882,
                  'D4': [82, 87, 106, 199, 186, 6, 91, 145]},
              {'D1': [12, -40, 80, 111],
                  'D2': 42553,
                  'D3': 14685,
                  'D4': [124, 136, 75, 154, 11, 18, 94, 46]},
              {'D1': [14, 100],
                  'D2': 56245,
                  'D3': 38534,
                  'D4': [37, 74, 219, 170, 92, 96, 178, 95]},
              {'D1': [9, 80, 31],
                  'D2': 34544,
                  'D3': 60252,
                  'D4': [224, 113, 95, 41, 63, 131, 225, 163]}]},
'C2': -32472,
'C3': [-1183603062, 1827627515, -180400932]}

```

## 2. Двоичные данные:

```

(b'"DZLNQ\x00\x00\x00\xd7\xf0f\xa2\xbc\xff\x06vN\xfd\xb1\x81\xea\x9fR'
b'\x84\xe8\x7f;Y\x08,]\xea\xc7r\x039\x11\xba\x05\x05\x83\x03D\x0f\xae\xdd'
b'\x98\xd9?]\x00\x00\x00v\x00\x00\x00\x8e\x00\x00\x00\xa6\x00\x00\x00\x9b'
b'$\x1e\xf9\x18&\x11\x1d\x14\x9b\x13\x89[\x1armzj\xb5\xdeI\x86\xc0\xf`'
b'H\x04\x00\x00\x00Y\x00\xceSk\xc0\xebMC\x93rN+\x88\x9a\xea\xf0\x96'
b'\xb5\xdc\x07\x00\x00\x00o\x00\x1e\x1a\xa4\xbf\x01\xc47\xac\xd4\xd2i\x92'
b'\xdc /\xc4\xa3\xc8\x06\x00\x00\x00\x88\x00\xd4a\x8f1\xba\xdf\xf4\r'
b'2\x0b\xa4\x08\x9c\xee\x6C\xcd\x06\x00\x00\x00\xa0\x00\xee>\xe3\x13'
b'\x96\x13\xb8\xd1L\x1c0\x1d')

```

## Результат разбора:

```

{'A1': {'B1': 'rmzj', 'B2': 2252988085},
'A2': 8504766131979350231,
'A3': 64846,
'A4': 9216762127818850737,
'A5': 8270836926580807995,
'A6': 253051526217545987,
'A7': 0.39995513919733505,
'A8': {'C1': [{'D1': [-64, 15, 96, 72],
                  'D2': 21454,
                  'D3': 49259,

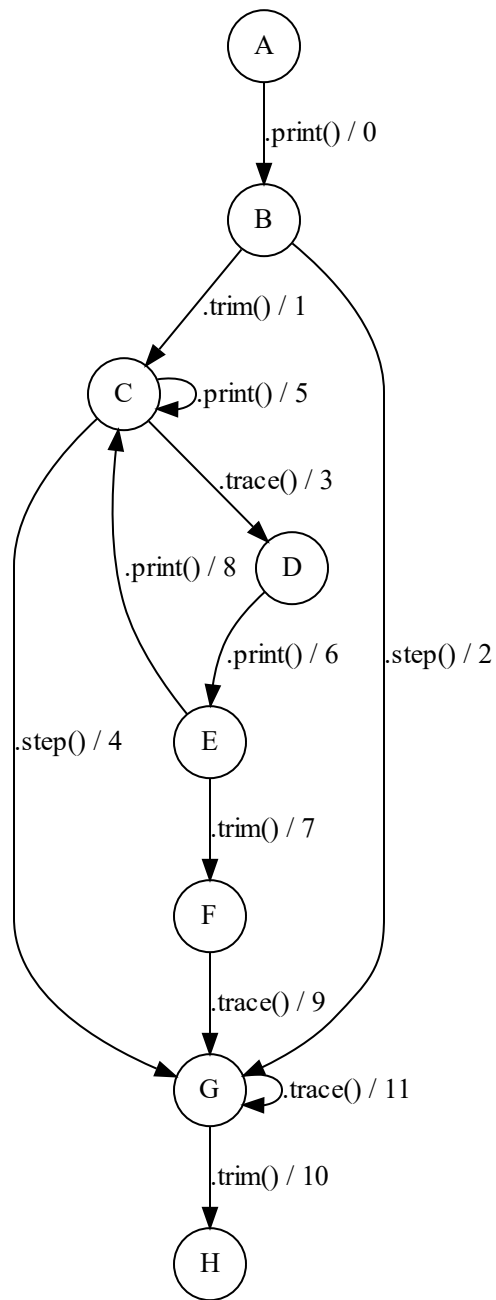
```

```

'D4': [235, 77, 67, 147, 114, 78, 43, 136]},
{'D1': [-102, -22, 121, -16, -106, -75, -36],
'D2': 6686,
'D3': 49060,
'D4': [1, 196, 55, 172, 212, 210, 105, 146]},
{'D1': [-36, 32, 47, -60, -93, -56],
'D2': 25044,
'D3': 27791,
'D4': [186, 223, 244, 13, 50, 11, 164, 8]},
{'D1': [-100, -18, 116, -74, 67, -51],
'D2': 16110,
'D3': 5091,
'D4': [150, 19, 184, 209, 76, 28, 48, 29]}],
'C2': 9371,
'C3': [639170846, -1693180655, 442206483]}}
```

**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.print()          0
o.trim()           1
o.trim()           RuntimeError
o.print()          5
o.trace()          3
o.print()          6
o.trace()          RuntimeError
o.print()          8
o.trace()          3
o.print()          6
o.trim()           7
o.trace()          9
o.trace()          11
o.trace()          11
o.trim()           10

```

## 2. Пример использования класса C32:

```

o = C32()
o.print()          0
o.trim()           1
o.print()          5
o.print()          5
o.trace()          3
o.print()          6
o.trace()          RuntimeError
o.print()          8
o.trace()          3
o.print()          6
o.trace()          RuntimeError
o.trim()           7
o.trace()          9
o.trace()          11
o.trim()           10

```

## Вариант №16

**Задача 3.1.** Реализовать разбор двоичного формата данных (в духе формата WAD игры Doom или графического формата PNG). Данные начинаются с сигнатуры 0x49 0x42 0x43 0xd3, 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 (uint32) массива char
	2	int32
	3	Структура B
	4	Массив структур F, размер 3

Структура B:	1	Адрес (uint32) структуры C
	2	Структура E
	3	int64

Структура C:	1	float
	2	uint16
	3	Структура D
	4	Массив uint8, размер 4
	5	int32
	6	int32

Структура D:	1	int8
	2	int16
	3	int16
	4	double
	5	uint16

Структура E:	1	int16
	2	uint16
	3	uint8
	4	int32

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

Примеры разбора двоичного формата с помощью функции f31:

1. Двоичные данные:

```
(b'IBC\xd3\x00\x03\x00\x00\x00A}e\x92\x19\x00\x00\x00D\x94z\x\xbd\xdefI\xdd\xccs'  
b'\x95\\\xbdb\xf9c\x00\x00\x00\x00\x03\x00\x00\x00e\x1d\xae\x00\x00\x00'  
b'\x04\x00\x00\x00qW&\x00\x00\x00\x04\x00\x00\x00\x813\xd5mtx>\xf8dm'  
b'\xe7\xb2\x9d\x8dz\x9fG?\xe0J\t\xdf\xe8\x8c\xe4Y\x03T\xcd\xc4ZvE\x1c'  
b'\xf7\x8e\xb1y\x98?\x06\xfb\xc3\xbe\x8e\xd1\xb9:\xff\x125>\xbc\xa1'
```

b'\xbf\xbe\xd9\xba\x89\xbfF\xecS\xbc\xe8\x161\xbd\xbd/\xeb\xfb\x1b'  
b'\x93\xbfV\xaf\x1e?=\x90\xda')

Результат разбора:

```
{'A1': 'mtx',  
'A2': 2103808537,  
'A3': {'B1': {'C1': 0.48514118790626526,  
              'C2': 59314,  
              'C3': {'D1': -99,  
                    'D2': -29318,  
                    'D3': -24761,  
                    'D4': 0.5113572476588186,  
                    'D5': 22787},  
              'C4': [84, 205, 196, 90],  
              'C5': 1984240887,  
              'C6': -1900971624},  
      'B2': {'E1': -27526, 'E2': 30909, 'E3': 222, 'E4': 1716116940},  
      'B3': 8328665054370685696},  
'A4': [{'F1': [0.5272790789604187, -0.278943806886673, 0.0019460382172837853],  
        'F2': 7598},  
        {'F1': [0.3684215247631073,  
                -0.4252512753009796,  
                -0.7770435214042664,  
                -0.02833089418709278],  
        'F2': 22310},  
        {'F1': [-0.09237655252218246,  
                -0.8832332491874695,  
                -0.8386095762252808,  
                0.740491509437561],  
        'F2': 13269}]]}
```

2. Двоичные данные:

(b'IBC\xd3\x00\x02\x00\x00\x00A\x1f:j\x9f\x00\x00\x00C\xf0q\xf8\xc0o\xeb'  
b'\x0b\xcd\xac\x98\x1e?";\x8b\xca\x94\x00\x00\x00\x02\x00\x00\x00d\xe3'  
b'\xb3\x00\x00\x00\x02\x00\x00\x001\xd2)\x00\x00\x00\x06\x00\x00\x00td'  
b'\x07vu\xbd\xd8L~\x9e\xa8G\x9b\x90\x92h?\xe2=\xac{\xf7a\xb6\xc9\xcfc\xf1\x81|'  
b'\xd6'\xcb\xb8\xf6\x1bX\x12\xbd\x13\xd3\x9c>\xfe\x8d\xb9\xbf\x19{g?\$\xc6\xcf'  
b'>U\xe2d\xbfq\x88\xeb\xbfq=-\xbe\xea\xc3\xa5>\xf1\x82]\xbd\x93l"'')

Результат разбора:

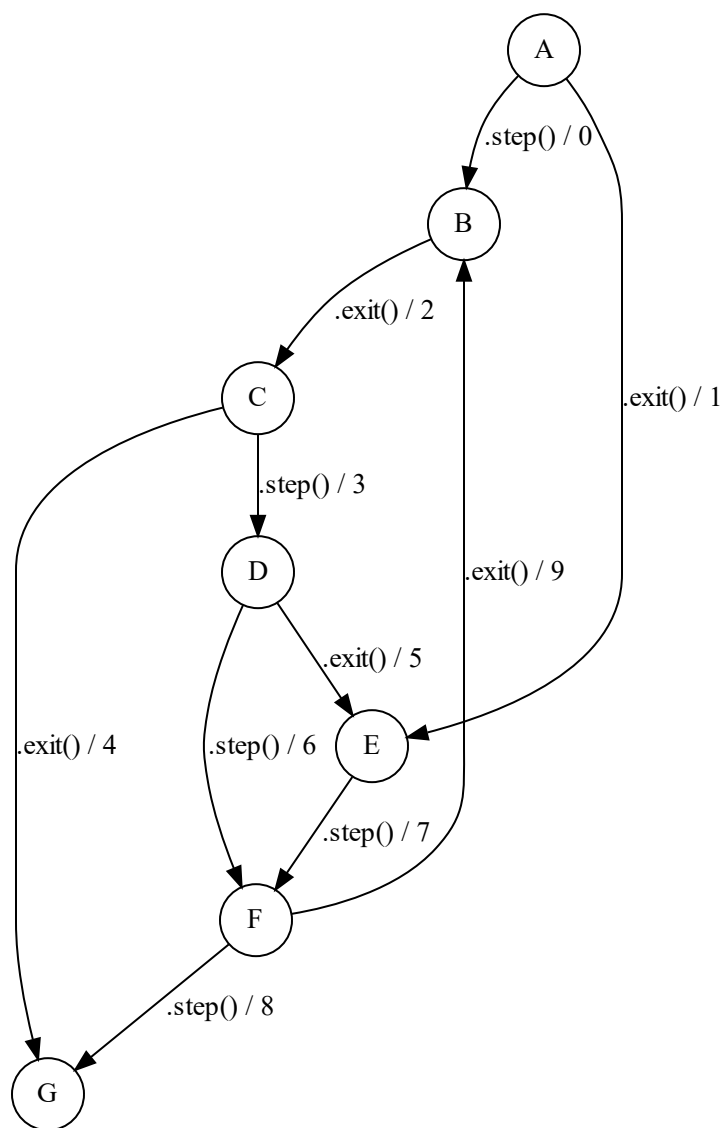
```
{'A1': 'vu',  
'A2': 523922079,  
'A3': {'B1': {'C1': -0.10561464726924896,
```

```

        'C2': 40616,
        'C3': {'D1': 71,
                'D2': -25712,
                'D3': -28056,
                'D4': 0.5700285359590029,
                'D5': 51663},
        'C4': [99, 241, 129, 124],
        'C5': -698299464,
        'C6': -165980142},
    'B2': {'E1': -3983, 'E2': 63680, 'E3': 111, 'E4': -351547988},
    'B3': -7485476114382730604},
    'A4': [{'F1': [-0.03609047830104828, 0.497175008058548], 'F2': 58291},
            {'F1': [-0.5995392203330994, 0.6436585783958435], 'F2': 53801},
            {'F1': [0.20887142419815063,
                    -0.9434954524040222,
                    -0.9423397183418274,
                    -0.4585238993167877,
                    0.47169771790504456,
                    -0.07198359072208405],
            'F2': 25607}]]}

```

**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.step()</code>	0
<code>o.exit()</code>	2
<code>o.step()</code>	3
<code>o.exit()</code>	5
<code>o.step()</code>	7
<code>o.exit()</code>	9
<code>o.exit()</code>	2
<code>o.step()</code>	3
<code>o.step()</code>	6
<code>o.exit()</code>	9
<code>o.step()</code>	RuntimeError
<code>o.exit()</code>	2
<code>o.exit()</code>	4

## 2. Пример использования класса C32:

<code>o = C32()</code>	
<code>o.step()</code>	0
<code>o.exit()</code>	2
<code>o.step()</code>	3
<code>o.step()</code>	6
<code>o.exit()</code>	9
<code>o.exit()</code>	2
<code>o.step()</code>	3
<code>o.exit()</code>	5
<code>o.step()</code>	7
<code>o.step()</code>	8

## Вариант №17

**Задача 3.1.** Реализовать разбор двоичного формата данных (в духе формата WAD игры Doom или графического формата PNG). Данные начинаются с сигнатуры 0x43 0x51 0x44 0x56, 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	uint8
	3	int16
	4	Адрес (uint32) структуры В
	5	Структура Е
	6	float
	7	Массив float, размер 6
	8	int8

Структура В:	1	uint16
	2	int64
	3	int8
	4	Адрес (uint32) структуры С
	5	uint32
	6	uint16

Структура С:	1	double
	2	Размер (uint16) и адрес (uint16) массива адресов (uint32) структур D

Структура D:	1	int8
	2	int32

Структура Е:	1	Массив char, размер 7
	2	Массив float, размер 4

Примеры разбора двоичного формата с помощью функции f31:

1. Двоичные данные:

```
(b' CQDV\xdd\xe3\xecz\xcdz^\xa3{\xdf\xb6e\x00\x00\x00lmjtiye2\xa8\xc8\xbe\xf5['  
b'\x87\xbeb\xa0\xb1\xbd\x9b\xa5\xde>\x8a\xb3\xe9\xbd\xd8\xf0M\xbfR\x12'  
b'\xb0\xbe\xd0\xcf\x94\xbe\xb3 D>5\x99\x06=\x15}.?UP5`n\x14\xf3Y\x07\x98'  
b'iG\x00\x00\x00L\x00\x00\x00n~\xfb6\xd84\xe0?\x02\x00Q\x00n\xa6\xeb'  
b'\xffI\xe0\xe2{\x19)\xd7Y\x00\x00\x00\xf5\x80\xfe1\x18d')
```

Результат разбора:

```
{'A1': 11771981498942284765,  
 'A2': 123,
```



```

'A3': -18721,
'A4': {'B1': 42606,
      'B2': 2961534444328583147,
      'B3': -41,
      'B4': {'C1': 0.5064507554888575,
            'C2': [{'D1': 80, 'D2': 342777909},
                  {'D1': -13, 'D2': 1771571033}]},
      'B5': 1828618485,
      'B6': 25624},
'A5': {'E1': 'lmjtiye',
      'E2': [-0.3919082283973694,
            -0.26437345147132874,
            -0.0867316871881485,
            0.43485721945762634]},
'A6': -0.11411197483539581,
'A7': [-0.8044562339782715,
      -0.3438897728919983,
      -0.2906479835510254,
      0.1915309876203537,
      0.032860953360795975,
      0.6815961003303528],
'A8': 85}

```

## 2. Двоичные данные:

```

(b' CQDV\x81\xc3\x1a\xa4\x80\x10\xe4\xd4\xdag\xf6e\x00\x00\x00qpwttrypc\x96'
 b'\xef\xbe\xf3q\n\xbf)\xba\xc7\xbe\xd9\xad-\xbf\xec\xf5o\xbdK \x0b?\xdd\xd7'
 b'\xd7\xbe\xa5Zi?\x1b\xff\x95=\xbd\x1a\>i\n/?]\x81\x10c\x0e\x8b\x08t\x0f\xed'
 b'VG\x00\x00\x00L\x00\x00\x00\xde!\x15\x86g\xfb\xef?\x02\x00Q\x00\x0b\xd2e'
 b'\xfaa?LrYG\xc1Y\x00\x00\x00\xca\x88o[\x97\xea' )

```

## Результат разбора:

```

{'A1': 15340404375425827713,
 'A2': 218,
 'A3': -2457,
 'A4': {'B1': 53771,
      'B2': 5141266121426598501,
      'B3': -63,
      'B4': {'C1': 0.9994390124130026,
            'C2': [{'D1': -127, 'D2': -1961991408},
                  {'D1': 8, 'D2': 1458376564}]},
      'B5': 1534036170,
      'B6': 60055},
 'A5': {'E1': 'qpwttryp',
      'E2': [-0.4679442346096039,

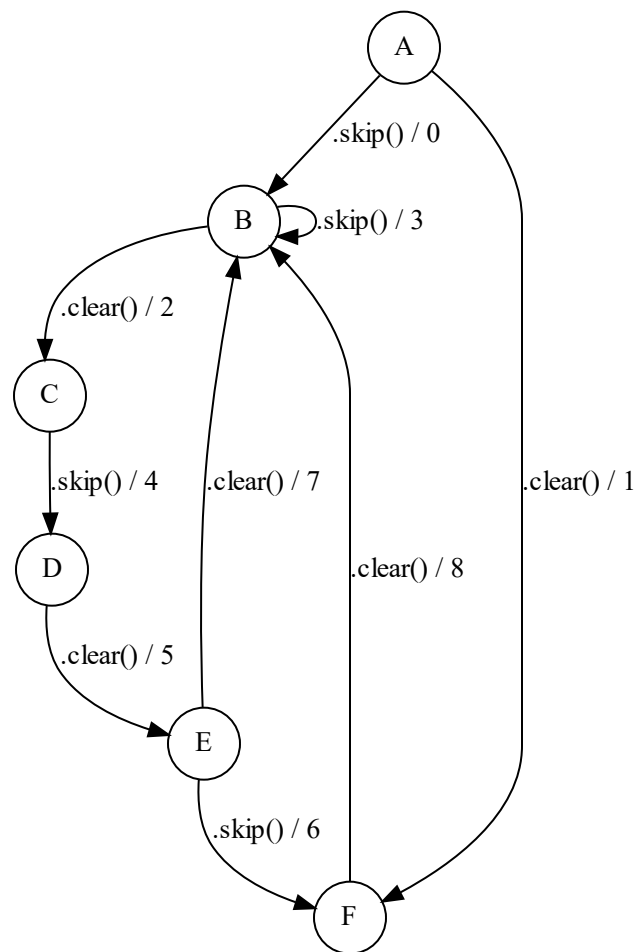
```

```

        -0.5408012270927429,
        -0.3900921642780304,
        -0.6784339547157288]},
'A6': -0.05858413875102997,
'A7': [0.5434615015983582,
        -0.42156878113746643,
        0.9115393757820129,
        0.0732404813170433,
        0.2149457484483719,
        0.6837525963783264],
'A8': 93}

```

**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.skip()	0
o.skip()	3
o.clear()	2
o.skip()	4
o.clear()	5
o.skip()	6
o.clear()	8
o.skip()	3
o.clear()	2
o.skip()	4
o.skip()	RuntimeError
o.clear()	5
o.clear()	7

## 2. Пример использования класса C32:

o = C32()	
o.skip()	0
o.skip()	3
o.skip()	3
o.clear()	2
o.skip()	4
o.clear()	5
o.skip()	6
o.clear()	8
o.clear()	2
o.skip()	4
o.clear()	5
o.clear()	7

## Вариант №18

**Задача 3.1.** Реализовать разбор двоичного формата данных (в духе формата WAD игры Doom или графического формата PNG). Данные начинаются с сигнатуры 0x50 0x4c 0x54 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	Размер (uint16) и адрес (uint16) массива char
	2	Адрес (uint32) структуры В
	3	Адрес (uint16) структуры С
	4	int16
	5	int8
	6	int16

Структура В:	1	int64
	2	int16

Структура С:	1	Размер (uint16) и адрес (uint32) массива адресов (uint16) структур D
	2	Массив uint32, размер 7
	3	double

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

Примеры разбора двоичного формата с помощью функции f31:

1. Двоичные данные:

```
(b'PLTR\x03\x00\x13\x00\x16\x00\x00\x00a\x00\xd1\xfc\xaaToza\xce.'
b'\xc3\xba\xbc\xcc\xfb\xfa\xd2\x8fay\xd8\xd7\xccK\xae\x16\xc4?P\x03\x00 \x00'
b'\x00\x00\xa4\xe3\xd1\x02|\xcc\xbb\xe0q\xa1Y\xfa\xfb\xac\xbfR\x07\x00'
b'2\x00\x00\x00\xe9r\x12\xbe\x1cp\xa2\x84,t\xe1\xbfV\x04\x00H\x00\x00\x00#'
b'\x009\x00L\x00\x03\x00[\x00\x00\x00\x99\x18\xb3\xe7\xc0\x0c\xf0\xcc\xae'
b"! )t\xaa2'\x17[# >po\x05\x83\xa5X\xc49\xe0\x04i\x0b=\x99\xcc?")
```

Результат разбора:

```
{'A1': 'oza',
'A2': {'B1': 7636922704096079566, 'B2': -11526},
'A3': {'C1': [{'D1': 0.15694216443233588, 'D2': 80, 'D3': [143, 97, 121]},
{'D1': -0.056045185033074096,
'D2': 82,
'D3': [164, 227, 209, 2, 124, 204, 177]},
{'D1': -0.5454313841360121,
'D2': 86,
'D3': [233, 114, 18, 190]}],
```

```

'C2': [3887274137,
        3438283968,
        1948852654,
        388444842,
        1042293595,
        2198171504,
        969169061],
'C3': 0.22342646653309028},
'A4': -11912,
'A5': -4,
'A6': 21674}

```

## 2. Двоичные данные:

```

(b'PLTR\x05\x00\x13\x00\x18\x00\x00\x00}\x000\x93\xb4\xfa\x12cqokf\xd1uL\xf5'
 b'\x85E=\xbf\x04\x1a\xeb\x8d\xb2\xdb\x8b\x89a\x14\xacc\x9f^\x11\xda'
 b'\xbf\x7f\x07\x00"\x00\x00\x00\xf3\x12k\xb0\x9eow\xfaC4\xeb\xbf\xa6\x04\x008'
 b'\x00\x00\x00\xf7\xccN1\xb6H\x08\xb8+c\x10\xdf\xbf\x8a\x05\x00K\x00\x00\x008'
 b'\x8f1\xbb\xe9\xe8v\xd8\x93\xa2\xd6o\xe6\xc6\xbf]\x07\x00_\x00\x00\x00)\x00<'
 b'\x00P\x00f\x00\x04\x00u\x00\x00\x00\x00\x1f\x9f\xc1\x01h\xefK\xd9\xe1B5e\x9a'
 b'\xb5%\xb8\xe4\x90\xed\xdcB\x073.\x13\x14\xc2h\x16\x93\xe1\x94z;\xe7\xbf')

```

## Результат разбора:

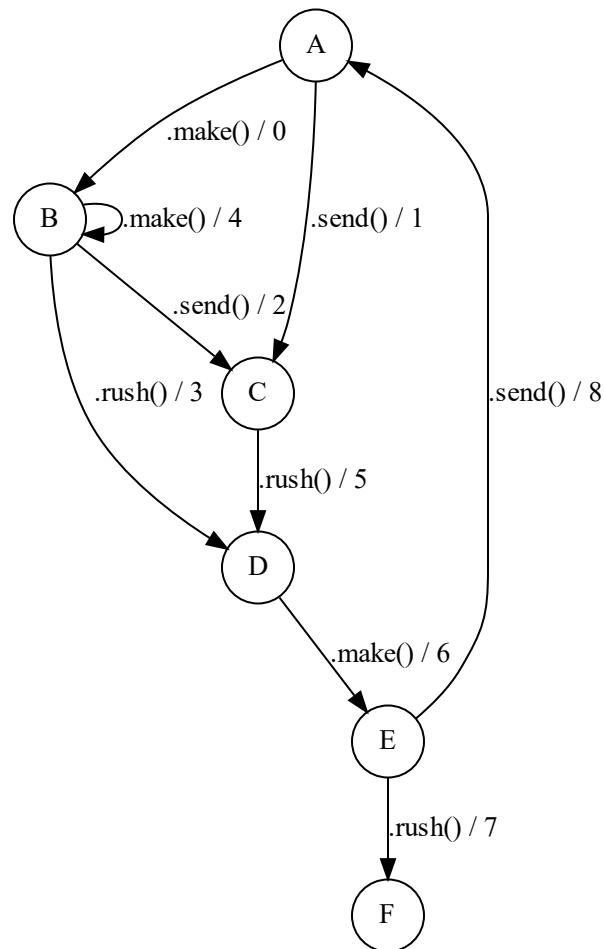
```

{'A1': 'cqokf',
 'A2': {'B1': -4666497197237570095, 'B2': 6660},
 'A3': {'C1': [{'D1': -0.4073101574455802,
                  'D2': 127,
                  'D3': [235, 141, 178, 219, 139, 137, 97]},
                {'D1': -0.8501300708698982,
                  'D2': 166,
                  'D3': [243, 18, 107, 176]},
                {'D1': -0.48537520665570755,
                  'D2': 138,
                  'D3': [247, 204, 78, 49, 182]},
                {'D1': -0.1789073751471324,
                  'D2': 93,
                  'D3': [56, 143, 49, 187, 233, 232, 118]}]},
 'C2': [29466399,
        3645632360,
        1697989345,
        2334504346,
        3706556644,
        775096130,
        1757549587],
 'C3': -0.7260105998835538},

```

```
'A4': -27856,  
'A5': -76,  
'A6': 4858}
```

**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.make()</code>	0
<code>o.make()</code>	4
<code>o.make()</code>	4
<code>o.rush()</code>	3
<code>o.send()</code>	RuntimeError
<code>o.make()</code>	6
<code>o.send()</code>	8
<code>o.send()</code>	1
<code>o.rush()</code>	5
<code>o.make()</code>	6
<code>o.make()</code>	RuntimeError
<code>o.rush()</code>	7

## 2. Пример использования класса C32:

<code>o = C32()</code>	
<code>o.send()</code>	1
<code>o.rush()</code>	5
<code>o.make()</code>	6
<code>o.send()</code>	8
<code>o.make()</code>	0
<code>o.make()</code>	4
<code>o.rush()</code>	3
<code>o.make()</code>	6
<code>o.make()</code>	RuntimeError
<code>o.rush()</code>	7

## Вариант №19

**Задача 3.1.** Реализовать разбор двоичного формата данных (в духе формата WAD игры Doom или графического формата PNG). Данные начинаются с сигнатуры 0xb9 0x4b 0x48 0x4b 0x54, 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 (uint32) и адрес (uint32) массива int8
	2	Размер (uint16) и адрес (uint16) массива структур B
	3	Адрес (uint32) структуры C
	4	int32
	5	int32
	6	uint8
	7	uint32
	8	Адрес (uint16) структуры D

Структура B:	1	int32
	2	int32

Структура C:	1	int32
	2	int32

Структура D:	1	uint16
	2	uint64
	3	Массив uint32, размер 3
	4	int32
	5	Массив float, размер 5
	6	uint64

Примеры разбора двоичного формата с помощью функции f31:

1. Двоичные данные:

```
(b'\xb9КНКТ\x00\x00\x00\x06\x00\x00\x00$\x00\x02\x00*\x00\x00\x00:{iX\x95\x02nD'
b'\xbdz1b|h\x00B\xe0\xbb\xbaK\x94\xc4\xf7\xaah\xc10(\xab\xd1\xb5\xf9\x1dT\xa2v'
b'Se4\x06J\x0e\xce\xdf\x9f~\xe2\xe0M\x05\x0bs\x0f&q\xe4Az\x987\xc7\xf9\xa4K'
b'o\x9e\xe8\x97/\xa8\x18\xec>\xe4\x96s?R\x84G\xbfI\xe3\xfb?\x02\xd1\x17?fV\x1f'
b'\xa4(\x94\xec\xc1\xd5\x01\xc6')
```

Результат разбора:

```
{'A1': [-32, -69, -70, 107, -108, -60],
'A2': [{'B1': -139827007, 'B2': 807971793},
{'B1': -1241965228, 'B2': -1569303707}],
'A3': {'C1': 872827406, 'C2': -824205442},
'A4': 2070501525,
```



```
'A5': 40780989,
'A6': 122,
'A7': 1818393704,
'A8': {'D1': 58080,
      'D2': 5549854704607326692,
      'D3': [1098553399, 3355026507, 1872685207],
      'D4': 799545580,
      'D5': [0.4464603364467621,
              0.822330892086029,
              -0.7886349558830261,
              0.5110029578208923,
              0.899751603603363],
      'D6': 11828868165873762758}}
```

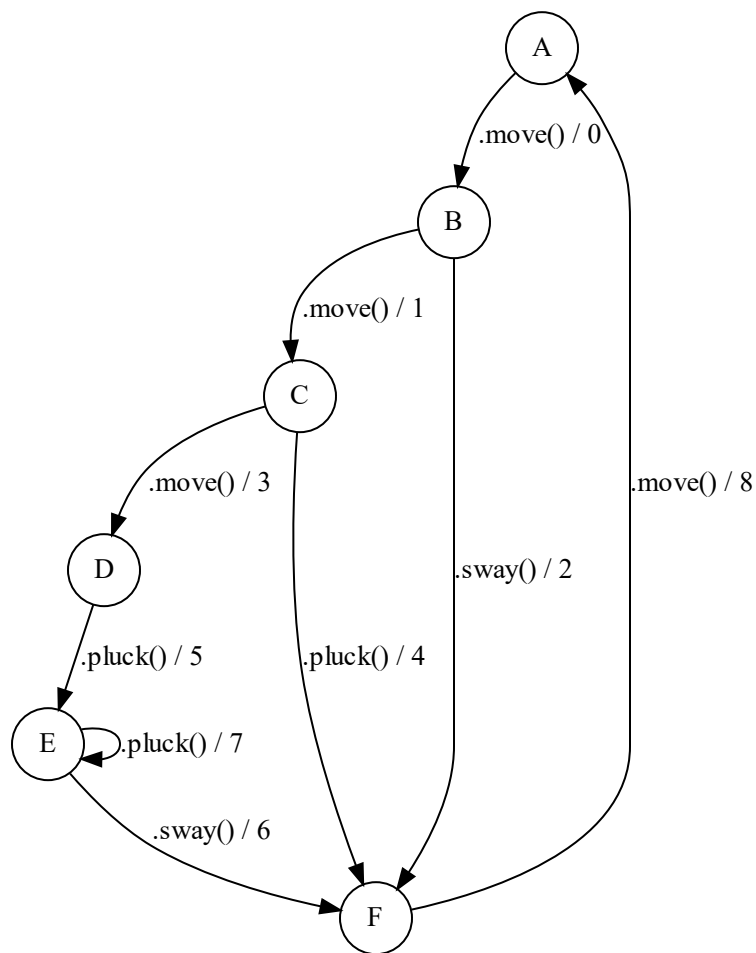
## 2. Двоичные данные:

```
(b'\xb9КНКТ\x00\x00\x00\x05\x00\x00\x00$\x00\x02\x00)\x00\x00\x009\x85\xae\xa7'
b'\x8c\xf8\xda:\x90A\x96\xaf\xc4\x95\x00A\xd3\n\xf4\x0f\x85!\xef\x84'
b'\xb9\xfce\x08\x92B\x12\x1b?\x1ey\xeb\x87:0\xe0\xb9U\xa5\xad\xe3\x80\x955'
b"\xa5\xc0\xfe#9\x1f'vw{\xa3\x1fT\xaa\xc0\x03\xb6<\xfcH\x1ex\x03\xbfq\xae1\xbf"
b'\x1e\xe1>\xcfc\xfe\xdb<\xa0o\xc7?Y\xb1idW\xaeF{\xcbg\xa2')
```

## Результат разбора:

```
{'A1': [-45, 10, -12, 15, -123],
'A2': [{'B1': 569345209, 'B2': -60487534},
        {'B1': 1108482879, 'B2': 511306631}],
'A3': {'C1': 978313401, 'C2': 1436921315},
'A4': -2052151412,
'A5': -119915888,
'A6': 65,
'A7': 2528101525,
'A8': {'D1': 32917,
      'D2': 3865708052912611111,
      'D3': [1987541923, 525642432, 62274812],
      'D4': 1209956355,
      'D5': [-0.9440677165985107,
              -0.6206264495849609,
              0.4062412679195404,
              0.019584549590945244,
              0.8503633141517639],
      'D6': 7230439344514492322}}
```

**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.sway()</code>	<code>RuntimeError</code>
<code>o.move()</code>	0
<code>o.sway()</code>	2
<code>o.move()</code>	8
<code>o.move()</code>	0
<code>o.move()</code>	1
<code>o.move()</code>	3
<code>o.pluck()</code>	5
<code>o.pluck()</code>	7
<code>o.move()</code>	<code>RuntimeError</code>
<code>o.sway()</code>	6
<code>o.move()</code>	8
<code>o.move()</code>	0

2. Пример использования класса C32:

<code>o = C32()</code>	
<code>o.move()</code>	0
<code>o.move()</code>	1
<code>o.pluck()</code>	4
<code>o.move()</code>	8
<code>o.move()</code>	0
<code>o.pluck()</code>	RuntimeError
<code>o.move()</code>	1
<code>o.move()</code>	3
<code>o.pluck()</code>	5
<code>o.pluck()</code>	7
<code>o.pluck()</code>	7
<code>o.sway()</code>	6

## Вариант №20

**Задача 3.1.** Реализовать разбор двоичного формата данных (в духе формата WAD игры Doom или графического формата PNG). Данные начинаются с сигнатуры 0x48 0x4e 0x45 0x47, 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	int64
	3	Массив int64, размер 2

Структура В:	1	float
	2	double
	3	Адрес (uint32) структуры С
	4	uint32
	5	int16

Структура С:	1	double
	2	Массив структур D, размер 6
	3	int8
	4	Адрес (uint32) структуры Е
	5	Массив int32, размер 4
	6	int8

Структура D:	1	int8
	2	uint16

Структура Е:	1	double
	2	Размер (uint16) и адрес (uint32) массива uint16
	3	int64
	4	uint16

Примеры разбора двоичного формата с помощью функции f31:

### 1. Двоичные данные:

```
(b'HNEGn\x00\x00\x00\x13\xd9\x9f!;\xb5\x1fH\xc6\x8f\xacF\x06\x8b`\x82'
b'\x080\x11\x8a\xd8\xc2&\xf6W{\x11C\x9e`\xd8W,N\x1c\xb5\xbf\x03\x00'
b' \x00\x00\x00a\xda|3\xc0[\xcaj\xeanx\xc0\xc1\xb8MN\xc6\xbf8\xa8'
b'\xa9\xcd\x92\xb0\xe1[\x0e\xb0F\xb1\r[\xf0\x01@\x12\xbd&\x00\x00\x00\xdd\xd11'
b'q#Z\xf5\x0b}\xc7}\x088\xe0\xf2\x1a<\x88\xb1\xde=\x00\xa0\xcf\xe2\xeb\x15'
b'\xa2\xbf>\x00\x00\x00Kr\x1a\xb9\xc46')
```

Результат разбора:

```
{'A1': {'B1': 0.108737051486969,
```

```

'B2': -0.03532349723519701,
'B3': {'C1': -0.1742646362431548,
      'C2': [{'D1': 56, 'D2': 43432},
              {'D1': -51, 'D2': 45202},
              {'D1': -31, 'D2': 3675},
              {'D1': -80, 'D2': 45382},
              {'D1': 13, 'D2': 61531},
              {'D1': 1, 'D2': 4672}],
      'C3': -67,
      'C4': {'E1': -0.08246315558048556,
              'E2': [22518, 4475, 40515],
              'E3': 7695063794371582561,
              'E4': 28394},
      'C5': [1902957021, 200628771, 142460797, 452124728],
      'C6': 60},
'B4': 3105518155,
'B5': 14020},
'A2': 5197071760580663571,
'A3': [-9052082391942918202, 2756417207235850248]}

```

## 2. Двоичные данные:

```

(b'HNEGt\x00\x00\x00h\x9b \xf1\x19U\xe023~\xec\x84\xd3\x1d\xcc\xae\tW\xa5\x03'
b'\x19xYyJ^\xbd.w29g\x10\xdc\xea\xb9 F+\x80%i\xa4?\x06\x00 \x00'
b'\x00\x00\xa1\xd4\x9b\x99\x07}\xb2\x06=:\x10\xe7\xcf\x87\xf2%\xc4?'
b'g\xcd\xd3\xd8\xdd\x9e\xd5\x8era*ib\xfa\x94\x0e\x9fA\x9d,\x00\x00\x00\xc0'
b']\xaa\xb6\x8be\x9d\xd0\x9e?\x80\x15"\xc0\x1f\xb4\x7f]b?p\xca\x10\xc0'
b'\xa8.\xe5?D\x00\x00\x00\x81\xd2\xf9_\xec\xe7')

```

## Результат разбора:

```

{'A1': {'B1': 0.884239137172699,
        'B2': 0.6619457007676157,
        'B3': {'C1': 0.15740806228114268,
                'C2': [{'D1': 103, 'D2': 54221},
                        {'D1': -40, 'D2': 40669},
                        {'D1': -43, 'D2': 29326},
                        {'D1': 97, 'D2': 26922},
                        {'D1': 98, 'D2': 38138},
                        {'D1': 14, 'D2': 16799}],
                'C3': -99,
                'C4': {'E1': 0.03986470403254949,
                        'E2': [24138, 11965, 12919, 26425, 56336, 47594],
                        'E3': 482585581677434017,
                        'E4': 14909},
                'C5': [-1230348864, -794991221, -2141438050, 532685333],

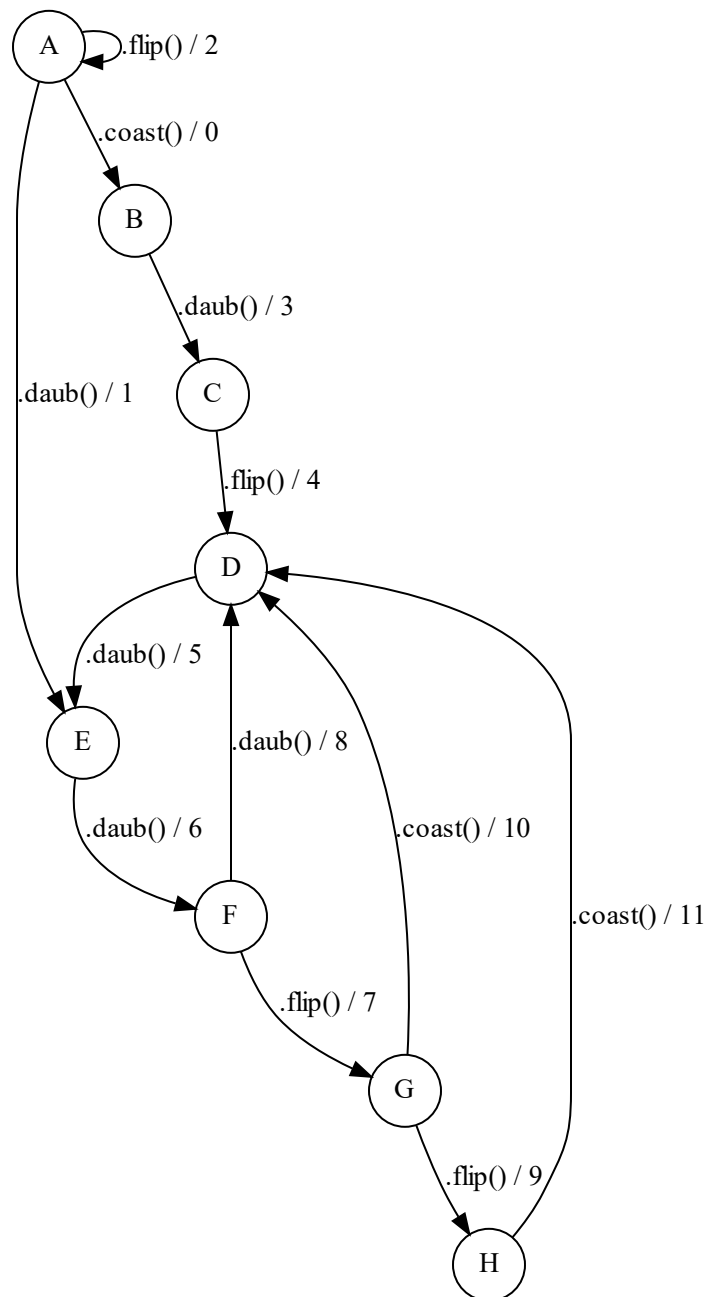
```

```

        'C6': -76},
        'B4': 1610207873,
        'B5': -6164},
        'A2': 3666023666587573096,
        'A3': [-5851269021555720653, 8744152200347211529]}

```

**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.flip()	2
o.coast()	0
o.daub()	3
o.flip()	4
o.flip()	RuntimeError
o.daub()	5
o.daub()	6
o.flip()	7
o.coast()	10
o.daub()	5
o.coast()	RuntimeError
o.daub()	6
o.flip()	7
o.flip()	9
o.coast()	11
o.daub()	5
o.daub()	6
o.daub()	8

## 2. Пример использования класса C32:

o = C32()	
o.flip()	2
o.coast()	0
o.daub()	3
o.flip()	4
o.flip()	RuntimeError
o.daub()	5
o.daub()	6
o.flip()	7
o.coast()	10
o.daub()	5
o.daub()	6
o.daub()	8
o.daub()	5
o.daub()	6
o.coast()	RuntimeError
o.flip()	7
o.flip()	9
o.coast()	11

## Вариант №21

**Задача 3.1.** Реализовать разбор двоичного формата данных (в духе формата WAD игры Doom или графического формата PNG). Данные начинаются с сигнатуры 0x53 0x5a 0x53 0x1d, 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	uint32
	2	float
	3	Структура В
	4	float
	5	uint32
	6	uint16
	7	int16
	8	Размер (uint16) и адрес (uint32) массива uint32

Структура В:	1	uint32
	2	int8
	3	Массив адресов (uint16) структур С, размер 2

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

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

Примеры разбора двоичного формата с помощью функции f31:

1. Двоичные данные:

```
(b'SZS\x1d\x1e\xbf\xcd\xbfD\x89\xc2\xdcT%qX\x00a\x00\xaa>\x846\xc7\xa7\xe4w'
b'3G\xe8(\xae\x00\x04\x00\x00\x00\xbb\xf4v\xe0\xd4G\x84\xa8\xd7\x86'
b'\xf5\xfb\xb5(*\x9d~\x13\x06Y\xe4\x1e\x1d\x0f\nN\x88\xeb\xdf\xbfP\xaf\x1b\x00'
b"\x00\x00\x02\x00'\x07k\x844\x00\x06\x007\xb1.\xa1\x85x\x06\x82&\x15\x80d"
b",\x00\x00\x00C\x99\xdf\xd0\x9e\xe3\xf4\xfc\xda\x00\x03\x00U\xf7(\xa6\xe6q'"
b'I\xff+v\x88}=\x12*2/n>p\x020\x05tri>\x15!\x1a\x00\x00\x00\x02\x00r4\xd6'
b'/\x7f\x00\x05\x00\x82\xc8kY\xd4\x8fX\xfd\x8b"\xe1\xe2\xe5\x00\x00'
b'\x00\x8c\x8di\xfa\xc9\x8aJ\xe9w\x0b\x00\x03\x00\x9e\x1c\x84\x11@\xf3'
b'\xf1\xc7\x11\xae\x1f\xfc\x14W\x8a>\x0f')
```



Результат разбора:

```
{'A1': 515886385,
'A2': -0.7677270174026489,
'A3': {'B1': 3696502129,
      'B2': 88,
      'B3': [{'C1': {'D1': -0.8151718974113464,
                    'D2': [17615514194843838679, 9724955725857987966],
                    'D3': 124486708,
                    'D4': [4870, 23012, 7709, 3850, 20104, 60383]}},
            'C2': 39391,
            'C3': 208,
            'C4': -1628818444,
            'C5': 252,
            'C6': -38,
            'C7': [-1322344059, 2013692454, 360735788]}},
{'C1': {'D1': 0.14563408493995667,
      'D2': [17809668235107715583, 3131840662359648818],
      'D3': 886452095,
      'D4': [12142, 15984, 591, 1396, 29289]}},
      'C2': 36201,
      'C3': 250,
      'C4': -913683735,
      'C5': 119,
      'C6': 11,
      'C7': [-932488748, -1889993333, 585229029]}]},
'A4': 0.25823041796684265,
'A5': 2816767795,
'A6': 18408,
'A7': 10414,
'A8': [478417216, 4092708625, 2921331732, 1468677647]}
```

## 2. Двоичные данные:

```
(b'SZS\x1d\x14F3&\xbfg\xfb>\xcdfv\r\xd9\x00c\x00\xa6?Vva\xefz\x8d'
b'\xcd\x1e\xe1\xfeh\x00\x03\x00\x00\x00\xb7P\xb0Xhpxu\x0f\xda\x1cK0\xca'
b'\x98\xa6\x1a\x87\xc2\xb8t\xf6\xd7j:dd\xfa\xd9\x91\\?L\x9f\x05\x00\x00\x00'
b'\x02\x00'\xad4\xd5q\x00\x07\x007\xf3\xbc\x90\xcf"\xbe\xb5Y\x96\xb6]\x99\x00'
b'\x00\x00E:\x80\x18\xae\xbd\xa4\n\xfc\x90\x00\x03\x00Wh-q\xb8\x06\x90\x89\xde'
b'l\xa1\xd6,\xd9\xc3\xe4\x17a\x1e\x1a\xc4\xbf\x05\x07\x8e\x00\x00\x00\x02'
b'\x00t\x80\rK\xea\x00\x02\x00\x84\x9c*N\xb8\x13\xc7\xea\xf2N\xc8\xb1)\x00\x00'
b'\x00\x88q\xc0\x80\xbd/0\xe3\xdc\x01\x00\x03\x00\x9a@\xd3G\xf4\xc7f\xa0\xf2'
b'I=\xfd')
```

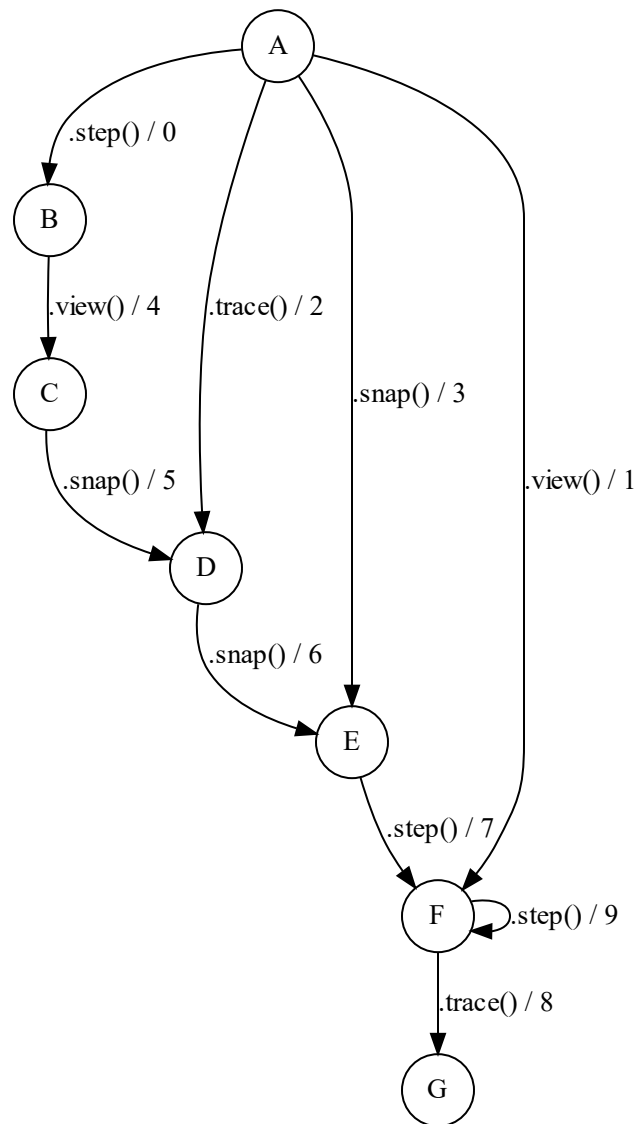
Результат разбора:

```

{'A1': 340144934,
 'A2': -0.9060095548629761,
 'A3': {'B1': 3446044274,
        'B2': -39,
        'B3': [{'C1': {'D1': 0.7993014454841614,
                        'D2': [5814244324522095887, 15716519605689689626],
                        'D3': 2905920881,
                        'D4': [34754, 47220, 63191, 27194, 25700, 64217, 37212]},
                  'C2': 14976,
                  'C3': 24,
                  'C4': -1363303414,
                  'C5': 252,
                  'C6': -112,
                  'C7': [-205745969, 582923609, -1766433383]},
                {'C1': {'D1': -0.5196465253829956,
                        'D2': [7506781189094541790, 7827773115467031575],
                        'D3': 2148355050,
                        'D4': [24862, 6852]},
                  'C2': 29120,
                  'C3': 128,
                  'C4': -1120980765,
                  'C5': 220,
                  'C6': 1,
                  'C7': [-1674948936, 331868914, 1321775401]}]}},
 'A4': 0.8377438187599182,
 'A5': 4017786317,
 'A6': 7905,
 'A7': -408,
 'A8': [1087588340, 3345391695, 4064886269]}

```

**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.step()</code>	0
<code>o.view()</code>	4
<code>o.snap()</code>	5
<code>o.snap()</code>	6
<code>o.step()</code>	7
<code>o.step()</code>	9
<code>o.step()</code>	9
<code>o.trace()</code>	8

2. Пример использования класса C32:

<code>o = C32()</code>	
<code>o.step()</code>	0
<code>o.trace()</code>	RuntimeError
<code>o.view()</code>	4
<code>o.snap()</code>	5
<code>o.step()</code>	RuntimeError
<code>o.snap()</code>	6
<code>o.step()</code>	7
<code>o.step()</code>	9
<code>o.trace()</code>	8

## Вариант №22

**Задача 3.1.** Реализовать разбор двоичного формата данных (в духе формата WAD игры Doom или графического формата PNG). Данные начинаются с сигнатуры 0x51 0x43 0x55 0x53 0x62, 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	Адрес (uint32) структуры B
	3	int8

Структура B:	1	Размер (uint16) и адрес (uint16) массива структур C
	2	float
	3	uint32
	4	Структура D
	5	uint64
	6	double
	7	uint16
	8	Массив uint16, размер 7

Структура C:	1	uint16
	2	int8
	3	int8

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

Структура E:	1	Размер (uint16) и адрес (uint32) массива double
	2	Размер (uint16) и адрес (uint32) массива int32
	3	uint64

Примеры разбора двоичного формата с помощью функции f31:

1. Двоичные данные:

```
(b'QCU$B\xe9\x919\xbe\x00\x00\x00F\x87\xba\xca\xa8\xbf\xd9\xf0\x8b\xbf\xe0'
b'CsU\xbc;\x1a?\xecp\xd7%\xdeL\x1a!\xff\x99\xd9R\x82\xa7\x84r\x81'
b'\x99\xb2\x00\x02\x00\x00\x00\x16\x00\x03\x00\x00\x00&\xaa\xc8%NC\x99'
b'\x1f\xe6\x00\x02\x00\x0e\xbfw\x9f4\x9e\x98\xd0\xfa\x00\x00\x002?\xe9'
b'\xa0\xe6\xa2\x1f\x08\xd8\xa2\xbd=\xf9\x1b+\x9e\x7f\xbf\xe6^\xc2\xd7\x91'
b'\xfd\xec\xb8\x9e>\xae#\xfa7\x0c\x9b\x92\x1f\xd7\xe4\x1am| ')
```

Результат разбора:

```
{'A1': 3918608830,
'A2': {'B1': [{'C1': 34746, 'C2': -54, 'C3': -88},
```

```

        {'C1': 49113, 'C2': -16, 'C3': -117}],
'B2': -0.9672729969024658,
'B3': 2660815098,
'B4': {'D1': {'E1': [-0.5082337069050908, 0.8887744655243097],
                'E2': [570399193, 1384294276, 1921096114],
                'E3': 12306126999861665766},
        'D2': 0.8008912245207069},
'B5': 11726597144854634111,
'B6': -0.6990675173146292,
'B7': 47262,
'B8': [16046, 9210, 14092, 39826, 8151, 58394, 28028]},
'A3': 120}

```

## 2. Двоичные данные:

```

(b'QCU$B/\x8c\xad\x18\x00\x00\x00B\x91\xc9\xfe\xd73b\xe7\xd2\xe2?\xd1'
 b'f\xf2\xe6\x16\x18T\bfb\x80\x86\x86X\xd8\xba\x8b1\xf2\xa3\xdb'
 b"K'\x00\x02\x00\x00\x00\x16\x00\x02\x00\x00\x00&1\x94\xd2)\xbb\xb0"
 b'\xe7\xd5\x00\x02\x00\x0e?%i\xa1n\x90\xcc\xb8\x00\x00\x00.\xc6|o\x1d0'
 b'\xa2\xb8\x8c\t\b07\x88\xf3\x843?\xbbS\x051\xa2\x05\x10H\x1d\xdec\xf4\n'
 b'z?q\xd0\xf3\xdaf\rY1')

```

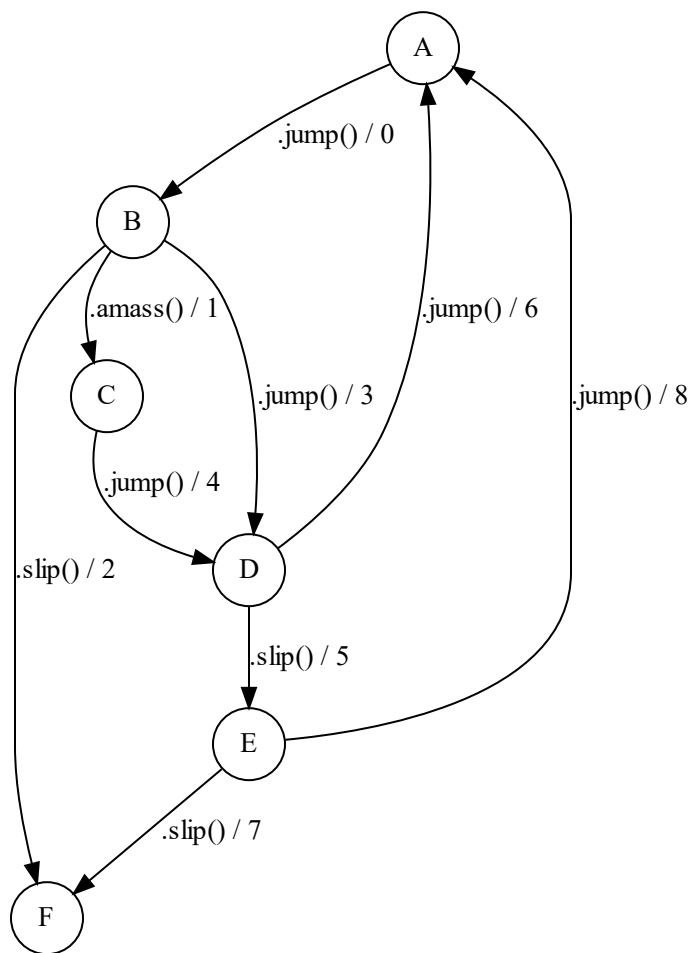
## Результат разбора:

```

{'A1': 797748504,
 'A2': {'B1': [{'C1': 51710, 'C2': -41, 'C3': 51},
                {'C1': 25319, 'C2': -46, 'C3': -30}],
        'B2': 0.6461430191993713,
        'B3': 1854983352,
        'B4': {'D1': {'E1': [0.271908497518216, -0.19334417885392585],
                'E2': [-1165282830, -1545909465],
                'E3': 3572711481096071125},
        'D2': 0.17567242556767382},
        'B5': 10090790192667657267,
        'B6': 0.10673553907146904,
        'B7': 18461,
        'B8': [56931, 62474, 31295, 29136, 62426, 26125, 22833]},
 'A3': -111}

```

**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.slip()</code>	<code>RuntimeError</code>
<code>o.jump()</code>	0
<code>o.amass()</code>	1
<code>o.jump()</code>	4
<code>o.jump()</code>	6
<code>o.jump()</code>	0
<code>o.jump()</code>	3
<code>o.slip()</code>	5
<code>o.jump()</code>	8
<code>o.jump()</code>	0
<code>o.slip()</code>	2

2. Пример использования класса C32:

o = C32()	
o.jump()	0
o.jump()	3
o.jump()	6
o.jump()	0
o.amass()	1
o.jump()	4
o.slip()	5
o.amass()	RuntimeError
o.jump()	8
o.jump()	0
o.slip()	2



## Вариант №23

**Задача 3.1.** Реализовать разбор двоичного формата данных (в духе формата WAD игры Doom или графического формата PNG). Данные начинаются с сигнатуры 0xff 0x56 0x56 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	Массив структур B, размер 5
	2	int64
	3	int16
	4	Структура C
	5	Адрес (uint32) структуры F

Структура B:	1	Массив char, размер 3
	2	int16

Структура C:	1	Размер (uint32) и адрес (uint16) массива float
	2	int32
	3	Структура D
	4	Структура E
	5	int32
	6	uint32

Структура D:	1	float
	2	int16
	3	uint32

Структура E:	1	uint8
	2	uint16
	3	uint64
	4	double
	5	uint32
	6	float
	7	float

Структура F:	1	double
	2	int32
	3	Массив uint8, размер 3
	4	float

Примеры разбора двоичного формата с помощью функции f31:

1. Двоичные данные:

```
(b'\xffVVGbge\xb81tz1\xa9\xb6pqd\xb4(gnx\xea\xeejlp\n\x96\xe4)\xd8\r\xe0h\xc2'  
b'\x9eLh\x00\x00\x00\x02\x00f\xf0\x87\x0c\xd7?h$\xb5*R|[\x01\xa0\x0c'
```

```

b'\xc0\xb2\xe6\xe9/<\xfe9@\xfe\xbf\xd8\xa0\xf1\xe8U'\xfc\x04\xc7\xdeJ>\x98'
b'\xeeG\xbe\x07\xd8(SI\xa5\x0f\xda\x8f\x85\xa8\x00\x00\x00n?\x00q\xc3\xbf\x16'
b'\xb7\x13?\xe9\xddf-sS\xd0\xf9\x7f\x14\xc6\xef\xe9!\xbe\xb7>\xb2')

```

Результат разбора:

```

{'A1': [{'B1': 'bge', 'B2': -18383},
        {'B1': 'tzl', 'B2': -22090},
        {'B1': 'pqd', 'B2': -19416},
        {'B1': 'gnx', 'B2': -5394},
        {'B1': 'jlp', 'B2': 2710}],
'A2': -2005834604905708898,
'A3': 19560,
'A4': {'C1': [0.5017358660697937, -0.5887309908866882],
        'C2': -259584809,
        'C3': {'D1': 0.9068101048469543, 'D2': 10834, 'D3': 2086338976},
        'C4': {'E1': 12,
                'E2': 49330,
                'E3': 16638882237306716414,
                'E4': -0.3848233002550072,
                'E5': 80207434,
                'E6': 0.298692911863327,
                'E7': -0.13266050815582275},
        'C5': 1397335311,
        'C6': 3666838922},
'A5': {'F1': 0.8082762611107857,
        'F2': -109112122,
        'F3': [239, 233, 33],
        'F4': -0.3579002022743225}}

```

## 2. Двоичные данные:

```

(b'\xffVVGym\x7fi jag\xfc\x8dzjv$\xb6fzu\xa2\xc2ovw\x18\xbf\x95\xb7\x18:?$'
b'\xd6,\xf9\x00\x00\x00\x02\x00f\xd3\xfc\xc7\xaf\xbc\xb4\xb8F\x8d\x00\x1d'
b'\xda;\xc9\xc0xbc%I\xb1D\x11\xa7d\x93\xf4\xbf\xe8\xa3\xe2\xf8\xa8'
b"\xf9\xfe\xe8KD\xbb\xbe\x8a'P>\xc8\xa5\x8d\xa5a\xcb\x9a\x7f\xe0"
b'\xab\xd0\x00\x00\x00n\xbe\x9a\xc1\xa6\xbfFa"? \xbb\x8ee\x87\\\xc3\x90\xa9\xb9'
b'\xe5\xf6\xd6k<?y\x10\x19')

```

Результат разбора:

```

{'A1': [{'B1': 'gym', 'B2': 32617},
        {'B1': 'jag', 'B2': -883},
        {'B1': 'zjv', 'B2': 9398},
        {'B1': 'fzu', 'B2': -23870},
        {'B1': 'ovw', 'B2': 6335}],

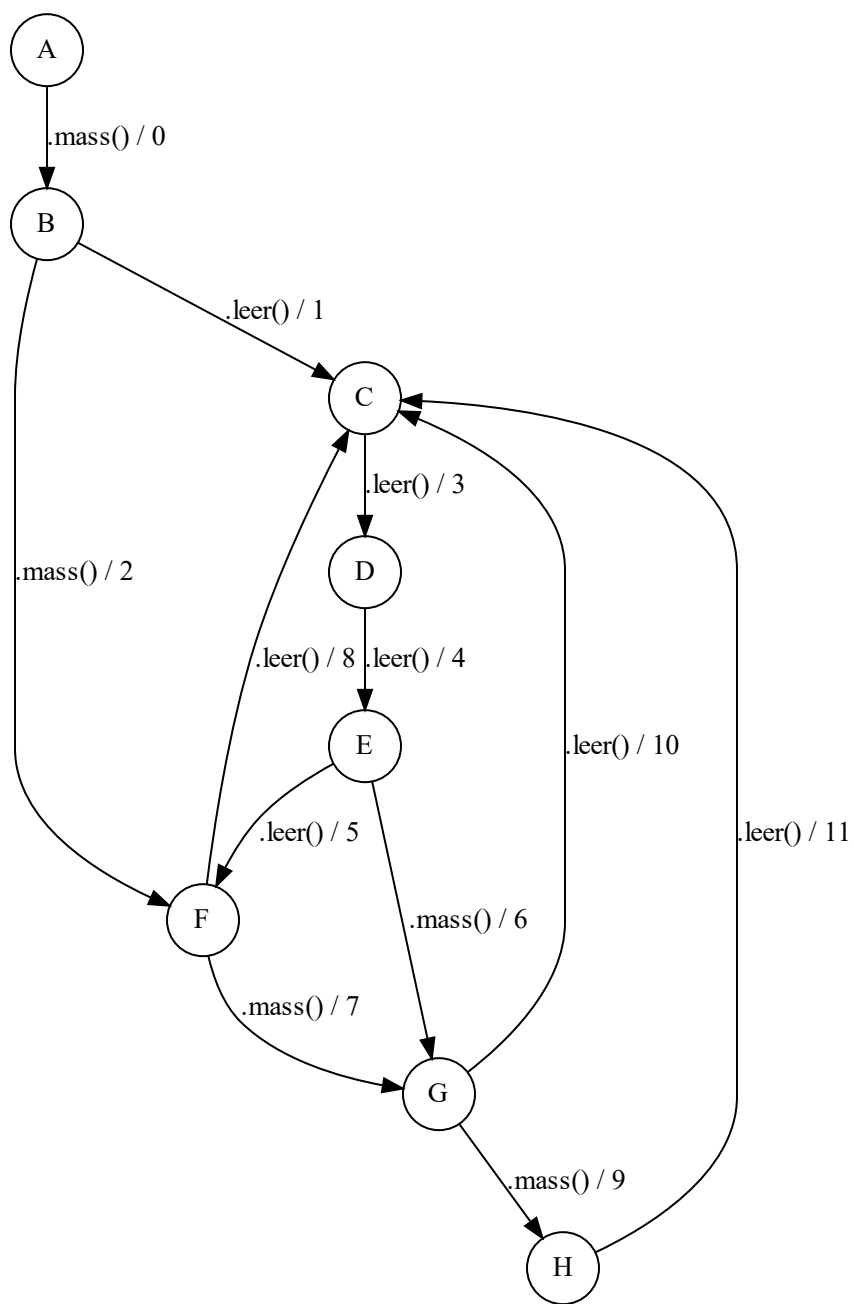
```

```

'A2': -7658626002872163114,
'A3': 11513,
'A4': {'C1': [-0.3022586703300476, -0.7749196290969849],
      'C2': -738408529,
      'C3': {'D1': -0.02206052467226982, 'D2': -29440, 'D3': 500841417},
      'C4': {'E1': 192,
            'E2': 48165,
            'E3': 5310100278260044788,
            'E4': -0.7700056890824383,
            'E5': 3897246907,
            'E6': -0.26983118057250977,
            'E7': 0.3918880522251129},
      'C5': -1520317542,
      'C6': 2145430480},
'A5': {'F1': 0.1076415496325025,
      'F2': -1447434762,
      'F3': [214, 107, 60],
      'F4': 0.9729018807411194}}

```

**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.mass()          0
o.leer()          1
o.leer()          3
o.leer()          4
o.mass()          6
o.leer()          10
o.leer()          3
o.leer()          4
o.leer()          5
o.leer()          8
o.leer()          3
o.leer()          4
o.leer()          5
o.mass()          7
o.mass()          9
o.leer()          11

```

## 2. Пример использования класса C32:

```

o = C32()
o.mass()          0
o.mass()          2
o.mass()          7
o.mass()          9
o.leer()          11
o.leer()          3
o.leer()          4
o.mass()          6
o.leer()          10
o.leer()          3
o.leer()          4
o.leer()          5
o.leer()          8
o.mass()          RuntimeError
o.leer()          3
o.leer()          4

```



```
{'A1': 'iztp',
'A2': 217,
'A3': {'B1': [{'C1': 97, 'C2': -78, 'C3': 0.7251012325286865},
               {'C1': 192, 'C2': -96, 'C3': -0.5463756918907166},
               {'C1': 5, 'C2': -26, 'C3': -0.6979178190231323},
               {'C1': 8, 'C2': 57, 'C3': -0.6086090207099915}],
'B2': 174,
'B3': [0.09780109344533816,
       0.22746646520356784,
       0.9710072032369277,
       -0.8022167559305005,
       -0.9671589726468341,
       -0.3787631804240652],
'B4': 4883366810375890704},
'A4': {'D1': 580275905, 'D2': 17616836063485292460, 'D3': 0.6605730652809143},
'A5': 65437,
'A6': 3523280495563374546,
'A7': [9139, 37374, 26784, 17500, 10873, 40859],
'A8': {'E1': -0.6123086214065552,
'E2': [-7, -127, -96, -38, 90, -41, 21, -107]}}
```

## 2. Двоичные данные:

```
(b'YTPS\x06\x00\x00\x009b\x87\x00Y\xe8q5\x11\xb5\xbbI\x07'
b'\xc3Bq\xea\xeaW>\xcd\xeb\xcb\\xa5\x0b\xbd8;\xf3-\x1bT\xcdhd\xc9R\xd3C\xf8'
b'\xaf\x9e\x00\x00\x00rtgdmh\xbc\t\x94\xc4(\xbe\xcd\xa1M\x89\xc6=\x9a'
b'\xd4\x08\x91\x7f\xbd\x1c/\xd0U?J\xa07\xb1\xd9F\xea?\\~)\xf6\x85'
b'\x8c\xdd\xbf\xfc\xd3`3A\xda?\xd4sZ\xd9\x8a\xfe\xdf?\xa4s\x02\x01u'
b'\xea\xd1\xbf\xba4/\xba}\[\xe7??\x00E\x00K\x00Q\x00\x1f\x06\x00\x00\x00'
b'W\x00.\xd7\xe0Ex\b4\xd7\x9cH\xf5\xe9>\xad\xc4\xe6$\xd7\xfd(')
```

## Результат разбора:

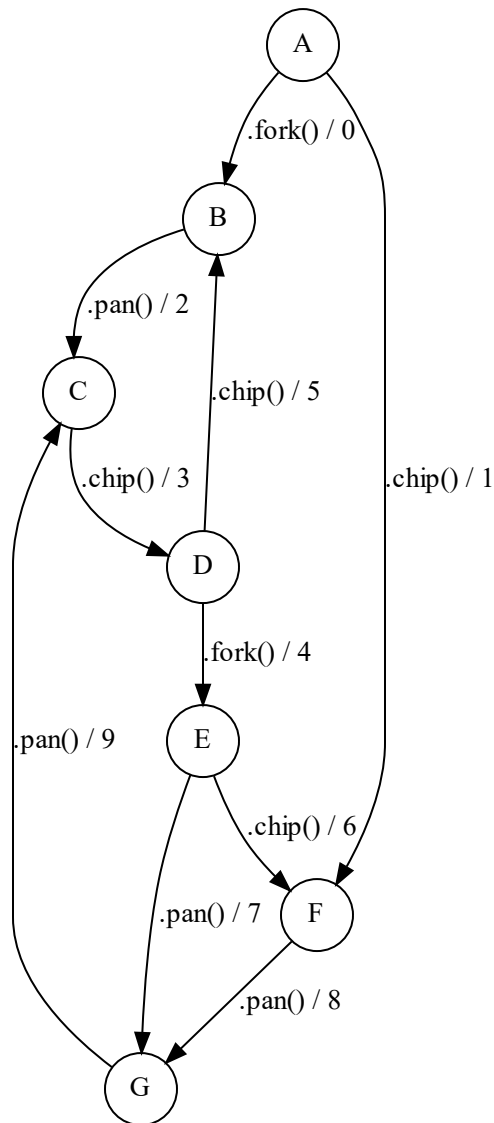
```
{'A1': 'rtgdmh',
'A2': 155,
'A3': {'B1': [{'C1': 188, 'C2': 116, 'C3': -0.1648123860359192},
               {'C1': 205, 'C2': -95, 'C3': 0.09694156795740128},
               {'C1': 154, 'C2': -44, 'C3': -0.062394171953201294},
               {'C1': 105, 'C2': 28, 'C3': 0.8352078795433044}],
'B2': 31,
'B3': [0.8211487256312548,
       -0.46170186080232667,
       0.41022953411385266,
       0.4999110338774788,
       -0.27993512247649677,
       0.7299183498484951],
```

```

'B4': 11301700218570987310},
'A4': {'D1': 896657497, 'D2': 8161299910817985809, 'D3': 0.2108570635318756},
'A5': 60365,
'A6': 17526664759437778118,
'A7': [6957, 52564, 25704, 21193, 17363, 45048],
'A8': {'E1': 0.45694947242736816, 'E2': [-83, -60, 49, -26, 36, -41, -3, 40]}}

```

**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.chip()</code>	1
<code>o.fork()</code>	RuntimeError
<code>o.pan()</code>	8
<code>o.pan()</code>	9
<code>o.chip()</code>	3
<code>o.chip()</code>	5
<code>o.pan()</code>	2
<code>o.pan()</code>	RuntimeError
<code>o.chip()</code>	3
<code>o.fork()</code>	4
<code>o.pan()</code>	7
<code>o.chip()</code>	RuntimeError
<code>o.pan()</code>	9
<code>o.chip()</code>	3
<code>o.fork()</code>	4
<code>o.chip()</code>	6

## 2. Пример использования класса C32:

<code>o = C32()</code>	
<code>o.fork()</code>	0
<code>o.pan()</code>	2
<code>o.chip()</code>	3
<code>o.fork()</code>	4
<code>o.chip()</code>	6
<code>o.pan()</code>	8
<code>o.chip()</code>	RuntimeError
<code>o.pan()</code>	9
<code>o.pan()</code>	RuntimeError
<code>o.chip()</code>	3
<code>o.chip()</code>	5
<code>o.chip()</code>	RuntimeError
<code>o.pan()</code>	2
<code>o.pan()</code>	RuntimeError
<code>o.chip()</code>	3
<code>o.pan()</code>	RuntimeError
<code>o.fork()</code>	4
<code>o.pan()</code>	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 0x47 0x59 0x58, 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	Размер (uint16) и адрес (uint16) массива адресов (uint32) структур В
	2	int8
	3	uint8
	4	Структура С
	5	float

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

Структура С:	1	Размер (uint16) и адрес (uint16) массива uint8
	2	uint8
	3	Адрес (uint32) структуры D
	4	float
	5	Размер (uint16) и адрес (uint16) массива float
	6	int16

Структура D:	1	Массив uint8, размер 3
	2	int16
	3	uint64

Примеры разбора двоичного формата с помощью функции f31:

### 1. Двоичные данные:

```
(b'GYX\x03\x00F\x00\x8b\xb7\x05\x00R\x00JW\x00\x00\x002F\xbf=\x04\x00d\x00\xae'
b'\xdb"\x0f\xbfzlwdr\xde\x18\x05\x00\x00\x00 \x00lo\r\xeb\x02\x00\x00'
b'\x00-\x00rrwepz\xd2\xd9\x07\x00\x00\x007\x00%\x00\x00\x00/\x00\x00\x00>\x00'
b'\x00\x00\xe7n\xf6\ry)\x9cN\xf8U\x05\xbf[\xb9{\xdb,\xd8I\x8e\xd4\xbe\xa60(?)'
b'9r\x0c?\x8e\xfb8\xbf')
```

Результат разбора:

```
{'A1': [{'B1': 6366, 'B2': 'zlwdr'},
        {'B1': 60173, 'B2': 'lo'},
        {'B1': 55762, 'B2': 'rrwepz'}],
'A2': -117,
'A3': 183,
'A4': {'C1': [231, 110, 246, 13, 121],
        'C2': 74,
```

```

'C3': {'D1': [41, 156, 78], 'D2': 22008, 'D3': 15577066535604961029},
'C4': 0.09339560568332672,
'C5': [-0.4151480495929718,
        0.6569923162460327,
        0.5486178994178772,
        -0.7225884199142456],
'C6': 26090},
'A5': -0.5591256022453308}

```

## 2. Двоичные данные:

```

(b'GYX\x04\x00S\x00\xad\xc2\x05\x00c\x00\xffh\x00\x00\x00\\\x9c\x1b\xbd\x03\x00'
 b'u\x00!0>\xe9\x16?tak\xdd\x9c\x03\x00\x00\x00 \x00bsadwkb\xe5i'
 b'\x07\x00\x00\x00+\x00vo\xa8\x8c\x02\x00\x00\x00:\x00vczwiwt\xe6'
 b'\xa1\x07\x00\x00\x00D\x00#\x00\x00\x002\x00\x00\x00<\x00\x00\x00K'
 b'\x00\x00\x00\x9b\xf2\x8a\xb3q7\xad\xf9\x1a\x1b\x1e\xc3\xe6\xe64\xd0\xed'
 b'\x96\xb7\x80\xf0>\x8c\xba\xb4\xbe\xf0,f?')

```

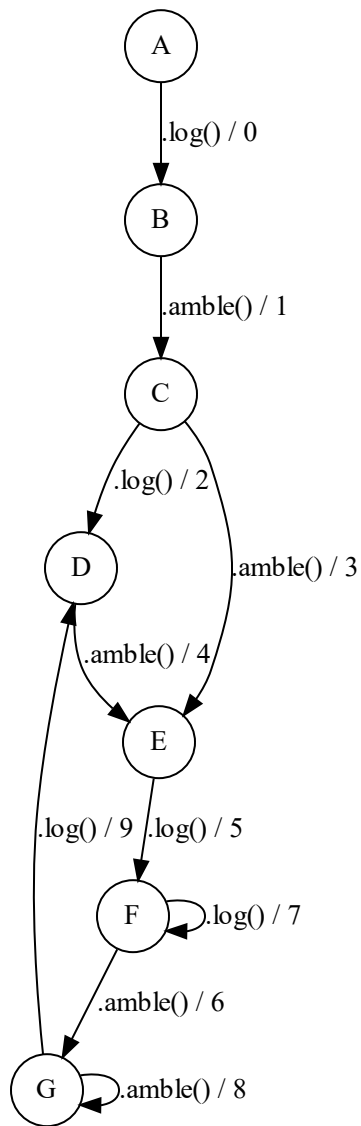
Результат разбора:

```

{'A1': [{'B1': 40157, 'B2': 'tak'},
        {'B1': 27109, 'B2': 'bsadwkb'},
        {'B1': 36008, 'B2': 'vo'},
        {'B1': 41446, 'B2': 'vczwiwt'}],
'A2': -83,
'A3': 194,
'A4': {'C1': [155, 242, 138, 179, 113],
        'C2': 255,
        'C3': {'D1': [55, 173, 249], 'D2': 6938, 'D3': 10875577600800375582},
        'C4': -0.037990912795066833,
        'C5': [0.46973201632499695, -0.3529857397079468, 0.8991231918334961],
        'C6': 20257},
'A5': 0.5894964933395386}

```

**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.amble()	RuntimeError
o.log()	0
o.amble()	1
o.log()	2
o.amble()	4
o.amble()	RuntimeError
o.log()	5
o.log()	7
o.amble()	6
o.amble()	8
o.amble()	8
o.log()	9
o.amble()	4

## 2. Пример использования класса C32:

o = C32()	
o.amble()	RuntimeError
o.log()	0
o.log()	RuntimeError
o.amble()	1
o.amble()	3
o.amble()	RuntimeError
o.log()	5
o.log()	7
o.amble()	6
o.amble()	8
o.amble()	8
o.log()	9
o.amble()	4
o.amble()	RuntimeError
o.log()	5
o.amble()	6
o.amble()	8
o.log()	9

## Вариант №26

**Задача 3.1.** Реализовать разбор двоичного формата данных (в духе формата WAD игры Doom или графического формата PNG). Данные начинаются с сигнатуры 0x58 0x4f 0x4c 0xbb, 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	int64
	4	int32
	5	Массив char, размер 6
	6	Структура B
	7	double
	8	Адрес (uint16) структуры D

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

Структура C:	1	uint8
	2	Массив uint8, размер 5

Структура D:	1	Массив float, размер 2
	2	uint8
	3	uint32
	4	float
	5	int16

Примеры разбора двоичного формата с помощью функции f31:

### 1. Двоичные данные:

```
(b'X0L\xbb\x05u\xf85\x98\x0ep*B\xdc\x07?\xe5`E\xf6-W\x8a\xfb\xbdQ\xcd\xba\xhpq'
 b'dm{\x86\x93\xa7\x18A*J}\xd1&\xaf\xf6\\\x96\x81\x987-\x83a6\x8bj1\x05\x9b'
 b'.\xa2_\xbfn\xe8\xe2\xbfF\x00\xeb\xee\xce>D\x87c>\xa0J\x06\xbd\x01\xb1'
 b'\xd3\xa9\xbeZ\x80')
```

Результат разбора:

```
{'A1': 3057960193887859973,
 'A2': 0.5307046175003052,
 'A3': -321348568486682395,
 'A4': -1160949315,
 'A5': 'xhpqdm',
 'A6': {'B1': [{'C1': 123, 'C2': [134, 147, 167, 24, 65]},
               {'C1': 42, 'C2': [93, 209, 38, 175, 246]}],
```

```

        {'C1': 92, 'C2': [150, 129, 152, 55, 45]}},
        'B2': 11170388784434274691},
        'A7': -0.5908731210484512,
        'A8': {'D1': [0.4041665494441986, 0.22219568490982056],
        'D2': 160,
        'D3': 29165149,
        'D4': -0.33169320225715637,
        'D5': -32678}}

```

## 2. Двоичные данные:

```

(b'X0L\xbb\xce\xa0=\xa2s5K\x17\x0c!A\xbf\xdc/k\x91\xa4\xa6S\x92\xe4\xf0noeqws'
 b'hr\x00f\x8b4-\xae\xb4\x94\x1eH\xec$\xaa\x97\xa2\t\xa3\xd4\xf2powi0\x1e\xbf'
 b'\x9aV\xfa\xbd:\xae\xe4\xbfF\x000\x8f\x00\xbfz\xe5k?E9\xd9+\xc1\xa2'
 b'\xf5$\xbd\x94`')

```

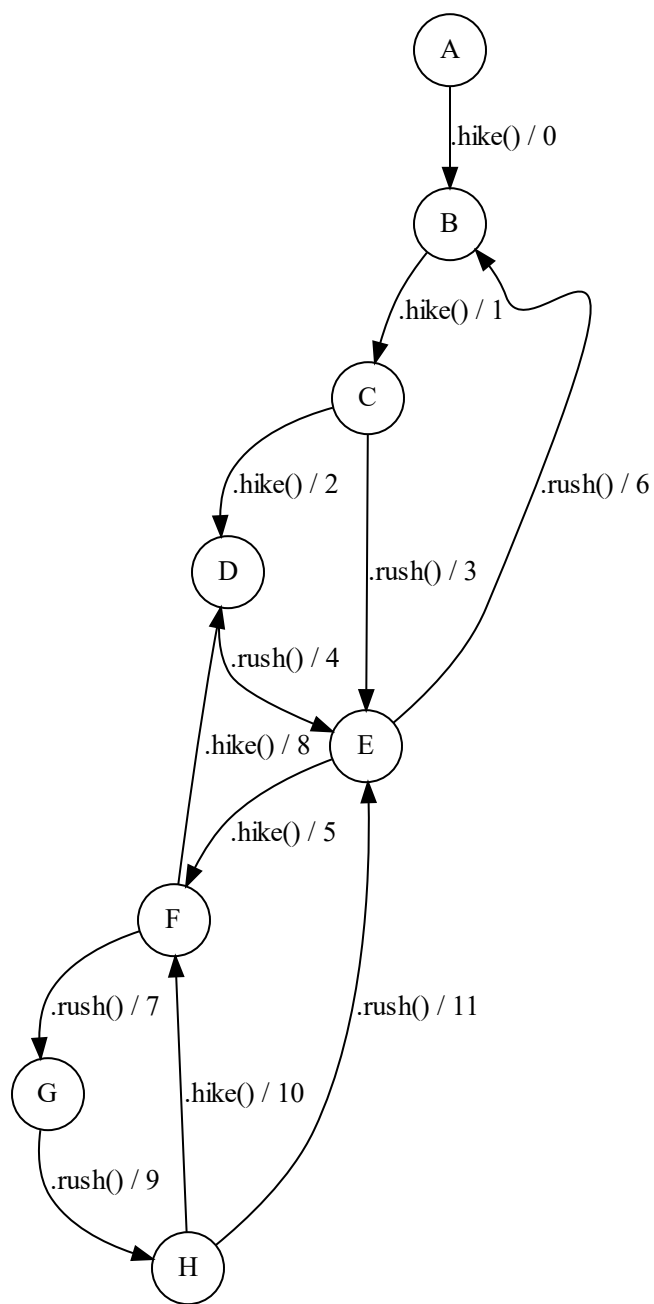
### Результат разбора:

```

{'A1': 1678494056885100750,
 'A2': -0.7544105052947998,
 'A3': -7902789695360520228,
 'A4': 1869541604,
 'A5': 'eqwshr',
 'A6': {'B1': [{'C1': 0, 'C2': [102, 139, 52, 45, 174]},
                {'C1': 180, 'C2': [148, 30, 72, 236, 36]},
                {'C1': 170, 'C2': [151, 162, 9, 163, 212]}],
        'B2': 13771532024939507954},
 'A7': -0.6462682447915598,
 'A8': {'D1': [-0.5021867156028748, 0.9214702844619751],
        'D2': 69,
        'D3': 3240876345,
        'D4': -0.04027331620454788,
        'D5': 24724}}

```

**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.hike()	0
o.rush()	RuntimeError
o.hike()	1
o.rush()	3
o.rush()	6
o.hike()	1
o.hike()	2
o.rush()	4
o.hike()	5
o.rush()	7
o.rush()	9
o.hike()	10
o.rush()	7
o.rush()	9
o.rush()	11
o.hike()	5
o.hike()	8

## 2. Пример использования класса C32:

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

## Вариант №27

**Задача 3.1.** Реализовать разбор двоичного формата данных (в духе формата WAD игры Doom или графического формата PNG). Данные начинаются с сигнатуры 0x13 0x50 0x56 0x49, 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	Размер (uint16) и адрес (uint16) массива char
	3	uint8
	4	int16
	5	int32
	6	int64
	7	Адрес (uint16) структуры B
	8	Адрес (uint32) структуры C

Структура B:	1	uint64
	2	int16

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

Структура D:	1	Массив uint8, размер 8
	2	Массив uint8, размер 5
	3	int8
	4	int32

Примеры разбора двоичного формата с помощью функции f31:

### 1. Двоичные данные:

```
(b'\x13PVI\x87\x00\x02\x00\x1e\xfd\xc2dz\x9c\x9a\xcd.\xdfk\xf9e\x06\xcd\xdb'
b'\x00 \x00\x00\x00fbfp\x85\xf3\xaa>\x8e\xcb\x18\xb3X\x15\x13%\x86\xe9\xcc'
b'g\x9f\xbb\xecaH\x8cH\x994\xb3\xa0\x84\xcf\r\xb9\tl\xf3\xf0\x03R\xd1\xaa'
b'\xc3\xde{\xfdd\x80(\xe8k ~{\x16\x1f\xe0XC7-d\xe0\x04\xdd\x00*\x00<'
b'\x00N\xbfBz\x88\xc2\x00\x00\x00\x03\x00\x00\x00`')
```

Результат разбора:

```
{'A1': -121,
'A2': 'bf',
'A3': 253,
'A4': -15772,
'A5': 2057083597,
'A6': 3377536964437134811,
```

```

'A7': {'B1': 8108154616669981464, 'B2': -19624},
'A8': {'C1': -0.7596821784973145,
      'C2': -62,
      'C3': [{'D1': [21, 19, 37, 134, 233, 204, 103, 159],
                'D2': [187, 236, 97, 72, 140],
                'D3': 72,
                'D4': -1724599392},
              {'D1': [132, 207, 13, 185, 9, 108, 243, 240],
                'D2': [3, 82, 209, 170, 195],
                'D3': -34,
                'D4': 2080203904},
              {'D1': [40, 232, 107, 32, 126, 123, 22, 31],
                'D2': [224, 48, 88, 67, 55],
                'D3': 45,
                'D4': 1692402909}]}}

```

## 2. Двоичные данные:

```

(b'\x13PVI5\x00\x02\x00\x1ef[F\xc5\xf2\xea~\x83\xab\x13\xb0\xe9\xf5&\xf1'
 b'\x00 \x00\x00\x00R1a7\x96\x12\t\xbf\x86\x02\xd2\x04\x8a{\xc4\xd6\xc4\x99\xc3'
 b'S\xa1\x06\x1c<\x15vF\xdb\xe8\xb3\xf5\xe5\x7fV{\xd9\x07c\tUmm\xec\x82he\x11'
 b'h\x80\x00*\x00<\xbdUL\x1b\x00\x00\x00\x00\x02\x00\x00\x00N')

```

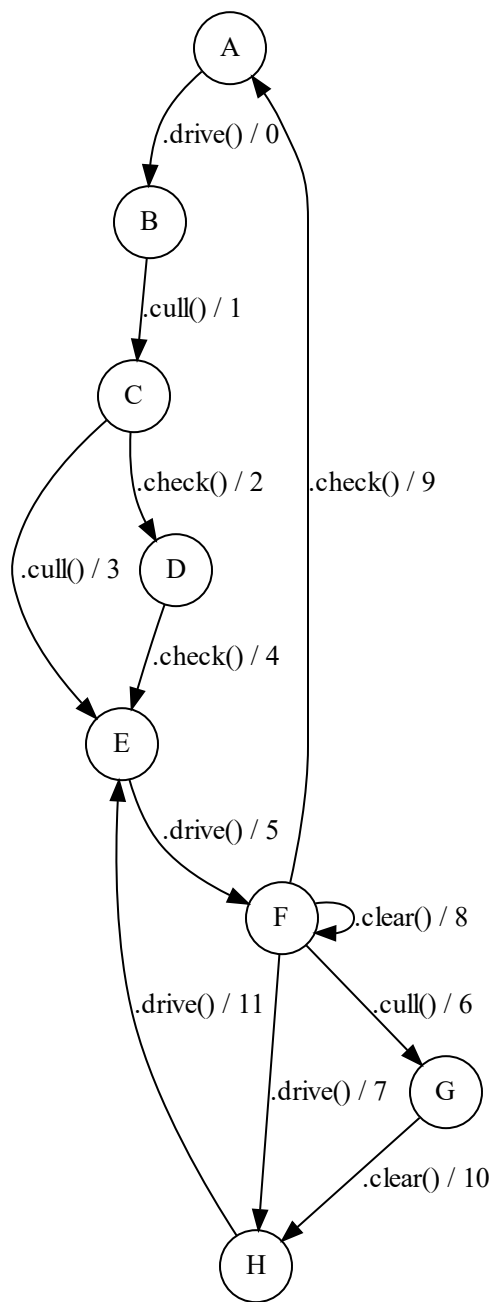
## Результат разбора:

```

{'A1': 53,
 'A2': '1a',
 'A3': 102,
 'A4': 23366,
 'A5': -973936002,
 'A6': -8959045383163140367,
 'A7': {'B1': 4005408751669871314, 'B2': 1162},
 'A8': {'C1': -0.052074532955884933,
      'C2': 0,
      'C3': [{'D1': [123, 196, 214, 196, 153, 195, 83, 161],
                'D2': [6, 28, 60, 21, 118],
                'D3': 70,
                'D4': -605506571},
              {'D1': [229, 127, 86, 123, 217, 7, 99, 9],
                'D2': [85, 109, 109, 236, 130],
                'D3': 104,
                'D4': 1695639680}]}}

```

**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.drive()	0
o.cull()	1
o.cull()	3
o.drive()	5
o.clear()	8
o.check()	9
o.drive()	0
o.cull()	1
o.check()	2
o.check()	4
o.drive()	5
o.drive()	7
o.drive()	11
o.drive()	5
o.cull()	6
o.clear()	10

## 2. Пример использования класса C32:

o = C32()	
o.drive()	0
o.cull()	1
o.cull()	3
o.drive()	5
o.cull()	6
o.clear()	10
o.drive()	11
o.drive()	5
o.check()	9
o.drive()	0
o.cull()	1
o.check()	2
o.check()	4
o.drive()	5
o.clear()	8
o.drive()	7

## Вариант №28

**Задача 3.1.** Реализовать разбор двоичного формата данных (в духе формата WAD игры Doom или графического формата PNG). Данные начинаются с сигнатуры 0x45 0x45 0x46 0x41, 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	Адрес (uint32) структуры B
	3	Структура F
	4	int16

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

Структура C:	1	uint8
	2	int8
	3	Массив uint8, размер 2

Структура D:	1	int32
	2	int8
	3	uint64
	4	Массив int32, размер 3
	5	uint32

Структура E:	1	double
	2	uint64
	3	uint16
	4	int64

Структура F:	1	int16
	2	int32
	3	int32
	4	Размер (uint16) и адрес (uint16) массива int16

Примеры разбора двоичного формата с помощью функции f31:

### 1. Двоичные данные:

```
(b'EEFA\x00\x03\x00\x1c\x00\x00\x00I\xb5\x8c\xbc8P\\\xc0\x92m\xa3\x00\x04'
 b'\x00p\x80\x89dve\xfc\t%\xf9\xc3\nP\xb4\x00\x00\x00\x1f\x00\x00\x00#\xbf'
 b"\xea\xbf\x96\xdc\xd5ZH\xf9b\xba#\x91\xd8\xf0\x1e\x87!\x16'\x81\x91\xdfMa"
 b"\xa6\x00\x02\x00\x00\x00'z\xc8\x9b\x1e;yL\x1a\xad?\x92\x04\x91M<\x82\xb6"
 b'~G\xee\xf4\x11\xe4\x1e\xd3\x0c\xfb\n\x9d\x00\x00\x00/\xf5\xd1!j\xa7CrF')
```

Результат разбора:

```
{'A1': 'dve',
'A2': {'B1': [{'C1': 252, 'C2': 9, 'C3': [37, 249]},
{'C1': 195, 'C2': 10, 'C3': [80, 180]}],
'B2': {'D1': 2059967262,
'D2': 59,
'D3': 8740390308217488529,
'D4': [1295811254, 2118643444, 300162771],
'D5': 217778845},
'B3': {'E1': -0.8358873665955082,
'E2': 17970130125095235614,
'E3': 34593,
'E4': 1596387056442761638}},
'A3': {'F1': -19060,
'F2': -1137160100,
'F3': -1064145501,
'F4': [-2607, 8554, -22717, 29254]},
'A4': -32631}
```

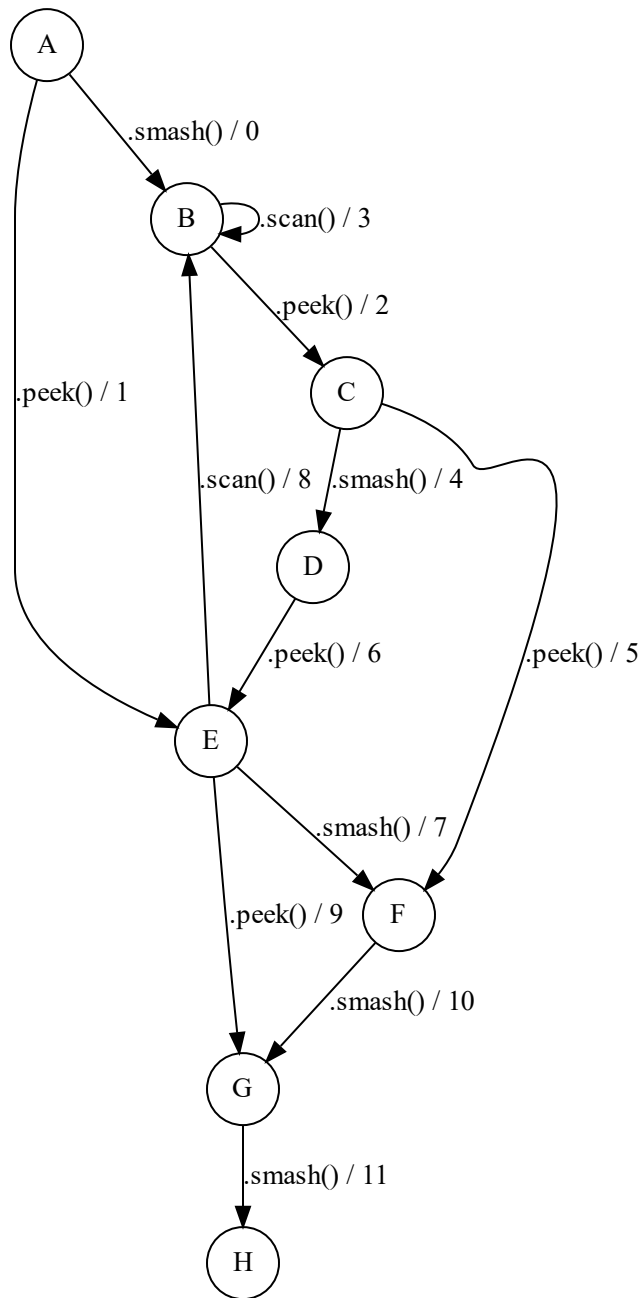
## 2. Двоичные данные:

```
(b'EEFA\x00\x07\x00\x1c\x00\x00\x00U\x1eH,\x0f\xa6L[A\x1c\x81\x00\x03\x00|_\x87'
b"ditjaim\xa9s5%[F\x96\xc8\xbd\xfc2@\x00\x00\x00#\x00\x00\x00'\x00\x00\x00+?"
b'\xe7\xf4T\xf4\xd2C\xfc`}L\xc4\x1d\x9f\x92L\\b1T\x97\x9e\xc9\xffm\xfc'
b'"'\x00\x03\x00\x00\x00/\xb3.+: \xe7\xd2\x9b\xd2\x914\x18Z\x15[\xd9\xfazzW\x07o'
b'\x1a\x91\xa7\xcfx9f\xfa\x1fR\x00\x00\x00;\xe5&yg\xe6(')
```

## Результат разбора:

```
{'A1': 'ditjaim',
'A2': {'B1': [{'C1': 169, 'C2': 115, 'C3': [53, 37]},
{'C1': 37, 'C2': 91, 'C3': [70, 150]},
{'C1': 200, 'C2': -67, 'C3': [242, 64]}],
'B2': {'D1': -1288819910,
'D2': -25,
'D3': 15175954890441120277,
'D4': [1541012090, 2052523887, 445753295],
'D5': 2683969362},
'B3': {'E1': 0.7485756665865888,
'E2': 6952797804924211788,
'E3': 23729,
'E4': 6095515211080268834}},
'A3': {'F1': 7752,
'F2': 739223116,
'F3': 1530993793,
'F4': [-6874, 31079, -6616]},
'A4': 24455}
```

**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.peek()</code>	1
<code>o.scan()</code>	8
<code>o.peek()</code>	2
<code>o.smash()</code>	4
<code>o.peek()</code>	6
<code>o.scan()</code>	8
<code>o.scan()</code>	3
<code>o.peek()</code>	2
<code>o.smash()</code>	4
<code>o.peek()</code>	6
<code>o.smash()</code>	7
<code>o.peek()</code>	RuntimeError
<code>o.smash()</code>	10
<code>o.smash()</code>	11

## 2. Пример использования класса C32:

<code>o = C32()</code>	
<code>o.smash()</code>	0
<code>o.scan()</code>	3
<code>o.scan()</code>	3
<code>o.peek()</code>	2
<code>o.smash()</code>	4
<code>o.peek()</code>	6
<code>o.scan()</code>	8
<code>o.scan()</code>	3
<code>o.scan()</code>	3
<code>o.scan()</code>	3
<code>o.scan()</code>	3
<code>o.scan()</code>	3
<code>o.peek()</code>	2
<code>o.peek()</code>	5
<code>o.smash()</code>	10
<code>o.smash()</code>	11

## Вариант №29

**Задача 3.1.** Реализовать разбор двоичного формата данных (в духе формата WAD игры Doom или графического формата PNG). Данные начинаются с сигнатуры 0x54 0x48 0x4d 0x1b, 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	uint64

Структура B:	1	uint32
	2	Адрес (uint16) структуры C

Структура C:	1	Массив структур D, размер 5
	2	uint8
	3	Массив float, размер 2
	4	Адрес (uint16) структуры E

Структура D:	1	uint32
	2	uint32

Структура E:	1	Массив int8, размер 3
	2	uint16
	3	Массив int16, размер 2
	4	uint8
	5	float
	6	Размер (uint32) и адрес (uint32) массива int8
	7	int8

Примеры разбора двоичного формата с помощью функции f31:

1. Двоичные данные:

```
(b'THM\x1b\x00[\x19\x8e\xf5\x8b\xff\xc6\x84j\xff\x96\x14\x94\xa0?Y\xde\xb8\xdb'
b"\xd0\x1eD;\xd3\xfb\x00\x00\x00\x03\x00\x00\x00\x0e\x860\xe4\t\xeb"
b'\x9a\xac<\x15\x1291\x8a\xbf\xa5Ay\x88I\xc1kA\x1a\xde\x07Z\xe6\xc00'
b'1\xab\xe4\x1csk\xde\xf8\x01\xdd\x9ej%?{\xf1\xc4\xbfV\xe8\x1c\x00\x11\xb1'
b'\x951\x85\x00(')
```

Результат разбора:

```
{ 'A1': { 'B1': 2979359877,
          'B2': { 'C1': [ { 'D1': 820251115, 'D2': 2594978837 },
                        { 'D1': 305738122, 'D2': 3215278457 },
                        { 'D1': 2286535019, 'D2': 1092279815 },
                        { 'D1': 1525071951, 'D2': 833348636 },
```

```

        {'D1': 1936449272, 'D2': 31301226}],
    'C2': 37,
    'C3': [0.9841578006744385, -0.8394792079925537],
    'C4': {'E1': [-108, -96, 63],
           'E2': 23006,
           'E3': [-18213, -12258],
           'E4': 68,
           'E5': 0.0025608527939766645,
           'E6': [-1, -106, 20],
           'E7': -122}}},
    'A2': 1841679279281570922}

```

## 2. Двоичные данные:

```

(b"THM\x1b\x00]\xdd\xcb\x8b\x0e=-\x07\x94\xb6&\x00\xa6i#H\x91\xa3!\x04\xbd\xc9"
b'\xb9>K\x1c\x12\x00\x00\x00\x05\x00\x00\x00\x0e19\xa0\x08\xfb0\xba\xe7S[\xc0'
b'\xd59\xe3\x0c\x142J2\xc9\x96\x810c_\xcb\xf1`\xd7\xa2m\xee\x7f\x9a&'
b'\x0e\x96\xa0G\x8d\x89W>\xb2\xdb\xe6:#nu\x00\x13D\x13\x9e\xc7\x00*')

```

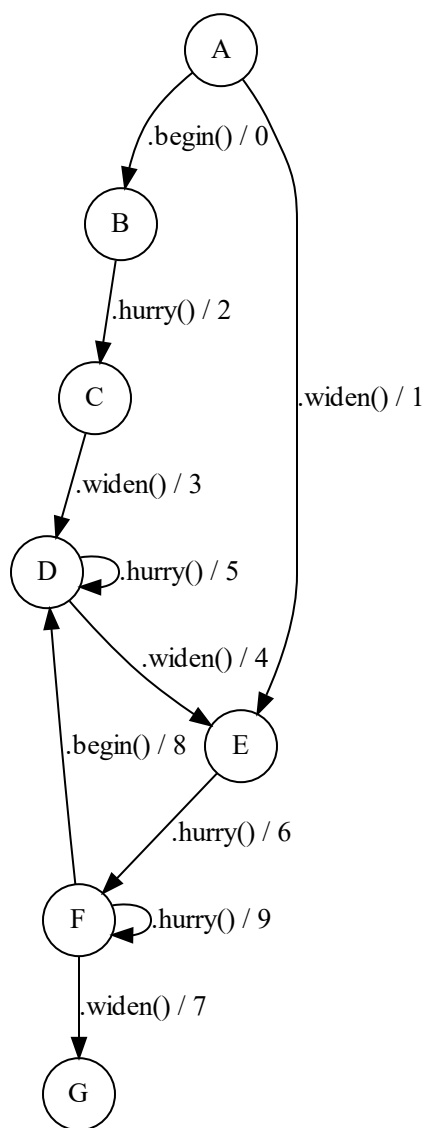
## Результат разбора:

```

{'A1': {'B1': 1142136519,
        'B2': {'C1': [{'D1': 966789371, 'D2': 1337648979},
                       {'D1': 1539364153, 'D2': 3809219634},
                       {'D1': 1244842390, 'D2': 2167432031},
                       {'D1': 3421593815, 'D2': 2725113471},
                       {'D1': 2586185366, 'D2': 2689043849}]},
        'C2': 87,
        'C3': [0.3493339419364929, 0.0006234415923245251],
        'C4': {'E1': [35, 72, -111],
                 'E2': 41761,
                 'E3': [9988, -16951],
                 'E4': 185,
                 'E5': 0.198349267244339,
                 'E6': [-74, 38, 0, -90, 105],
                 'E7': 108}}},
    'A2': 15982020595926501268}

```

**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.hurry()	RuntimeError
o.begin()	0
o.hurry()	2
o.widen()	3
o.hurry()	5
o.hurry()	5
o.widen()	4
o.hurry()	6
o.begin()	8
o.widen()	4
o.hurry()	6
o.hurry()	9
o.widen()	7

## 2. Пример использования класса C32:

o = C32()	
o.begin()	0
o.hurry()	2
o.widen()	3
o.hurry()	5
o.widen()	4
o.hurry()	6
o.hurry()	9
o.begin()	8
o.begin()	RuntimeError
o.widen()	4
o.hurry()	6
o.hurry()	9
o.widen()	7

## Вариант №30

**Задача 3.1.** Реализовать разбор двоичного формата данных (в духе формата WAD игры Doom или графического формата PNG). Данные начинаются с сигнатуры 0x41 0x5a 0x4f 0xc3, 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	int32
	2	Размер (uint32) и адрес (uint16) массива адресов (uint32) структур В
	3	Структура С
	4	int64
	5	int16

Структура В:	1	uint8
	2	Размер (uint16) и адрес (uint32) массива char

Структура С:	1	double
	2	float
	3	Адрес (uint16) структуры D
	4	double
	5	Структура E

Структура D:	1	int8
	2	Размер (uint32) и адрес (uint32) массива int64
	3	double
	4	Размер (uint16) и адрес (uint32) массива int8

Структура E:	1	int64
	2	int64
	3	int32
	4	double
	5	uint16

Примеры разбора двоичного формата с помощью функции f31:

1. Двоичные данные:

```
(b'AZ0\xc3\xb93\xb13\x00\x00\x00\x02\x00^\xbf\xe6^\x8e\xa9\x14J\x02>\x8c'  
b'\x97\xef\x00\x85?\xc4\x05)~\xb2\xe6`#\xa0r\xe5\xae<K0\xf2\xcf\x8e\xa0'  
b'\xd4\x9c\xdb\xda\xf4\x11\xdf\x15\xbf\xe4cf{\xe3\xf4\xe2\xa6\xe0\xe0E'  
b'K\xee\x92s\x12V\x85\xd8oq\xa5\x00\x02\x00\x00\x00Lg1\xb1\x00\x02\x00\x00'  
b'\x00U\x00\x00\x00N\x00\x00\x00W\x82j\xcb\x01\xb0\x93\xbc\x1c'  
b'\xc8\xe6\xea\xc0\x07\xbe+\xba\x14\x03J\x13\x0e\xb2\x81@\x98\xf3/A'  
b';\xf6\x00\x00\x00\x03\x00\x00\x00f?\xa2\x00#\xe1\x14\x93\x00\x00\x07'  
b'\x00\x00\x00~')
```

Результат разбора:

```
{'A1': -1187794637,
'A2': [{'B1': 165, 'B2': 'oq'}, {'B1': 177, 'B2': 'gl'}],
'A3': {'C1': -0.6990426351861176,
'C2': 0.2745966613292694,
'C3': {'D1': -10,
'D2': [-9049197293131492420,
7502091973398890430,
3150852893659303602],
'D3': 0.03515731928408705,
'D4': [-127, 64, -104, -13, 47, 65, 59]}},
'C4': 0.1564075344702358,
'C5': {'E1': 2567178118397446991,
'E2': -950384174938924070,
'E3': -200155371,
'E4': -0.6371338290915924,
'E5': 42720}},
'A4': -2286337747789344170,
'A5': -31272}
```

## 2. Двоичные данные:

```
(b'AZ0\xc3\x89\xc5\x15N\x00\x00\x00\x06\x00\x82?\xc8\x9a10\xa2W\xc8\xbe\xee'
b'\x07\xc4\x00\xb7\xbf\xec,\xe7!d\x07V\x13\xfc~\xac\xda\t\xf4\x10\x1fz\x8a\xe'
b'\xf7\xcf\xb5\x0f\xeb\x1b\\\xd0?\xe7\xf7!\x8b\xa7 \x1c\x1d,s|\xbe%\x84\xd0'
b'\x01Iq\x92emF\x00\x02\x00\x00\x00Loa\x96\x00\x02\x00\x00\x00Uti'
b'\x87\x00\x02\x00\x00\x00^neh\x00\x02\x00\x00\x00gumn\x00\x02\x00\x00\x00poe#'
b'\x00\x02\x00\x00\x00y\x00\x00\x00N\x00\x00\x00W\x00\x00\x00`\x00\x00'
b'\x00i\x00\x00\x00r\x00\x00\x00{\xe4\xc8\x15R\xf4\xae\x16\xe3H_\xfe\x13\x9b\r'
b'\xf7\x1e-\x9a\xa6 \xf4Ph\xf6\x82\xdb\xdb\xe7>d\x00\x00\x00\x03'
b'\x00\x00\x00\x9a\xbf\xd4\xf4\r\xf0\xe9\x98\xe4\x00\x05\x00\x00\x00\xb2')
```

## Результат разбора:

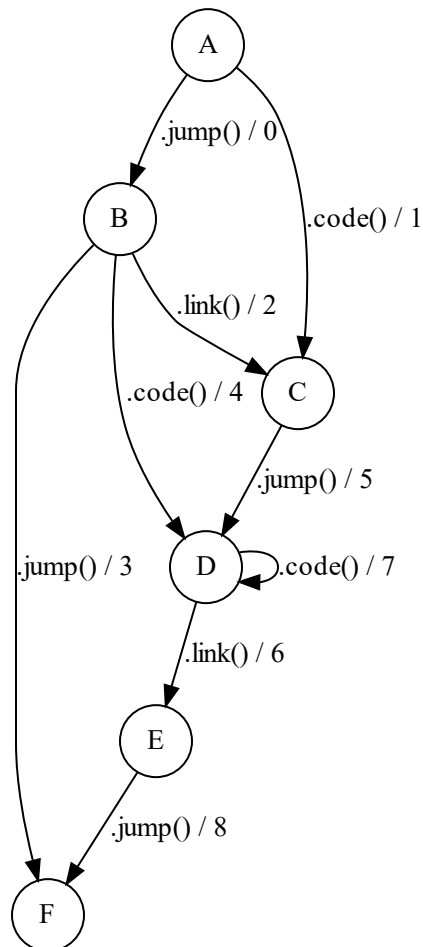
```
{'A1': -1983572658,
'A2': [{'B1': 70, 'B2': 'em'},
{'B1': 150, 'B2': 'oa'},
{'B1': 135, 'B2': 'ti'},
{'B1': 104, 'B2': 'ne'},
{'B1': 110, 'B2': 'um'},
{'B1': 35, 'B2': 'oe'}],
'A3': {'C1': 0.1922055853689797,
'C2': -0.46490299701690674,
'C3': {'D1': 100,
'D2': [-1961294191683299613,
5215166253677541150,
3286121538588272886],
```

```

'D3': -0.3273959016609196,
'D4': [-126, -37, -41, -25, 62]},
'C4': -0.8804813053286391,
'C5': {'E1': 1440165261709276176,
'E2': 2268277708982039823,
'E3': -350528304,
'E4': 0.748917362937394,
'E5': 7468}},
'A4': 8321735279825125705,
'A5': 29074}

```

**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.jump()</code>	0
<code>o.link()</code>	2
<code>o.jump()</code>	5
<code>o.code()</code>	7
<code>o.jump()</code>	RuntimeError
<code>o.link()</code>	6
<code>o.code()</code>	RuntimeError
<code>o.jump()</code>	8

## 2. Пример использования класса C32:

<code>o = C32()</code>	
<code>o.jump()</code>	0
<code>o.link()</code>	2
<code>o.jump()</code>	5
<code>o.code()</code>	7
<code>o.link()</code>	6
<code>o.jump()</code>	8

## Вариант №31

**Задача 3.1.** Реализовать разбор двоичного формата данных (в духе формата WAD игры Doom или графического формата PNG). Данные начинаются с сигнатуры 0x44 0x4a 0x4b 0x46, 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	Размер (uint32) и адрес (uint32) массива char
	3	uint8
	4	double
	5	Адрес (uint32) структуры B
	6	uint16
	7	uint16

Структура B:	1	Массив адресов (uint32) структур C, размер 4
	2	Структура D
	3	uint16

Структура C:	1	uint32
	2	int8
	3	Массив uint16, размер 2

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

Примеры разбора двоичного формата с помощью функции f31:

### 1. Двоичные данные:

```
(b'DJKF\xc1*\xd5C\x00\x00\x00\x02\x00\x00\x00!\b\xbf\xe7h:\x95\xd0\xa5'  
b'\x1e\x00\x00\x00gB\tqwoaJ#>b\xc4\xe2\xa4\x8aW\xd1\xd8\xd5f\xcd\xc3\xdc'  
b'/\x14\xba\x84\xe3\x02xM\xb5\xef\x87\xdc\xb8\xef\xfc\xad\xadre\xe5'  
b'\xae\xff\x00\x18c\x81\xa9\xa2\x16h\xd1\xc2R\xc7\xba\xa2m\xb04\x85'  
b'\xc8\x0f\xf1\xd90\xc7J]\x00\xd3u\xda\x00\x00\x00#\x00\x00\x00,\x00'  
b'\x00\x005\x00\x00\x00>?\xdd\xff\xaa\x98\xe0|\xd83\x02\x00\x04\x00\x00\x00Gx'  
b'\n\xacY\xc5\xa2\xf6\xe3\x84\x9a')
```

Результат разбора:

```
{'A1': -1054157501,  
'A2': 'oa',  
'A3': 98,  
'A4': -0.7314732481073951,
```

```

'A5': {'B1': [{'C1': 1243823714, 'C2': -60, 'C3': [58020, 35415]},
               {'C1': 3509967061, 'C2': 102, 'C3': [52675, 56367]},
               {'C1': 347768035, 'C2': 2, 'C3': [30797, 46575]},
               {'C1': 2279389423, 'C2': -4, 'C3': [44461, 29285]}],
      'B2': {'D1': 0.46872963837991,
              'D2': 13058,
              'D3': [16550446056507408809,
                     11679637933749618618,
                     11704204746235056113,
                     15658953631650248154],
              'D4': 8649915535886317283},
      'B3': 33946},
'A6': 16905,
'A7': 29047}

```

## 2. Двоичные данные:

```

(b'DJKF\xeewC\xcb\x00\x00\x00\x04\x00\x00\x00!\x87\xbf\xee\x5v\x17\x8e\x0c'
 b'\x84\x00\x00\x00\x89\xfc\xbf\xbe4rgsg\xef5I\\w\x0fk\xfb\xa5\xa7\x8ft\x13 k'
 b'E\xc6\xe6\x0eI\xf7\xe4\x84\xc3B\xe2m\x9aQy\xef\xcf\t\x18\xbbY\x0c\xe9N'
 b'\xff\x90\xa7q\x8ef[(\xbc?\xbe\x01\xe6[\xffzxm\xc8G13{\x81\xbb(\xf1\x1f'
 b'\x14\x181R\xef\xff\x08o\t\x9b\x95b\xb9\xc6\x8f\xb2\x0e\xae\x8d\xda=\x1e>\xc2'
 b'\xbb\x00\x1br\xa0U\xde\n)\x00\x00\x00%\x00\x00\x00.\x00\x00\x007\x00\x00\x00'
 b'@\?\xd0#\xc7Q\xe1\x81\xb0\n\xab\x00\x08\x00\x00\x00I\xa9T\x1f\x8c\xf9\xdab'
 b'\x85cg')

```

## Результат разбора:

```

{'A1': -294173749,
 'A2': 'rgsg',
 'A3': 135,
 'A4': -0.9596510372215481,
 'A5': {'B1': [{'C1': 4013246812, 'C2': 119, 'C3': [3947, 64421]},
               {'C1': 2811196435, 'C2': 32, 'C3': [27461, 50918]},
               {'C1': 239728612, 'C2': -124, 'C3': [49986, 57965]},
               {'C1': 2589030895, 'C2': -49, 'C3': [29720, 47961]}],
      'B2': {'D1': 0.25218375202350796,
              'D2': 2731,
              'D3': [930361657579237774,
                     7375533603737698790,
                     6629151842759100268,
                     3709701358915493652,
                     1759872745124425481,
                     11210975397448495630,
                     12577949290350953147,
                     7725857332136489],
              'D4': 8649915535886317283},
      'B3': 33946},
 'A6': 16905,
 'A7': 29047}

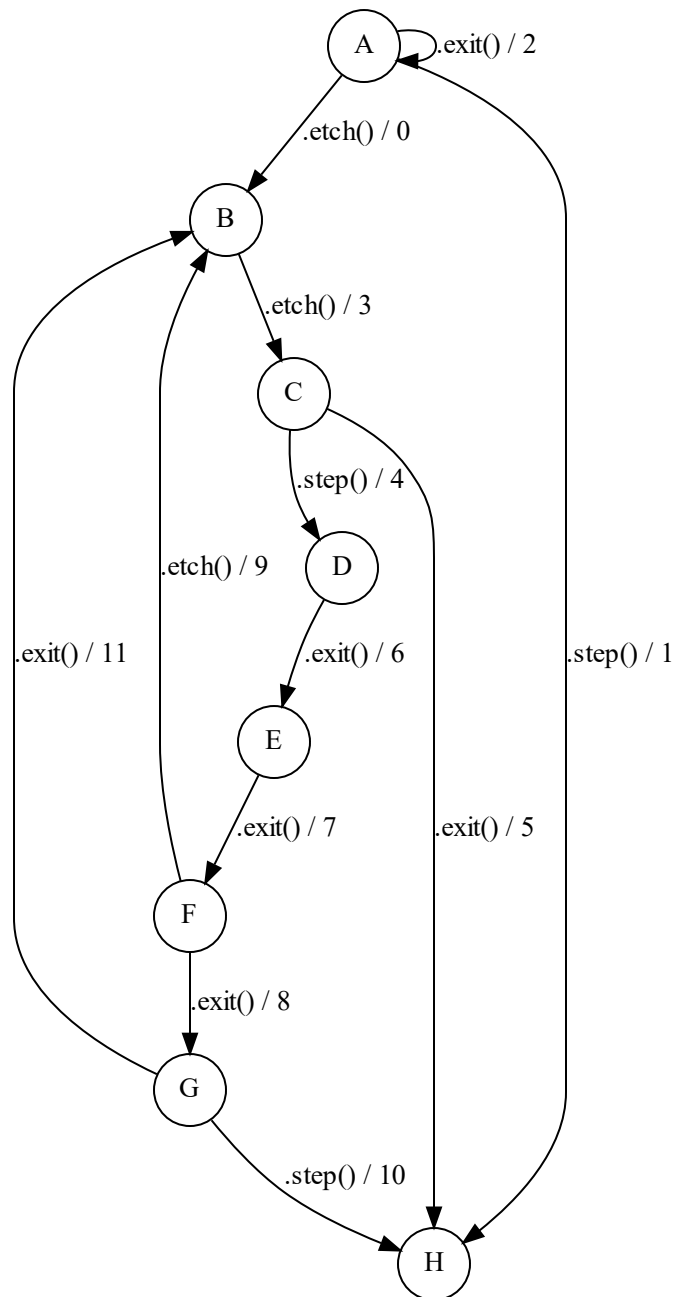
```

```

'D4': 12201411980801237637},
'B3': 25447},
'A6': 64703,
'A7': 48692}

```

**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.exit()</code>	2
<code>o.etch()</code>	0
<code>o.step()</code>	RuntimeError
<code>o.etch()</code>	3
<code>o.etch()</code>	RuntimeError
<code>o.step()</code>	4
<code>o.exit()</code>	6
<code>o.exit()</code>	7
<code>o.exit()</code>	8
<code>o.exit()</code>	11
<code>o.etch()</code>	3
<code>o.exit()</code>	5

## 2. Пример использования класса C32:

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

## Вариант №32

**Задача 3.1.** Реализовать разбор двоичного формата данных (в духе формата WAD игры Doom или графического формата PNG). Данные начинаются с сигнатуры 0x5b 0x48 0x4a 0x4d, 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	Структура В
	2	Структура Е
	3	int64
	4	uint64

Структура В:	1	uint32
	2	Адрес (uint32) структуры С
	3	int32
	4	Массив char, размер 7
	5	Размер (uint16) и адрес (uint32) массива структур D

Структура С:	1	uint16
	2	uint32
	3	int32
	4	float

Структура D:	1	uint8
	2	float
	3	Размер (uint16) и адрес (uint16) массива int8
	4	uint16
	5	int16

Структура Е:	1	uint64
	2	double
	3	Размер (uint32) и адрес (uint16) массива int8
	4	uint64

Примеры разбора двоичного формата с помощью функции f31:

### 1. Двоичные данные:

```
(b' [HJMF\b5#\x8e\x00\x00\x00K\x1f\x84\xbfTgemufbn\x00\x03\x00\x00\x00'
b' _\xda\x12L|`z\xba2\xbf\xc4fb\x89\x01R\xb0\x00\x00\x00\x05\x00\x86\xa3'
b' \x01\x11<V\xe7\xe6\x92z{H\xff\xf7w\xf9\x87N[\xa4d\x00A\xa7\xfd>\xa2\xa4\x83{'
b' \xd1h\xd2\x88\x10\xbf3^\xfe\xbb\xf6\xde\x81\xa6\xca\xbf\xbeL(\xa4'
b' \x00\x02\x00Y\xf3\x13mc\r\bfgd6\x00\x02\x00[\x88\xd0\xfc##>0\xc6j\x00\x02'
b' \x00]\x8aj\x18M,YK\xfb\xb9' )
```

Результат разбора:

```
{'A1': {'B1': 1186276238,
      'B2': {'C1': 16034,
            'C2': 2760080337,
            'C3': 1758627856,
            'C4': -0.700668215751648},
      'B3': 528793428,
      'B4': 'gemufbn',
      'B5': [{'D1': 191,
            'D2': -0.1993737816810608,
            'D3': [-69, -10],
            'D4': 62227,
            'D5': 28003},
            {'D1': 13,
            'D2': -0.9038728475570679,
            'D3': [-34, -127],
            'D4': 35024,
            'D5': -989},
            {'D1': 35,
            'D2': 0.1726318895816803,
            'D3': [-90, -54],
            'D4': 35434,
            'D5': 6221}]},
'A2': {'E1': 15713706146927393330,
      'E2': -0.15937453927712886,
      'E3': [44, 89, 75, -5, -71],
      'E4': 11745688254012712594},
'A3': 8825728158968314247,
'A4': 5646287307247036413}
```

## 2. Двоичные данные:

```
(b' [HJM\x88>\xb8\xe4\x00\x00\x00K\xad\xac\x1c\x14echijzx\x00\x04\x00\x00\x00'
 b'a\xe0\xcdL\xab6\x11e\xae?\xd6\xfe\x80\xeeDJH\x00\x00\x00\x05\x00\x95\xc3'
 b'\xbe@T\xfe~`C\x0c\x80x-z\x15\xb9R\x9c;jc\xfb\x96\x13\xeb\xf1\x83d}z'
 b"\xb9\xb6\tM|\xbff' ]\xaa\xfb\x11?:Jy\xee\xb4\xbe\xab9\x04\x00\x02\x00Y(\x9c"
 b'<\x0f\xfb3>\x9c\xc2\xd9\x00\x02\x00[4\x03\xd8:D?\x01\x8aD\x00\x02\x00] '
 b'se\xad\xb4[\xbe\t\xbeB\x00\x02\x00_\xb0\xa1\x01\x98\xef\xdc\x97\xa9r' )
```

## Результат разбора:

```
{'A1': {'B1': 2285811940,
      'B2': {'C1': 61827,
            'C2': 1685945017,
            'C3': -1240904324,
            'C4': -0.6537729501724243},
      'B3': -1381229548,
```

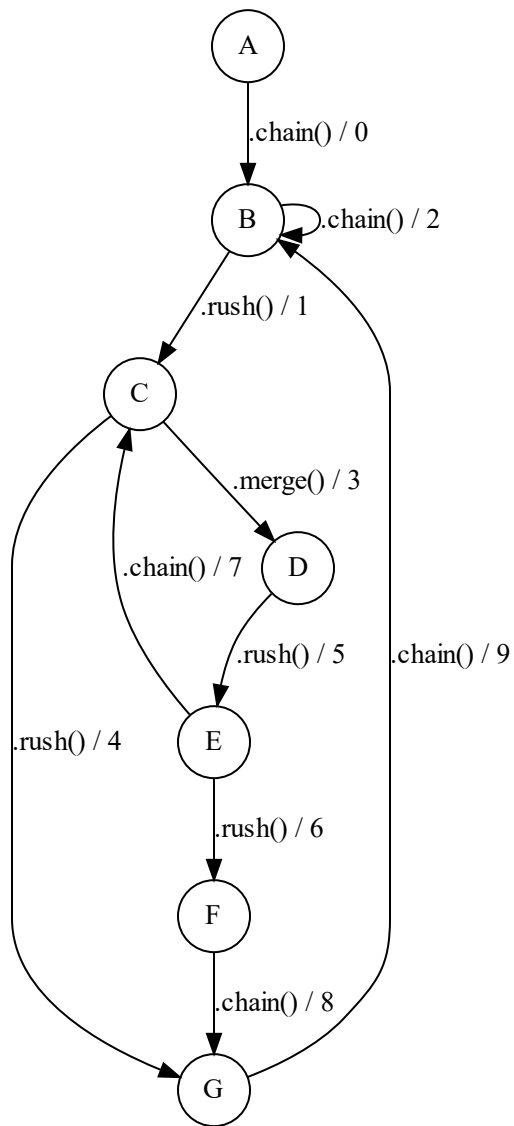
```

'B4': 'echijzx',
'B5': [{ 'D1': 180,
        'D2': -0.3344193696975708,
        'D3': [-9, 98],
        'D4': 10396,
        'D5': 15375},
      { 'D1': 243,
        'D2': 0.30617406964302063,
        'D3': [17, 63],
        'D4': 13315,
        'D5': -10182},
      { 'D1': 68,
        'D2': 0.5060160160064697,
        'D3': [58, 74],
        'D4': 29541,
        'D5': -21068},
      { 'D1': 91,
        'D2': -0.13451483845710754,
        'D3': [121, -18],
        'D4': 45217,
        'D5': 408}]],
'A2': { 'E1': 16198687732951770542,
        'E2': 0.3592836691688359,
        'E3': [-17, -36, -105, -87, 114],
        'E4': 14104781816762097731},
'A3': 900852062191204690,
'A4': 11257708671197909995}

```

**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.chain()          0
o.rush()           1
o.merge()          3
o.rush()           5
o.merge()          RuntimeError
o.chain()          7
o.rush()           4
o.chain()          9
o.rush()           1
o.merge()          3
o.rush()           5
o.rush()           6
o.chain()          8
o.chain()          9
o.chain()          2

```

## 2. Пример использования класса C32:

```

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

```

## Вариант №33

**Задача 3.1.** Реализовать разбор двоичного формата данных (в духе формата WAD игры Doom или графического формата PNG). Данные начинаются с сигнатуры 0x35 0x44 0x4e 0x4b, 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	uint32
	2	Адрес (uint16) структуры В
	3	float
	4	uint32

Структура В:	1	Структура С
	2	Адрес (uint16) структуры Е
	3	float
	4	int8
	5	int8
	6	int64
	7	Размер (uint32) и адрес (uint16) массива int8

Структура С:	1	uint32
	2	Размер (uint16) и адрес (uint16) массива адресов (uint32) структур D
	3	uint32
	4	uint16
	5	int8
	6	uint8
	7	float

Структура D:	1	uint32
	2	uint16
	3	uint16

Структура Е:	1	Структура F
	2	int16
	3	Структура G
	4	uint16

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

Структура G:	1	Размер (uint16) и адрес (uint16) массива int8
	2	uint16
	3	uint16
	4	int32

Примеры разбора двоичного формата с помощью функции f31:

#### 1. Двоичные данные:

```
(b'5DNK\xd2\x12>\x9e`\x00\x1d\x14D?j\xb2\rT\x80\xe9\x07\xb4\xe3\x9b;[\x9er'
b'G\xde\x8f\x83\xb8\xcd\x12\x00\x00\x00\x1a\x00\x00\x00\x13\xbb'
b'\xd0\xca\x1c\xfe\x13&\{\x03\x00\x00\x00*\x00\xb0f\xc9\xf1\xcd\xcf'
b'\xfc\xcc6\x17\x92p^\xa4f.\xf9G\x02\x00\x00\x00\xac1\xac\x89-\xfa{.0d\xee0'
b'\xacV"N\x19\xfc\x8dC\x02\x00"\x00\x86\xf0\x0c\xb8\xc9i\xac\xe4j\x90\xb1\xbe'
b'2\x00(\x97C>\xef\x1f\xfb\x8fI37\x81\xdd\x86\x06\x00\x00\x00Z\x00')
```

Результат разбора:

```
{'A1': 2654868178,
 'A2': {'B1': {'C1': 1133378585,
               'C2': [{'D1': 3020417408, 'D2': 39907, 'D3': 23355},
                      {'D1': 3729224350, 'D2': 33679, 'D3': 52664}],
               'C3': 3087855750,
               'C4': 27081,
               'C5': -84,
               'C6': 228,
               'C7': -0.3468049168586731},
 'B2': {'E1': {'F1': 31580,
               'F2': [-17645, -13616, -484],
               'F3': 14770909361300137648,
               'F4': 1888622390,
               'F5': 778478686},
        'E2': 18425,
        'E3': {'G1': [19, 38],
               'G2': 27820,
               'G3': 35244,
               'G4': 779876909},
        'E4': 25648},
 'B3': 0.19100630283355713,
 'B4': -17,
 'B5': 31,
 'B6': -8728678428690509829,
 'B7': [-18, 48, -84, 86, 34, 78]},
 'A3': 0.7659319043159485,
 'A4': 1410183786}
```

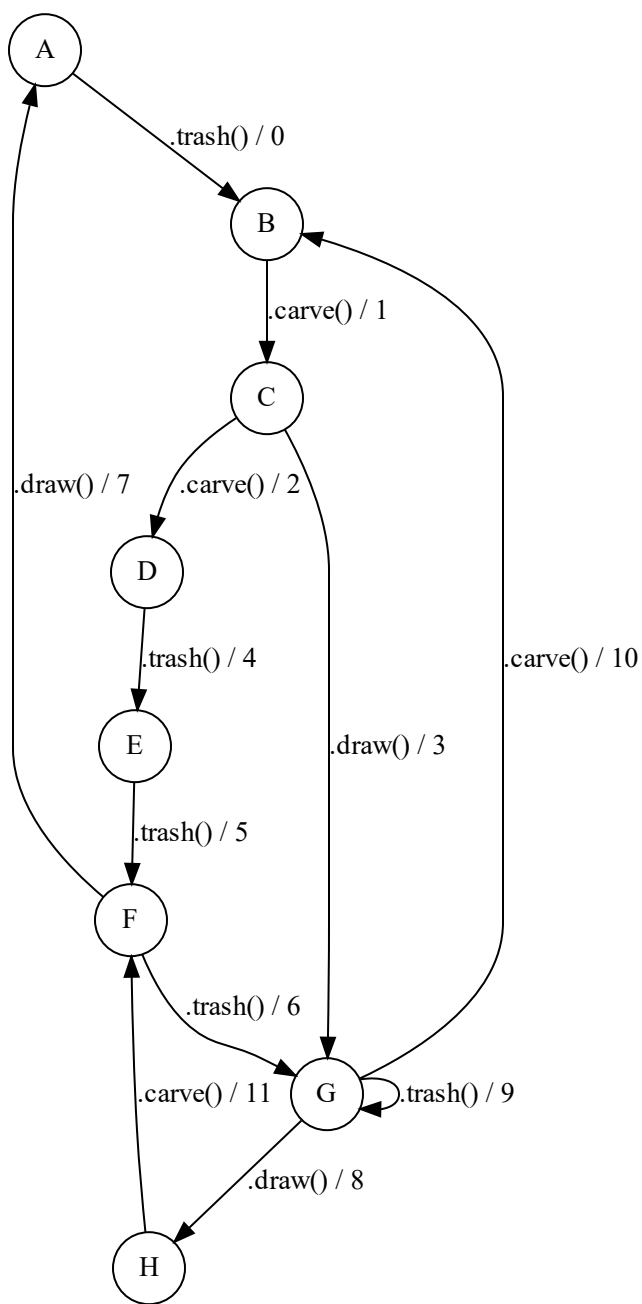
#### 2. Двоичные данные:

```
(b'5DNK\xbbb{\xca\xcb`\x00\xc4\x7fd\xbf\x93\x9e9G\xae\xe7\x12\x8eN\x88'
b"4\xa7\xd7\x1e\xb7U\x18\x07~u\x12\x00\x00\x00\x1a\x00\x00\x00'>2+^wD\x94+\xa7"
b'\xdfK\x03\x00\x00\x00*\x00\xea\x83f\x12\xdb\xfb\x5\xc6\x07\x88\xc3\xb4\x1b'
b"? \x9d\xdb\x1epT\x04\x000\x00\xe6\x9f*\xbd'ojp:\xd9\x12\x1ey\x8c"
b'\xda\xd8\xa8\xbd\x02\x00"\x00/(\r\xba\xadG\x8d\xedY\x98^\xbf4\x00\xa2\x9c'
b'z\xbf\xd3\x87\x1f\x8f0\xdc\x80\xfa\xfb\x6\xd1\x04\x00\x00\x00\\\x00')
```

Результат разбора:

```
{'A1': 3419044795,
'A2': {'B1': {'C1': 3181959386,
'C2': [{'D1': 2383603630, 'D2': 34894, 'D3': 42804},
{'D1': 1438064343, 'D2': 1816, 'D3': 30078}],
'C3': 3121424431,
'C4': 18349,
'C5': -115,
'C6': 237,
'C7': -0.869512140750885},
'B2': {'E1': {'F1': 19423,
'F2': [15911, 11058, 30558],
'F3': 560405524909556714,
'F4': 464831368,
'F5': 517709119},
'E2': 21616,
'E3': {'G1': [68, -108, 43, -89],
'G2': 40934,
'G3': 48426,
'G4': 1886023463},
'E4': 55610},
'B3': -0.9789525270462036,
'B4': -45,
'B5': -121,
'B6': -3317188644154863841,
'B7': [18, 30, 121, -116]},
'A3': -0.8925745487213135,
'A4': 1206483091}
```

**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.trash()	0
o.draw()	RuntimeError
o.carve()	1
o.carve()	2
o.carve()	RuntimeError
o.trash()	4
o.trash()	5
o.draw()	7
o.trash()	0
o.carve()	1
o.draw()	3
o.trash()	9
o.draw()	8
o.carve()	11
o.carve()	RuntimeError
o.trash()	6
o.trash()	9
o.carve()	10
o.trash()	RuntimeError
o.carve()	1

## 2. Пример использования класса C32:

o = C32()	
o.trash()	0
o.carve()	1
o.draw()	3
o.trash()	9
o.draw()	8
o.carve()	11
o.trash()	6
o.carve()	10
o.carve()	1
o.carve()	2
o.trash()	4
o.trash()	5
o.draw()	7

## Вариант №34

**Задача 3.1.** Реализовать разбор двоичного формата данных (в духе формата WAD игры Doom или графического формата PNG). Данные начинаются с сигнатуры 0x4e 0x41 0x58 0x45, 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 7
	3	int16
	4	Размер (uint16) и адрес (uint16) массива структур C
	5	Структура D
	6	Размер (uint16) и адрес (uint16) массива float
	7	uint32

Структура B:	1	uint32
	2	Массив char, размер 7

Структура C:	1	uint16
	2	int32
	3	double

Структура D:	1	uint32
	2	int8

Примеры разбора двоичного формата с помощью функции f31:

1. Двоичные данные:

```
(b'NAXEr\xd6\xb3\xc2\xac\xdbZ\x00\x00\x00;\x00\x00\x00F\x00\x00\x00Q'
b'\x00\x00\x00\\\x00\x00\x00g\x00\x00\x00r\x00\x00\x00}\x12B\x00\x03'
b'\x00\x88\x8d\x8d\xf2f\x9f\x00\x02\x00\xb2\xae\x9c\x93\xb6\xc1$\xd3Cpjmwm'
b'at#\x9fW\xd3kvsfeub\xcf\xb3sQbhzqfaq6f9nvywtwagj\xc6\xdcnqzlpkbf>~\xf7xcs'
b'fqtrk\x16\xe8\x16\xbacehbqai\xf3\x0e\x97\x0b\xeb\x1e?\x8c\xea\xe9v\xb4'
b'\xac\x00\xd4\xb1vmt\xaf\xbf\xdd'\xbfCm\xbe\x94\x1dq~1\xc4\xea?\xe4}uJV'
b'\xee\xb4?:\xff\xe4\xbf,Z\x15')
```

Результат разбора:

```
{ 'A1': 8274999014038494042,
  'A2': [{ 'B1': 3240416067, 'B2': 'pjwmmat' },
        { 'B1': 597645267, 'B2': 'kvsfeub' },
        { 'B1': 3484644177, 'B2': 'bhzqfaq' },
        { 'B1': 912669038, 'B2': 'vywtwag' },
        { 'B1': 1791417454, 'B2': 'qzlpkbf' },
        { 'B1': 1048508280, 'B2': 'csfqtrk' },
        { 'B1': 384308922, 'B2': 'cehbqai' } ],
```



```
'A3': 4674,
'A4': [{'C1': 62222, 'C2': -1760826594, 'C3': 0.014119934020149216},
        {'C1': 54449, 'C2': 1986884783, 'C3': -0.45902997574646887},
        {'C1': 7537, 'C2': 2117190890, 'C3': 0.6403147174782675}],
'A5': {'D1': 2379805286, 'D2': -97},
'A6': [0.7304670810699463, -0.6732495427131653],
'A7': 2929497014}
```

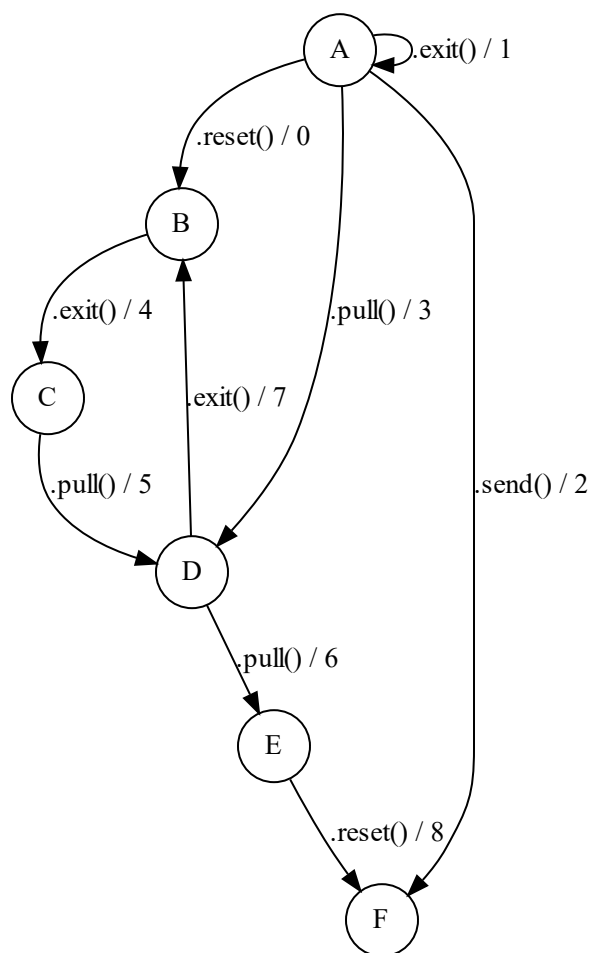
## 2. Двоичные данные:

```
(b'NAXE\x94\xe9\xdf\tEy_\xaf\x00\x00\x00;\x00\x00\x00F\x00\x00\x00Q'
b'\x00\x00\x00\\\x00\x00\x00g\x00\x00\x00r\x00\x00\x00}\xd5\xf4\x00\x02'
b'\x00\x88\xf0\xd7\x1d7\x00\x03\x00\xa4\x10\xb9\x8b&_n_\xdduwrusns\xf0v'
b'\x82\xcbfoklkxi\x80\xf0\xce4eismvai\xebdE\x9fvrcihnr\xddVX\x94sujrwoh\xa9T'
b'\xff\xbatdvixvhU.\xa6\x16eymeuax\xf6\x19\xf4J\xf7\x15?\xd7T<%t\xe6 \xc4!'
b'\xfdX\x19r?\xc9G\n\x05\x84\xb7x<8F\xfe\xbf\xc6\xb3\xbf\x06\xa6\xac')
```

## Результат разбора:

```
{'A1': 10730352818100199343,
'A2': [{'B1': 1601069021, 'B2': 'uwrusns'},
        {'B1': 4034298571, 'B2': 'foklkxi'},
        {'B1': 2163265076, 'B2': 'eismvai'},
        {'B1': 3949217183, 'B2': 'vrcihnr'},
        {'B1': 3713423508, 'B2': 'sujrwoh'},
        {'B1': 2840919994, 'B2': 'tdvixvh'},
        {'B1': 1429120534, 'B2': 'eymeuax'}],
'A3': -10764,
'A4': [{'C1': 63001, 'C2': -196413675, 'C3': 0.36451629312395717},
        {'C1': 50209, 'C2': -44557966, 'C3': 0.19748044270948717}],
'A5': {'D1': 4029404957, 'D2': 55},
'A6': [0.011247394606471062, -0.9717819094657898, -0.5259807109832764],
'A7': 280595238}
```

**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.exit()</code>	1
<code>o.pull()</code>	3
<code>o.exit()</code>	7
<code>o.exit()</code>	4
<code>o.pull()</code>	5
<code>o.exit()</code>	7
<code>o.exit()</code>	4
<code>o.pull()</code>	5
<code>o.pull()</code>	6
<code>o.pull()</code>	RuntimeError
<code>o.reset()</code>	8

2. Пример использования класса C32:

<code>o = C32()</code>	
<code>o.exit()</code>	1
<code>o.pull()</code>	3
<code>o.exit()</code>	7
<code>o.pull()</code>	RuntimeError
<code>o.exit()</code>	4
<code>o.pull()</code>	5
<code>o.pull()</code>	6
<code>o.exit()</code>	RuntimeError
<code>o.reset()</code>	8

## Вариант №35

**Задача 3.1.** Реализовать разбор двоичного формата данных (в духе формата WAD игры Doom или графического формата PNG). Данные начинаются с сигнатуры 0x49 0x4d 0x54 0x45 0x4c, 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	int16
	2	Адрес (uint16) структуры B
	3	int64
	4	Адрес (uint16) структуры E
	5	uint64
	6	int16
	7	float

Структура B:	1	uint8
	2	int16
	3	uint64
	4	uint8
	5	double
	6	int8
	7	Адрес (uint32) структуры C

Структура C:	1	Адрес (uint16) структуры D
	2	uint64
	3	float
	4	int8
	5	float

Структура D:	1	uint16
	2	Массив int8, размер 5
	3	uint8
	4	uint16

Структура E:	1	int16
	2	uint32
	3	uint8
	4	Массив адресов (uint16) структур F, размер 2
	5	Размер (uint32) и адрес (uint16) массива int8

Структура F:	1	uint16
	2	int32
	3	int32
	4	Размер (uint32) и адрес (uint32) массива uint64

Примеры разбора двоичного формата с помощью функции f31:

## 1. Двоичные данные:

```
(b'IMTELn\xf6>\x00\xa8\x97\xb2\n\xb0Xb\xbb\x9e\x00\x15\xe9"\xa4D\x12\xa8k\xaf'  
b'\x03\xad\x95\xce\xbe\xb3l\xe3\t\t\xcf\xe0\x11t\xde!\x00\xe7/\xf3'  
b'\x8f\x90\xe3\x98v\x16\xb8\x82\xbe\x0c\xaaaf,\xbd\x81\x03\t\xf2w\x14Kp[M'  
b"'P\n\x16]\xb8\xc9\x1f\xe6?\x10+\x00\x00\x00\xb7\x95z\xbc\xbf\xcbc\x99\x18"  
b'\xcda\x02\xb1\x81\x04\xe8\xab2\x1dc\x00\t\x9d!\n\xf0\x02\x00\x00'  
b"\x00W\x00\x00\x00\xd9\x1a\x13\xbf\xc5\xb6?\x9e\x15\xe7\x1c'\xa4\xa2\xb0"  
b'\xd6,2\x14#\xbe]\x08_6\xf5\x02\x00\x00\x00y\x00\x00\x00\xcb\xf3H\n\x81'  
b'\xcc\xc9\xf2\xd6\xd6g\x00\x89\x00\x03\x00\x00\x00\x9b\x00')
```

## Результат разбора:

```
{ 'A1': -2450,  
  'A2': { 'B1': 129,  
          'B2': 2307,  
          'B3': 2832020278540007410,  
          'B4': 80,  
          'B5': 0.691380283541265,  
          'B6': 16,  
          'B7': { 'C1': { 'D1': 27827,  
                        'D2': [-29, 9, 116, -49, -32],  
                        'D3': 17,  
                        'D4': 56948},  
                  'C2': 8545830502965391335,  
                  'C3': -0.25531071424484253,  
                  'C4': 12,  
                  'C5': -0.042090095579624176}},  
  'A3': -4944291927782418520,  
  'A4': { 'E1': -32502,  
          'E2': 3606235596,  
          'E3': 214,  
          'E4': [{ 'F1': 12971,  
                  'F2': 151020317,  
                  'F3': -267771491,  
                  'F4': [11052901934858671543, 16718630313955282200]},  
                { 'F1': 12844,  
                  'F2': 1572741908,  
                  'F3': -180986104,  
                  'F4': [11403033741955898073, 15470043545932195605]}]},  
  'E5': [-53, -13, 72]},  
  'A5': 7757470444166506773,  
  'A6': 943,  
  'A7': -0.4034856855869293}
```

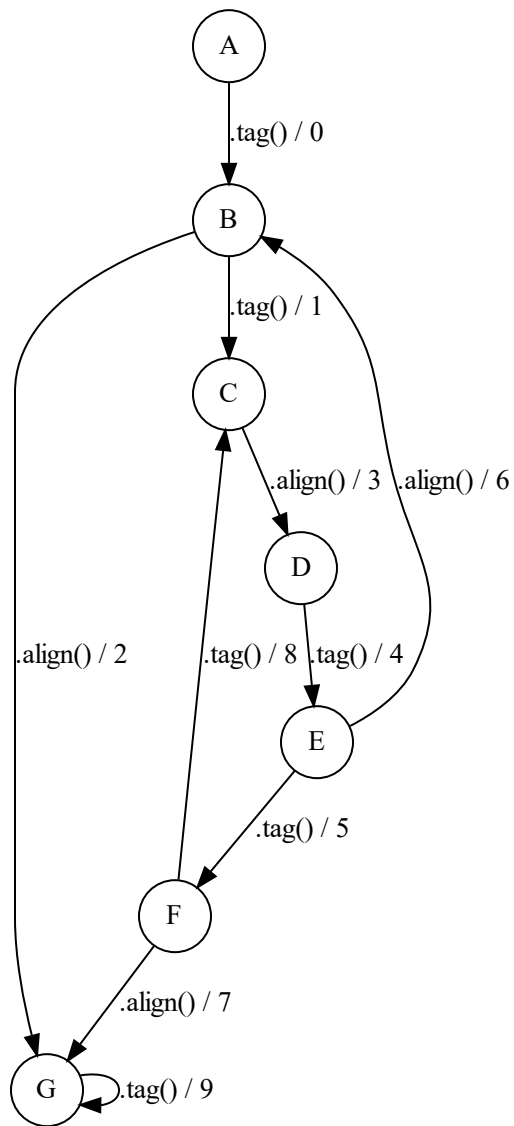
## 2. Двоичные данные:

```
(b'IMTEL3p>\x00\xf7\x15\x02\xa4\xef\xdd\xaf\xa5\x9f\x00\xd4{<\xf0\x86'
b"\x1d \xc0\xf4\x90\xa5\t` \xbf\\\x9dk\x92\x1e'\xefKE\xb2!\x00\xb8\x16\xaf"
b'\x1b\xde\xac\x02sM\xd6,?\xaa\x9c\xaf\xd8\xbe\x1d^\x9a9\xa6v\xc7\xbfU\xde'
b'g\r@$I}\x07\xb0\xd4\xbf\x88+\x00\x00\x00\xa8U\x93|\xb121\xf4\xaa'
b'\xb3\x10h\x817\xff\xa5\xccg`\x01\xb0\x7f\xe2?\x8b\xa1\x02\x00\x00'
b'\x00W\x00\x00\x00\x9cj\xbbKPM\x9f\xd6}\x831\x9c\xa7\x93\x93L\x8eP\x81\xca]`~'
b'\x06P\xe9\x02\x00\x00\x00y\x00\x00\x00Y+\xd3\xb2\xa8r\xe5\x04\xb1'
b'\x1d\x81g\x00\x89\x00\x04\x00\x00\x00\x9b\x00')
```

Результат разбора:

```
{'A1': 28723,
'A2': {'B1': 29,
'B2': -26018,
'B3': 7484513912909899321,
'B4': 13,
'B5': -0.32324397311146313,
'B6': -120,
'B7': {'C1': {'D1': 40284,
'D2': [107, -110, 30, 39, -17],
'D3': 75,
'D4': 45637},
'C2': 8287376334262310584,
'C3': 0.6751449704170227,
'C4': -86,
'C5': -0.42321479320526123}},
'A3': -6507738915208555017,
'A4': {'E1': 29352,
'E2': 498140389,
'E3': 129,
'E4': [{'F1': 26572,
'F2': 2142241120,
'F3': -1584709662,
'F4': [17612507980619797928, 11961340164255560618]},
{'F1': 20622,
'F2': 1616759425,
'F3': -380631426,
'F4': [15465164652676868764, 5517916316548301693]}],
'E5': [89, 43, -45, -78]},
'A5': 13844097719930223572,
'A6': -28428,
'A7': -0.8751471638679504}
```

**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.tag()          0
o.tag()          1
o.align()        3
o.tag()          4
o.tag()          5
o.tag()          8
o.align()        3
o.tag()          4
o.align()        6
o.align()        2
o.tag()          9
o.tag()          9
o.align()        RuntimeError
o.tag()          9

```

## 2. Пример использования класса C32:

```

o = C32()
o.tag()          0
o.tag()          1
o.align()        3
o.tag()          4
o.tag()          5
o.tag()          8
o.align()        3
o.tag()          4
o.align()        6
o.align()        2
o.align()        RuntimeError
o.tag()          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 0x5a 0x53 0x57, 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	uint32
	3	int8
	4	Массив int64, размер 5
	5	uint32
	6	int32
	7	double

Структура B:	1	Массив структур C, размер 5
	2	Структура D
	3	uint32

Структура C:	1	int32
	2	uint32
	3	int16

Структура D:	1	int64
	2	uint16
	3	Размер (uint32) и адрес (uint32) массива uint16
	4	Массив uint32, размер 4
	5	uint8
	6	int32
	7	int8
	8	int8

Примеры разбора двоичного формата с помощью функции f31:

### 1. Двоичные данные:

```
(b'ZSW\x00\x00\x00Nk\xbbjT4\xfd\xa5K?\x17\xbd\xe0\x9a\xd3Jz\xd1\x1bsb\xc8'
b'\xe3fEKgW-\xed\xd6\x88<\xfcgb\x05j\xb0\xe9\xec\nY\xdc\xd61K\xb6h=\xe6%\x88'
b'\xbf\xae/\x12\xec9\xde@rvW\x8e\xea4\x9fi\x06\xec\x08W\xbdh` v1\x01\x96'
b'z\x1a\x8e}\xa6w\xe3\xbe\xf5\x9b\xc1[\xd7\xa3\xea\xab\xee\xe4\xd5\xbb'
b'\x85\x18\x0b\x91\xb4\xd9\xe6\x9d>J\xb6\x9b\x14\x90Pu\x83\xb8d\x8a'
b'\x07\xa4\x8c\xb97\xfe\x002D\x00\x00\x00\x05\x00\x00\x00DoW\xc2\xc9z\x13'
b'\x80\xf4\xca\x12\xe1\xf3\xb8v\xeb\x8fF\xe2J3UJq\xa7\xa4\x955')
```

Результат разбора:

```
{'A1': {'B1': [{'C1': 139967848, 'C2': 1612740204, 'C3': 406},
               {'C1': 2048560765, 'C2': 2792874942, 'C3': -2661},
               {'C1': -1050945629, 'C2': 3937136356, 'C3': -10821},
               {'C1': -2062021743, 'C2': 3034179229, 'C3': 15946},
               {'C1': -1231350640, 'C2': 1349878712, 'C3': 25738}],
        'B2': {'D1': 550719781582274610,
                'D2': 49732,
                'D3': [29302, 22414, 59956, 40809, 1772],
                'D4': [1868022473, 2048098548, 3390235123, 3094801295],
                'D5': 70,
                'D6': -497208491,
                'D7': 93,
                'D8': 113},
        'B3': 2812581173},
'A2': 1807444564,
'A3': 52,
'A4': [-169646676603182950,
        -3221627544902868280,
        -2060883589316772115,
        -2988071297998462923,
        7687901763916651734],
'A5': 827045480,
'A6': 1038493064,
'A7': -0.058952895485103074}
```

## 2. Двоичные данные:

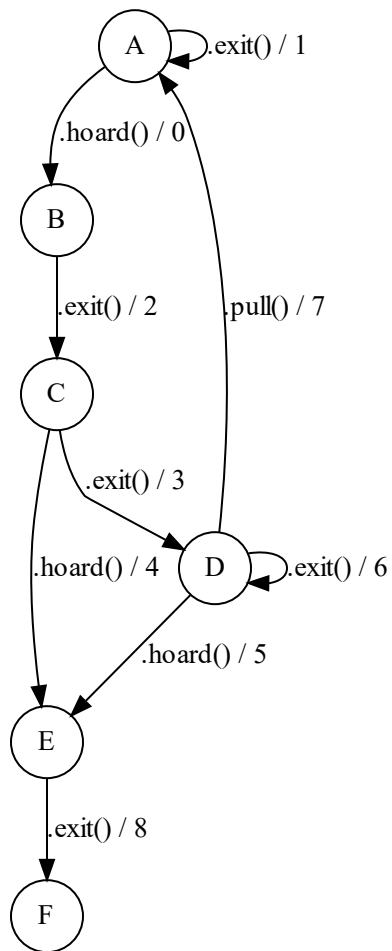
```
(b'ZSW\x00\x00\x00J\xea\xd1\x04zZ\xf3\x8d\xab"2\xaf\xbe\xfe\x89\x82E\xd6'
b"?e\x81\x96\xce'\xbe\xe49\xc1/\x89\xa06\x0f >\xb1H\xf0\xb5\xeeYh"
b'\xf1\x80\xa6\x94\xce\x81\x1d,\xc5qL\xeb?\xeb\xb9A\x88\xb0C\xd2\x8a\xe3T\x95'
b"\x1b\x97\x8d\xad\xd42\x92!r4\x03Y'\xf9!\x1f!\xc3a\xc3'\xfa\x84P\x962\x90u"
b'\xabC{2\xc8\n[\x8d\x14U\xec\xe2\xe0\xb1$< a\xaf1q\xa0\x8ab\xc2\x14\xe3'
b'\xbbj\x8a\x0f\x82X\x00\x00\x00\x03\x00\x00\x00D*4M\x99\xfe)cqa4'
b'\xa1\xf4\xc3\x7f?\xc7\x17\xa1\x18\xb0f$\xfd\xd9%\xbbs')
```

## Результат разбора:

```
{'A1': {'B1': [{'C1': -1917987790, 'C2': 2451665460, 'C3': 857},
               {'C1': 670638367, 'C2': 566452675, 'C3': 10234},
               {'C1': -2075093454, 'C2': 2423630659, 'C3': 31538},
               {'C1': -933466533, 'C2': 2366920172, 'C3': -7456},
               {'C1': -1323025376, 'C2': 1638871409, 'C3': -24438}],
        'B2': {'D1': 7116273329533258255,
                'D2': 33368,
                'D3': [35555, 21653, 7063],
                'D4': [708070809, 4264125297, 1630839284, 3279896519],
```

```
        'D5': 23,  
        'D6': -1592217498,  
        'D7': 36,  
        'D8': -3},  
    'B3': 3643128691},  
    'A2': 3939566714,  
    'A3': 90,  
    'A4': [-896872587409244418,  
           -8538185157052104298,  
           -3591692290373898359,  
           -6902312747733530384,  
           -5337230201124247916],  
    'A5': 3464568108,  
    'A6': -982430485,  
    'A7': 0.8663642568300196}
```

**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.hoard()           0
o.exit()            2
o.exit()            3
o.pull()            7
o.exit()            1
o.hoard()           0
o.exit()            2
o.exit()            3
o.exit()            6
o.exit()            6
o.hoard()           5
o.exit()            8

```

2. Пример использования класса C32:

o = C32()	
o.hoard()	0
o.exit()	2
o.exit()	3
o.exit()	6
o.exit()	6
o.pull()	7
o.exit()	1
o.hoard()	0
o.pull()	RuntimeError
o.exit()	2
o.hoard()	4
o.exit()	8

## Вариант №37

**Задача 3.1.** Реализовать разбор двоичного формата данных (в духе формата WAD игры Doom или графического формата PNG). Данные начинаются с сигнатуры 0x56 0x48 0x51 0x12, 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	int64
	2	Структура B
	3	int64
	4	Массив int32, размер 3

Структура B:	1	Адрес (uint32) структуры C
	2	uint8
	3	uint32
	4	int32
	5	uint16
	6	uint8

Структура C:	1	Массив структур D, размер 2
	2	Размер (uint32) и адрес (uint32) массива uint32

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

Примеры разбора двоичного формата с помощью функции f31:

1. Двоичные данные:

```
(b'VHQ\x12K\xc4RJ\xad\x0f\x18\xd6\x00\x00\x00`Umz\x06\xe3\xe5?\xdb'
b'\x88\x19\xec\xfd\x82m\xd9?\x03\xdc\x1b\xd7\xa82\x99\x03\xa9\x04z\xaa'
b'\xf7\x0fi8.5\xe4\xa7}!vP;[P@\xfa#z\xc6\x9b\xc1!\x96\xa\xdfH3s\xb9k\xfc'
b'=u\x83\t_\xbd#\xef\x0e\xb3\x81\x86(\xb5\\\xfex\xab\xfb\x0\x00\x00\x00\x02'
b'\x00\x00\x000\xbf g\x04\x0ej=\xa9]B\xc4\x06\x03\x81\x10\x00\x02\x008\x00\x00'
b'\x00\x06\x00\x00\x00<?n\xd3t\xb1\xb0\xde\xed\xc7V\xf72\x01\x9f\x00\x02\x00T'
b'\x00\x00\x00\x02\x00\x00\x00X')
```

Результат разбора:

```
{ 'A1': 5459579128964389078,
  'A2': { 'B1': { 'C1': [ { 'D1': [775283879, 2099344976],
                             'D2': -0.9060677289962769,
```

```

'D3': 1782425949,
'D4': 1120142851,
'D5': -32496,
'D6': [15195, 20544]},
{'D1': [4196629190,
        2613125526,
        2329888819,
        1941531644,
        1031111433,
        1606231023],
'D2': 0.9329140186309814,
'D3': -1313808659,
'D4': -950601934,
'D5': 415,
'D6': [3763, -32378]}},
'C2': [682974462, 2024537072]},
'B2': 85,
'B3': 1836713699,
'B4': -448799864,
'B5': 6636,
'B6': 253},
'A3': -9048337211646534697,
'A4': [-1473079037, -1459324246, -149984968]}

```

## 2. Двоичные данные:

```

(b'VHQ\x12@\xdd\xea\x9f\xb2&\x8f \x00\x00\x00z\x10\xd6\xed$B\xda\xbfp'
 b'\xaaaf\xbd\x18\xa2V\x00\xd5\x18\x1c\x94\x82\xc9\x02\xd4\xb3z\x1a\x12\xe1'
 b')\xf4\xbb;.\x8ahD\x86F\x0f0\xca\xda\xe9\xbad\x1c%3I\xa2\xc7&\xb0v\xd2\x9b\x9a'
 b'E\xb1kT3\xc2S\x13[\xfa\x1a\xe8t\xcc3d\x88\xd4\xf2mV\xe0j3\xb2\xfb<\xfe'
 b'\xb0\xb5\x9c\xa7pm9a\xe2\x03Hw\x9c\x80\x1b\xb9\x11\xdf\xed\xa5'
 b'\x11\x9a\x00\x00\x00\x07\x00\x00\x000\xbd\xe2\x8f\xdf\xab\xcd\x7f\x1a\x92A'
 b'M9\xc1A\x00\x04\x00L\x00\x00\x00\x06\x00\x00\x00T\xbf1\x01\x83\x7f\x1f\x1f\xaf'
 b'xFAwas\x00\x03\x001\x00\x00\x00\x02\x00\x00\x00r')

```

## Результат разбора:

```

{'A1': 4674149959890013984,
'A2': {'B1': {'C1': [{'D1': [780822596,
                              2252796106,
                              3672750692,
                              472200009,
                              2730960560,
                              1993513882,
                              1169255252],
'D2': -0.1106259748339653,

```

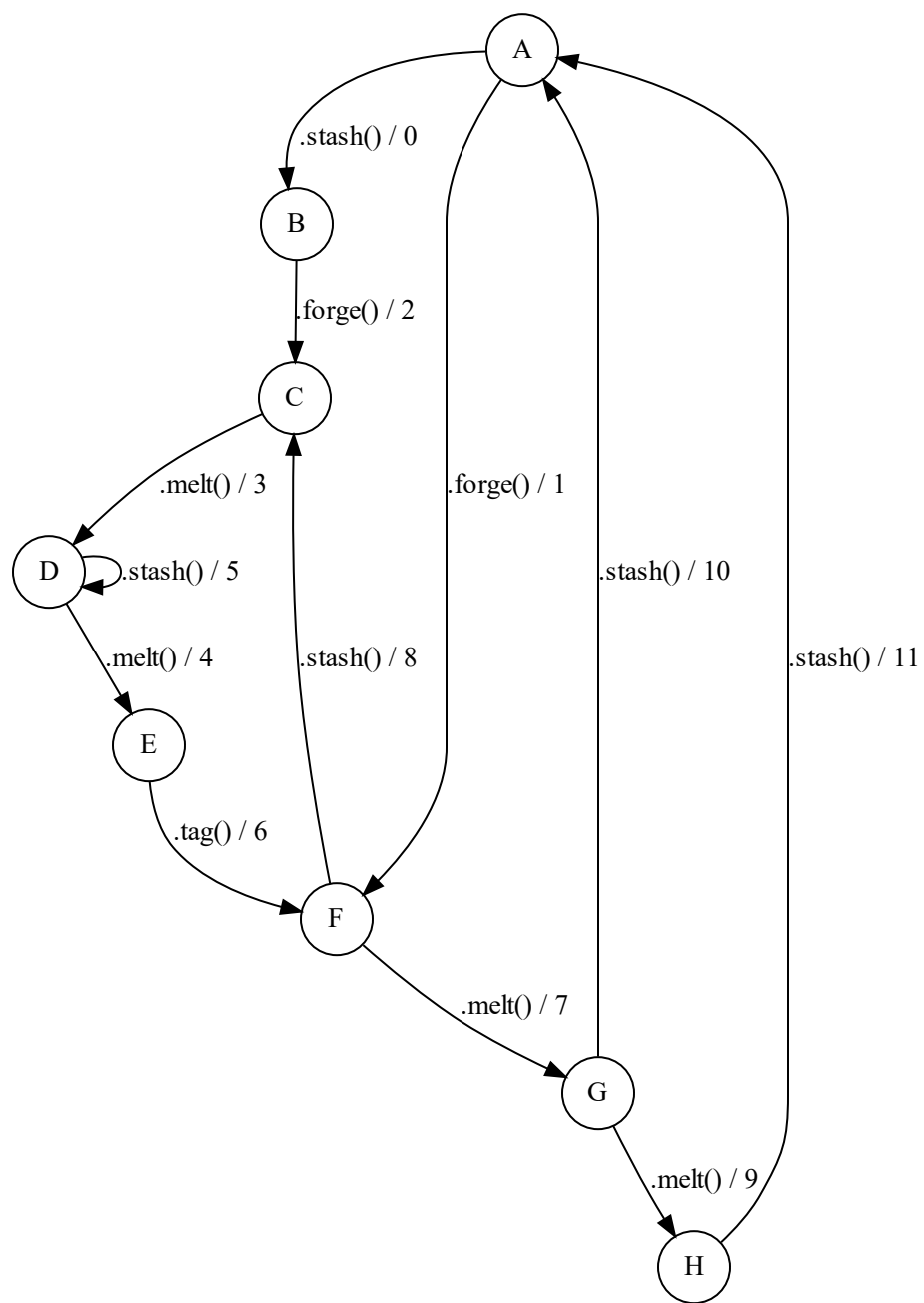
```

'D3': -1412595942,
'D4': -1841214151,
'D5': -16063,
'D6': [13250, 21267, 23546, 6888]},
{'D1': [1959539556,
        2295657069,
        1457547827,
        3002809598,
        2964692135,
        1886206305],
'D2': -0.9218980669975281,
'D3': 2146564710,
'D4': 2017870167,
'D5': 24947,
'D6': [-7677, 18551, -25472]}},
'C2': [465113567, 3987018138]},
'B2': 16,
'B3': 3605865538,
'B4': -624987990,
'B5': 26301,
'B6': 24},
'A3': -6749206076335549310,
'A4': [-922561357, 2048529121, 703904571]}

```

**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()	1
o.stash()	8
o.melt()	3
o.stash()	5
o.stash()	5
o.melt()	4
o.tag()	6
o.melt()	7
o.forge()	RuntimeError
o.melt()	9
o.stash()	11
o.melt()	RuntimeError
o.forge()	1
o.tag()	RuntimeError
o.melt()	7
o.stash()	10
o.stash()	0
o.forge()	2
o.melt()	3

## 2. Пример использования класса C32:

o = C32()	
o.forge()	1
o.stash()	8
o.stash()	RuntimeError
o.melt()	3
o.stash()	5
o.stash()	5
o.melt()	4
o.tag()	6
o.melt()	7
o.melt()	9
o.stash()	11
o.forge()	1
o.melt()	7
o.stash()	10
o.melt()	RuntimeError
o.stash()	0
o.forge()	2

## Вариант №38

**Задача 3.1.** Реализовать разбор двоичного формата данных (в духе формата WAD игры Doom или графического формата PNG). Данные начинаются с сигнатуры 0x56 0x4c 0x53 0xb, 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	Адрес (uint16) структуры D
	3	Размер (uint16) и адрес (uint32) массива int8

Структура В:	1	int32
	2	float
	3	Размер (uint32) и адрес (uint32) массива адресов (uint32) структур С

Структура С:	1	uint8
	2	float
	3	int64
	4	Размер (uint16) и адрес (uint16) массива uint8

Структура D:	1	int64
	2	uint32
	3	Структура E
	4	int32

Структура E:	1	int32
	2	uint16
	3	uint64
	4	Размер (uint32) и адрес (uint16) массива uint32
	5	uint64
	6	float
	7	Массив int8, размер 3

Примеры разбора двоичного формата с помощью функции f31:

1. Двоичные данные:

```
(b'VLS\x0b\xe6\x9b\xadWpmD\xbe\x02\x00\x00\x00E\x00\x00\x00Y\x00\x02\x00'
b'\x8c\x00\x00\x00\x9d\x14\x91\xc9m7\x81\x03\xbesSS\x17\xa2\xcd\xcb'
b'd\x04\x00\x1c\x00\x9dX\n\xe8\x0c;\x9a>R\xc3\x03[b#E\xb7\x03\x001'
b'\x00 \x00\x00\x004\x00\x00\x00s\xfb\xd3)\x15rM&\xbe\n)u\x0eumb\xeb=\xf4'
b'\x05\xea\x17\x80\xa3\xf9s\x7fc}\xac"\xe5\xff\x8e\xd0\xd7]#\x03\x00\x00\x00M'
b'\x00\xa3\xb8\xf2eE\x10\xd8hu\xdd\x97\xbeT)5\xd0\xde\xa8\xbd\xfb1')
```

Результат разбора:

```
{'A1': {'B1': 1470995430,
        'B2': -0.19182372093200684,
        'B3': [{'C1': 109,
                  'C2': -0.1284226030111313,
                  'C3': 7263124920124789619,
                  'C4': [157, 20, 145, 201]},
                {'C1': 232,
                  'C2': 0.3012317419052124,
                  'C3': -5240743686034963630,
                  'C4': [157, 88, 10]}]},
'A2': {'D1': 429035945684989198,
        'D2': 2743080938,
        'D3': {'E1': 1669297145,
                  'E2': 44157,
                  'E3': 2548430254913873186,
                  'E4': [701758323, 642609685, 1965623998],
                  'E5': 7554806265163200675,
                  'E6': -0.2966114580631256,
                  'E7': [84, 41, 53]},
        'D4': 1336467152},
'A3': [-67, -15]}
```

## 2. Двоичные данные:

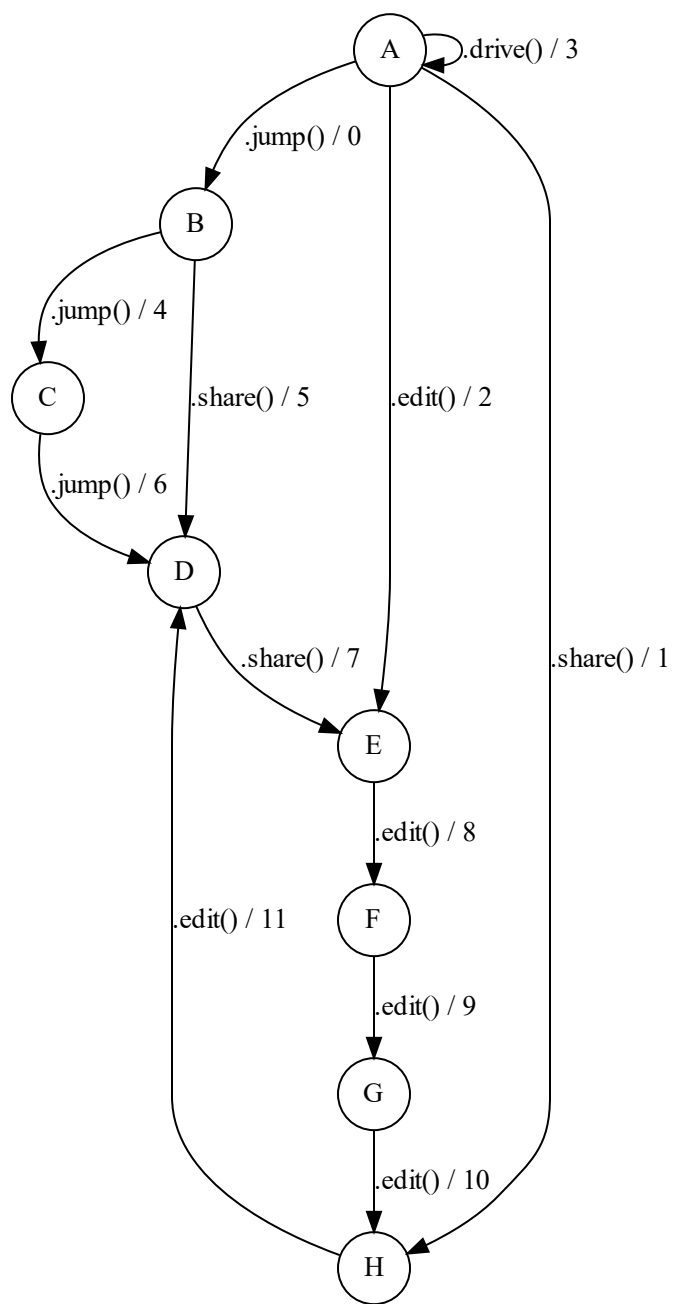
```
(b'VLS\x0b\xa9\xf1\xc3YF\x1d?\x02\x00\x00\x00D\x00\x00\x00X\x00\x02\x00'
 b'\x8b\x00\x00\x00\x81\x14\x0b-%\xa56?5(\xb3\x1e\x0b\x1e\xf5k\x03\x00\x1c\x00'
 b'\xec\x99\xb7+\xfd\n)?\xa8L\n\x8e8vo\x12\x03\x000\x00\x1f\x00\x00\x00'
 b'3\x00\x00\x00\xee"\x12~\x94\xb2\xf6V&(\CS\x1d\xd6\x0bs\L\xbe]\xbc\x97\xba'
 b'I\xf1\xc1F\x93\xbc{\xe7\x87\xf6"\x8a\xa8\x1b\x03\x00\x00\x00L\x00\x99e\x00W'
 b'\x13=\x83\x0e\x03\x14i? c\x9d\xa4t1f\xab\xfb')
```

### Результат разбора:

```
{'A1': {'B1': 869265833,
        'B2': 0.61435467004776,
        'B3': [{'C1': 45,
                  'C2': 0.7134574055671692,
                  'C3': 7779156964460931125,
                  'C4': [129, 20, 11]},
                {'C1': 43,
                  'C2': 0.6603239178657532,
                  'C3': 1328410400370871464,
                  'C4': [236, 153, 183]}]},
'A2': {'D1': -4734307459083657901,
        'D2': 3130506333,
        'D3': {'E1': 1187115337,
```

```
'E2': 48275,  
'E3': 1992994717881067387,  
'E4': [2115117806, 1459008148, 1130113062],  
'E5': 1045746691753403801,  
'E6': 0.9104616045951843,  
'E7': [32, 99, -99]},  
'D4': 1714517156},  
'A3': [-85, -5]}
```

**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()          3
o.jump()           0
o.jump()           4
o.jump()           6
o.jump()           RuntimeError
o.share()           7
o.edit()            8
o.edit()            9
o.edit()           10
o.edit()           11
o.share()           7
o.edit()            8
o.edit()            9
o.edit()           10

```

## 2. Пример использования класса C32:

```

o = C32()
o.drive()          3
o.drive()          3
o.jump()           0
o.jump()           4
o.jump()           6
o.share()           7
o.jump()           RuntimeError
o.edit()            8
o.edit()            9
o.edit()           10
o.edit()           11
o.share()           7
o.edit()            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 0x58 0x5a 0x50, 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	uint8
	2	int64
	3	Структура В
	4	Адрес (uint32) структуры F
	5	float

Структура В:	1	Адрес (uint16) структуры С
	2	Размер (uint16) и адрес (uint16) массива int8
	3	uint8
	4	int64
	5	uint32
	6	uint16

Структура С:	1	Массив структур D, размер 2
	2	Структура Е
	3	Размер (uint32) и адрес (uint32) массива uint64

Структура D:	1	Массив uint32, размер 6
	2	int16

Структура Е:	1	Массив int8, размер 7
	2	double

Структура F:	1	int8
	2	uint8

Примеры разбора двоичного формата с помощью функции f31:

1. Двоичные данные:

```
(b'XZP\xca\x13~\\\x07!\xd5\xd7\xc19\x00\x04\x00\x84\x00\xcd\x86\x12C_";\xcb\t'
b'^V\xfb\xe6\xdf\x88\x00\x00\x00_U\xed>V1\xd0\x7f\xb7\x12>\x95g\xf3 '
b'\x150\xf5\x8b\x82\xa1\xec}\xb4\xc2i\x18\xfcq\xfa\xed\xf2C6\xf3'
b"\x82\xcf\x89\xae\x1b\xf5W \xcc\x8a\xe5\x7f\x82'\xd6\xa8\xdc-Z\xd6"
b'\x0e\x96\xd2\x04\xdb\xc3$m-\x9f\xc5\xae\xb2|\xa2\x01\xa7k5a)>\xae\x93'
b'd\xcb\xdd\xaf\xe0\xee?\x02\x00\x00\x00)\x00\x00\x00\x9f\xe8\xb81\xdf\xd0')
```

Результат разбора:



```
{'A1': 202,
'A2': -4478876966562529773,
'A3': {'B1': {'C1': [{'D1': [3028151457,
                              4229458370,
                              4075682417,
                              2196977219,
                              464423375,
                              3424671733]},
'D2': -6774},
{'D1': [3592913535,
        1512955048,
        3533049558,
        616815364,
        3315543405,
        2726081198]},
'D2': -22783}],
'C2': {'E1': [107, 53, 97, 41, 62, -82, -109],
'E2': 0.947250693787606},
'C3': [10754053539539988822, 9406881833555129191]},
'B2': [-97, -24, -72, 108],
'B3': 205,
'B4': -3802409028873416058,
'B5': 4216741385,
'B6': 57318},
'A4': {'F1': -33, 'F2': 208},
'A5': 0.4635419547557831}
```

## 2. Двоичные данные:

```
(b'XZP\xcd\xb4\xb8\xe2v\xcb&\xa8\x879\x00\x02\x00\x84\x00*\xbe4\x16n1v\xb6z\r'
b'\xd3\rw\xb9\xed\x86\x00\x00\x00\xb6a\xbd>\x9f&F\xcc\xaa,\xb3\xb9\x93\xa9a'
b'\xd9\xd2@g>/S\xf9\xe6s\xa8\x9c\xf8\xe5\x12)Rw\xcf\x860\x147D2\xecu\x1e'
b'\x00[:E?\xfb\xfbI:L\xbaK\x87\x83W09\xdb\xce\x9f\x12\x81\x03%\xd0\xb7\n('
b'G:P\x0f\x90\x1c\xfb1t\xc1\xcb\x193"\xe9\xbf\x02\x00\x00\x00)\x00\x00\x00'
b'\xbb#\xfc\xab')
```

### Результат разбора:

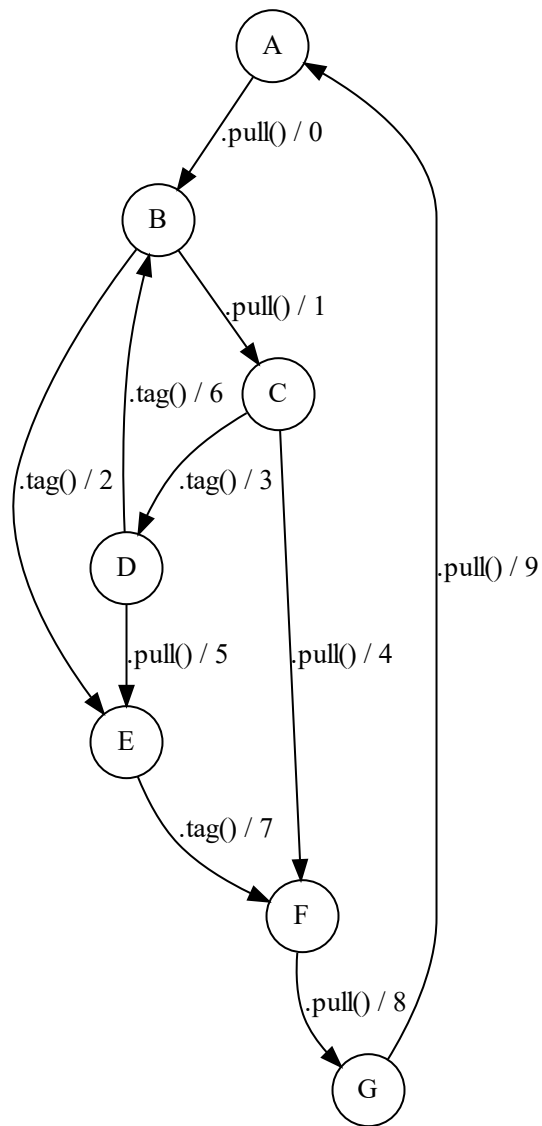
```
{'A1': 205,
'A2': -8671638427187103564,
'A3': {'B1': {'C1': [{'D1': [3875099439,
                              4171016307,
                              1378423525,
                              814141303,
                              843331348,
                              1996268]},
```

```

        'D2': 14939},
    {'D1': [4227546949,
            3125557833,
            1468237643,
            3470473551,
            58790559,
            179818533],
     'D2': 18216}],
    'C2': {'E1': [58, 80, 15, -112, 28, -5, 49],
           'E2': -0.7854247573687148},
    'C3': [13381088029931087519, 4496634027287095699]},
    'B2': [-69, 35],
    'B3': 42,
    'B4': 8842385126464042174,
    'B5': 1997394701,
    'B6': 60857},
    'A4': {'F1': -4, 'F2': 171},
    'A5': 0.3698861002922058}

```

**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.pull()	0
o.pull()	1
o.tag()	3
o.tag()	6
o.pull()	1
o.pull()	4
o.pull()	8
o.pull()	9
o.pull()	0
o.tag()	2
o.tag()	7
o.pull()	8

## 2. Пример использования класса C32:

o = C32()	
o.pull()	0
o.tag()	2
o.tag()	7
o.pull()	8
o.tag()	RuntimeError
o.pull()	9
o.tag()	RuntimeError
o.pull()	0
o.pull()	1
o.tag()	3
o.tag()	6
o.pull()	1
o.tag()	3
o.pull()	5

## Вариант №40

**Задача 3.1.** Реализовать разбор двоичного формата данных (в духе формата WAD игры Doom или графического формата PNG). Данные начинаются с сигнатуры 0xb6 0x47 0x45 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	Массив char, размер 5
	2	int64
	3	int64
	4	Размер (uint16) и адрес (uint16) массива адресов (uint32) структур B
	5	Структура C
	6	uint8
	7	uint32

Структура B:	1	uint32
	2	uint16
	3	uint16

Структура C:	1	uint8
	2	Адрес (uint32) структуры D
	3	int16
	4	Структура E

Структура D:	1	double
	2	uint16
	3	Массив float, размер 2

Структура E:	1	int32
	2	Размер (uint16) и адрес (uint32) массива int16
	3	double
	4	int16
	5	uint16
	6	int64
	7	int32
	8	Размер (uint16) и адрес (uint16) массива int64

Примеры разбора двоичного формата с помощью функции f31:

1. Двоичные данные:

```
(b'\xb6GEFtkpba-\xdf:\x87\x95\x05\xdd)\xa6$_\xc8\xa5\x04\xe2\x12\x04\x00o'
b'\x00\xb3\x7f\x00\x00\x00\xf0@\xf0e\x92]\x02\x00\x91\x00\x00\x00\xe0 '
b'\xd7\xd9\xa0\xc0\xc2\xbf\xedN\x85}\x07\x11\x00\xdfK]\x00\xaa\x9d'
b'\xf1\xdf\x03\x00\x95\x00\x0b\x04j\x069\x00\x18\xef\x7f\xebw\xbf\x02w'
b'\xe6\x139\xfc\x7fZ\x0b\xb1\x85J;\xaf/\xdb0}\xbc\x94\x19\xff\xdf\x150'
```

b'\x00\x00\x00W\x00\x00\x00\_\x00\x00\x00g\x00\x00\x00\xb8H\x06c\xa2'  
b'\xe7\xc8\xbf"/\xd3~\xea\xbe\xbc(\xa1\xbe8\xc1\xaa\x81S\xce\x97\xcf\xfe\xebm'  
b'ea\x9d\xd4h\x8c"&P\x12\xa3\xcem\xaf\xdaCr')

Результат разбора:

```
{'A1': 'tkpba',  
 'A2': 3016573465185148717,  
 'A3': 1360655147497104550,  
 'A4': [{'B1': 2146375680, 'B2': 30699, 'B3': 703},  
        {'B1': 957605495, 'B2': 32764, 'B3': 2906},  
        {'B1': 994739633, 'B2': 12207, 'B3': 56156},  
        {'B1': 2495380815, 'B2': 65305, 'B3': 5599}],  
 'A5': {'C1': 179,  
        'C2': {'D1': -0.1945689185636803,  
               'D2': 12066,  
               'D3': [-0.4579988420009613, -0.3147639036178589]}},  
        'C3': 16624,  
        'C4': {'E1': 1569875440,  
               'E2': [-16072, -32342],  
               'E3': -0.14650354992614378,  
               'E4': 28397,  
               'E5': 34126,  
               'E6': 26260593984997245,  
               'E7': -537813590,  
               'E8': [7308757249929236051,  
                      5775341558598770017,  
                      8233664990759789330]}},  
 'A6': 11,  
 'A7': 956721668}
```

## 2. Двоичные данные:

(b'\xb6GEF\xvkk1\t\x00\xdd?\xf602\xb9H)\xf5;\x84\x03F\x03\x00g\x00\x82s\x00'  
b'\x00\x00#\x9d\xea\xd1\x87<\x06\x00\x85\x00\x00\x00\xdc\xe5!\xf6\xd4P'  
b'\xef\xbf\xc6/\xb6\xa0^^\xf1\xfe\x80q\xa8\x8b\xb0\xd3K\x03\x00'  
b'\x91\x00\xc3\xf6\x94\xd6\x07\xb8\xab~\xd6\x8a-^\xee\xd8\xca\xdc\xba>'  
b'\xa7\xdf\xa1\xf2\x9c[\xf9\xfa\x05!\xf0\x00\x00W\x00\x00\x00\_  
b'\x00\x00\x00\xcc\xe6\xa7%=\xf0\xd2?\xbc\x97+\_m=\x81Z4?\xed\x08\xd3'  
b'\xed\x0f\$\x8c\xca\x89\x90\xdc\xe4\xadjE\x1a\xaf\x1a1F\x83\x08\x1f'  
b'\x9c\xf3\xb8\xeb\x85\xf3\x94\xcdw\xb6\xb3\x87\xd7')

Результат разбора:

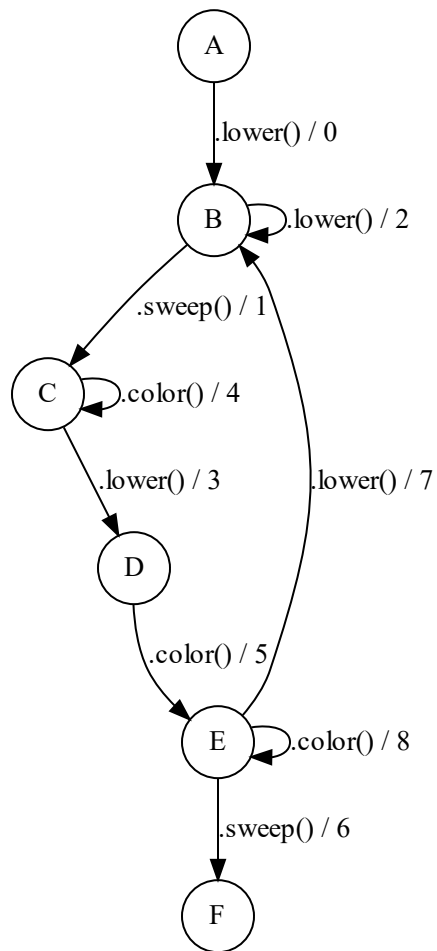
```
{'A1': 'xvkk1',  
 'A2': -5101961593665814519,
```

```

'A3': 5045021400637712712,
'A4': [{'B1': 3598625720, 'B2': 11658, 'B3': 61022},
        {'B1': 3135032024, 'B2': 42814, 'B3': 41439},
        {'B1': 4183530738, 'B2': 1530, 'B3': 65313}],
'A5': {'C1': 130,
        'C2': {'D1': 0.28218010594683274,
                'D2': 38844,
                'D3': [0.05795208737254143, 0.704505980014801]}},
        'C3': -25309,
        'C4': {'E1': 1015534058,
                'E2': [2285, -4653, 9231, -13684, -28535, -6948],
                'E3': -0.9786171729348854,
                'E4': 12230,
                'E5': 41142,
                'E6': -6309120341512724900,
                'E7': 1272164491,
                'E8': [5074460219502062253,
                        -8796734090702288765,
                        -2916164637423725325]}}},
'A6': 195,
'A7': 131503350}

```

**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.lower()           0
o.lower()           2
o.color()           RuntimeError
o.lower()           2
o.sweep()           1
o.color()           4
o.lower()           3
o.color()           5
o.color()           8
o.sweep()           6
  
```

2. Пример использования класса C32:



o = C32()	
o.sweep()	RuntimeError
o.lower()	0
o.color()	RuntimeError
o.sweep()	1
o.color()	4
o.lower()	3
o.color()	5
o.color()	8
o.lower()	7
o.color()	RuntimeError
o.sweep()	1
o.color()	4
o.lower()	3
o.color()	5
o.sweep()	6