

# Группа Н4

## Вариант №1

**Задача 3.1.** Реализовать разбор двоичного формата данных (в духе формата WAD игры Doom или графического формата PNG). Данные начинаются с сигнатуры 0x56 0x46 0x52 0x4e 0x59, за которой следует структура А. Порядок байт: от младшего к старшему. Адреса указаны в виде смещений от начала данных. В решении разрешено использовать модуль struct.

Структура А:	1	double
	2	uint64
	3	uint8
	4	double
	5	uint8
	6	uint64
	7	Адрес (uint16) структуры В

Структура В:	1	float
	2	int64
	3	Массив структур С, размер 2
	4	Массив uint32, размер 2
	5	Структура Е
	6	uint16
	7	float
	8	int16

Структура С:	1	int16
	2	Структура D
	3	Массив int8, размер 8

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

Структура Е:	1	double
	2	int32

Примеры разбора двоичного формата с помощью функции f31:

1. Двоичные данные:

```
(b'VFRNY\b6\b9b\b16\b06\Xe4\x96\xef?\xa1J\b4\b4%n,\Xe0\bba\b0n\bdbx0^'  
b'\xcc?\xa9<\xdd\xfa\b10\b92\xad\xdd\b9f1\b00n(\b909\b11\bdb\bba0\b6\bdb\b05'
```

```

b'\xbf\xfa7\xabL\xb78v\b4sk\x87\x9d\xceb\xe2\xa9(\xc8A\xa11\xb3\x89'
b'\xe6<\xdc\x8d\xa6\x04\xc0\xb9\x02\x00)\x00\x89\x80\xdc\x91\x85\xeb\xf7\x1d'
b'\x92T\xa9\x10#\xad.=\x98^L\xe1j\xe9L\xdc\x06\xfd\x0e\xaf\xeb\xcb\xa8'
b'do\xe3\x02\x00-\x00\x9b\xf9\xd6\xffj3\x1d\xd6\xba}\xee\x9c\xeb\xda\x86P]'
b'p\x80\xe98:{\xc1\xb7\bfb\xdd?\x8d\xcf\xf0\x1245 \xbc\x12>\x19y')

```

## Результат разбора:

```

{'A1': 0.9871692770874378,
 'A2': 16153407071695686305,
 'A3': 186,
 'A4': 0.2216244306135784,
 'A5': 169,
 'A6': 11519554264745303356,
 'A7': {'B1': -0.5228837728500366,
        'B2': -5443100739714205702,
        'B3': [{'C1': 27507,
                  'C2': {'D1': 1657707911,
                          'D2': 226,
                          'D3': 1103636649,
                          'D4': [12705, -30285, 15590, -29220, 1190, -17984],
                          'D5': [10350, 14736],
                          'D6': 2447147145},
                  'C3': [-123, -21, -9, 29, -110, 84, -87, 16]}],
        'C1': 22051,
        'C2': {'D1': 2554146477,
                  'D2': 94,
                  'D3': 3916095820,
                  'D4': [-9140, -762, -20722, -13333, 25768, -7313],
                  'D5': [56081, 20410],
                  'D6': 4292278683},
        'C3': [93, 51, 29, -42, -70, 125, -18, -100]}],
 'B4': [1351015147, 3917508701],
 'B5': {'E1': 0.4648265256372812, 'E2': 317771661},
 'B6': 13620,
 'B7': 0.1432957649230957,
 'B8': 31001}}

```

## 2. Двоичные данные:

```

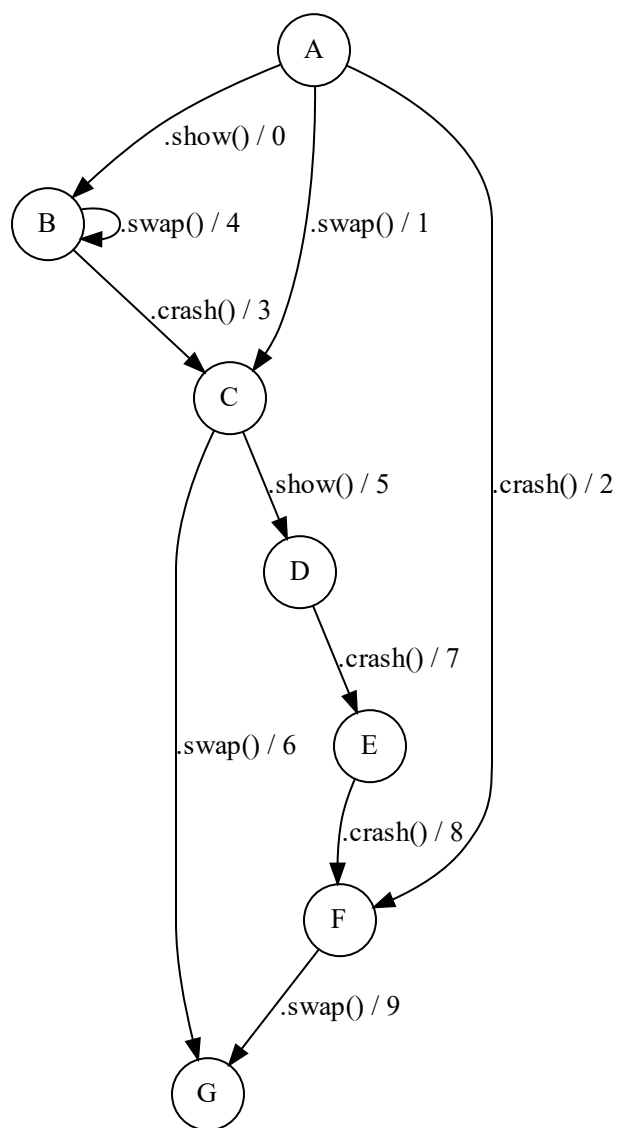
(b"VFRNY\b091o\xc1\x05\xd2\bfb'>1\xf3z_\xed\x1bE\xd81\xe5\xdf\xe05"
b'\xc5?\x80\xefP\bfb\xea\x8d\xc2\xb2\x181\x00g\xc4\x1ez\x84\x95q\x1f\xfaT'
b'\xbfqQ\xae\x8d\xf7[\xa5\x10\x95me\x80\x9f_\xc7\xa4\xe3\xa0j\xb86\x00'
b'2\xf3\x01\x078Q\xac\xf2\x02\x00)\x003z\xfe.`\xcb5\x18Z\x8b\x1e\xda'
b'^\xa2\xa8\x00!\xa7\xb2\xfeq\xcd\x07\xf9a\x8c\xc3e\x8e\xe2\xa7\xa7'
b'c\xba\xf4\x02\x00-\x00e.\x99T\x88\x90p+\x8dZ\x17\xc8\xb9\xe9\x083\xb4'
b"0^\x0e\xe8\xa3t\x15\xbe\x8b\xcf?\x1b@\x9f\x97\x19\x90SF\x1b\bfb'\xf3'")

```

Результат разбора:

```
{'A1': -0.28160129412290313,
 'A2': 2012369590179085863,
 'A3': 69,
 'A4': 0.16422723228823943,
 'A5': 128,
 'A6': 1779698717549351151,
 'A7': {'B1': -0.8319540619850159,
        'B2': -6522237041547259535,
        'B3': [{'C1': -27376,
                  'C2': {'D1': 2675991917,
                          'D2': 95,
                          'D3': 2699273415,
                          'D4': [-18326, 54, -3278, 1793, 20792, -3412],
                          'D5': [50279, 31262],
                          'D6': 788429363},
                  'C3': [96, -53, 83, 24, 90, -117, 30, -38]}],
        {'C1': -23970,
         'C2': {'D1': 2803957928,
                 'D2': 178,
                 'D3': 130904574,
                 'D4': [25081, -15476, -29083, -22558, 25511, -2886],
                 'D5': [38276, 8049],
                 'D6': 1419325029},
         'C3': [-120, -112, 112, 43, -115, 90, 23, -56]}],
        'B4': [856222137, 241061812],
        'B5': {'E1': 0.24645210311544408, 'E2': -1751171045},
        'B6': 36889,
        'B7': -0.6065418124198914,
        'B8': 10227}}
```

**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>	0
<code>o.swap()</code>	4
<code>o.swap()</code>	4
<code>o.crash()</code>	3
<code>o.show()</code>	5
<code>o.crash()</code>	7
<code>o.crash()</code>	8
<code>o.swap()</code>	9

2. Пример использования класса C32:

o = C32()	
o.show()	0
o.swap()	4
o.crash()	3
o.show()	5
o.crash()	7
o.crash()	8
o.swap()	9

## Вариант №2

**Задача 3.1.** Реализовать разбор двоичного формата данных (в духе формата WAD игры Doom или графического формата PNG). Данные начинаются с сигнатуры 0x48 0x4c 0x4b 0x72, за которой следует структура A. Порядок байт: от старшего к младшему. Адреса указаны в виде смещений от начала данных. В решении разрешено использовать модуль struct.

Структура A:	1	uint8
	2	Размер (uint32) и адрес (uint16) массива адресов (uint32) структур B
	3	int64
	4	int64
	5	Размер (uint32) и адрес (uint16) массива char
	6	Размер (uint32) и адрес (uint32) массива структур D
	7	double
	8	uint64

Структура B:	1	int32
	2	int64
	3	Структура C
	4	uint8
	5	double

Структура C:	1	uint32
	2	int8
	3	int16

Структура D:	1	Размер (uint16) и адрес (uint32) массива int8
	2	uint8

Примеры разбора двоичного формата с помощью функции f31:

1. Двоичные данные:

```
(b'HLKr\x7f\x00\x00\x00\x03\x00\x8dP\x1f\x8a\xd1P\x820\xf1\xfc.\xe8\x8c\xe'
b'gM\x01\x00\x00\x00\x02\x00\x99\x00\x00\x00\x02\x00\x00\x00\x9f\xbf\xba\xb2'
b'%\xfd5\x96 \x1d\xd7s&I`\xea\xcf\x9c\x12at\xb7\xf5R1_e\xbf\xd6\xc3uhTia\x1c'
b'\xe5\xbf\xe9\xc5N\x144\xaf\xc89\x0c\xd1\xec\xbc\xfa0\xc5;\xdf\xe'
b"\xaa#\xec\xa5xc0D\x8b\x8d\xef?\xe0\xf43L\xc2\xd50&x\xd5a<\x85\xb8\x90'\V\xa0"
b'6d\x97\xa0b\xbb\xca\x81\xb3\xbf\xa1\xf4\xb4?\xa0\xbc`\x00\x00\x00'
b'9\x00\x00\x00U\x00\x00\x00qji\x9b\x85\xb8*\x00\x02\x00\x00\x00'
b'\x9b\x1b\x00\x02\x00\x00\x00\x9d\x90')
```

Результат разбора:

```
{'A1': 127,
'A2': [{'B1': -1676517004,
```

```

'B2': -5191152370064179242,
'B3': {'C1': 3279251540, 'C2': 105, 'C3': 24860},
'B4': 229,
'B5': -0.8053350824970886},
{'B1': 957141484,
'B2': -4829494026730238294,
'B3': {'C1': 602711488, 'C2': 68, 'C3': -29811},
'B4': 239,
'B5': 0.5298096179236946},
{'B1': 645453153,
'B2': 4361094743452983350,
'B3': {'C1': 1687658594, 'C2': -69, 'C3': -13695},
'B4': 179,
'B5': -0.03507006909091426}],
'A3': 5773485878915772657,
'A4': -275026838988305151,
'A5': 'ji',
'A6': [{'D1': [-101, -123], 'D2': 27}], {'D1': [-72, 42], 'D2': 144}],
'A7': -0.1042808287798711,
'A8': 2150313955369741007}

```

## 2. Двоичные данные:

```

(b'HLKrr\x00\x00\x00\x04\x00\xa9\xad:\xd8\x88}\x8e\xb5\xa4?\xb7\xd0\xaen'
b'\xda\x8d,\x00\x00\x00\x02\x00\xb9\x00\x00\x00\x03\x00\x00\x00'
b'\xc1\xbf\xe5\xd29\xf6\xc9\x8f\x18`xA\xfa2\x87\x82\xe9\x0e\xea\xb0\x13d0x\xe9'
b'y\x18\xcb\x90\x9b\xab\x00;A\x84L5\xad?\xeeES7|\x8ap\x96\xda\xc2g(\xf5\x8f'
b'x\x0e\xfd<[0\xbd\x1b]\xc6\xa5u\xea?\xefm+k\xdfVV\xcb\xb0t\xba\xf1\xcb!'
b'\xd3\x84\x8a-\xc1\xdb"\x93\xa1U\xba\xd7\xb4\xbf\xc3\x89Z\x96\xb0\x99'
b'\xa0^\xc4B\x90\xaf\xad\x0cb\xdf\xbfct\xd62\xda&v]\x92\x1b\xbf\xd7\xf5'
b'\xa4\x134e\xe8\x00\x00\x009\x00\x00\x00U\x00\x00\x00q\x00\x00\x00\x8def\xe4'
b'\xed\x7fN\x9e\xcd\x00\x02\x00\x00\x00\xbb\xa3\x00\x02\x00\x00\x00\xbdm\x00'
b'\x02\x00\x00\x00\xbfI')

```

## Результат разбора:

```

{'A1': 114,
'A2': [{'B1': -357559452,
'B2': 3492798217342128283,
'B3': {'C1': 2868919105, 'C2': -124, 'C3': 19509},
'B4': 173,
'B5': 0.9459625323982674},
{'B1': -1764048281,
'B2': 2951422876621552731,
'B3': {'C1': 817699692, 'C2': -58, 'C3': -23179},
'B4': 234,

```

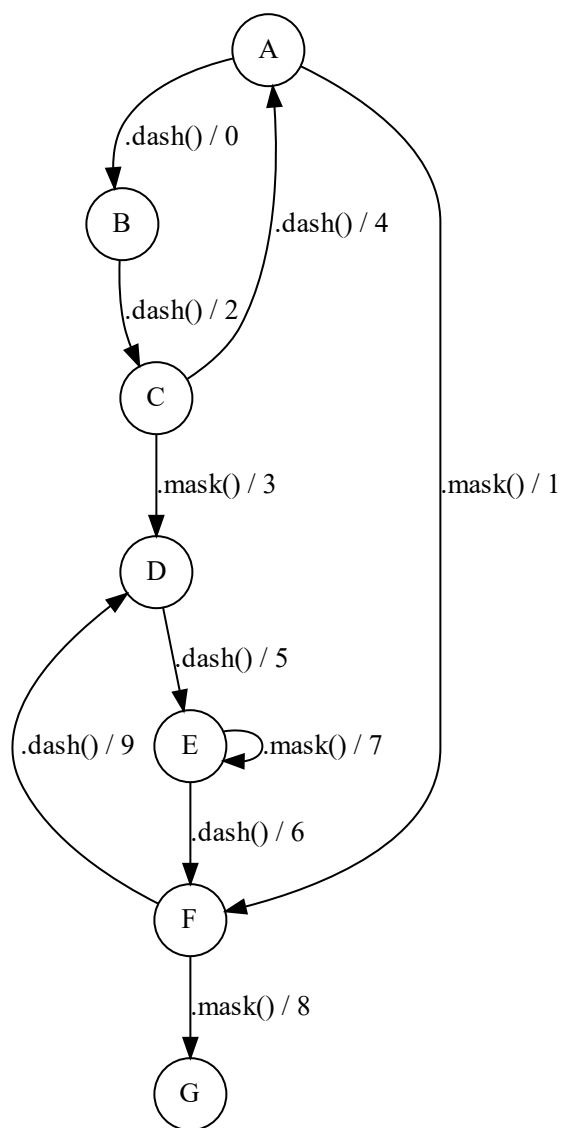
```

        'B5': 0.9820763689883261},
    {'B1': -877628230,
     'B2': -1023687297951191615,
     'B3': {'C1': 3676476321, 'C2': 85, 'C3': -17705},
     'B4': 180,
     'B5': -0.15262920720961137},
    {'B1': 1589920400,
     'B2': -5787956327301029004,
     'B3': {'C1': 3593656870, 'C2': 118, 'C3': 23954},
     'B4': 27,
     'B5': -0.37436773181397465}],
    'A3': -5964216675765144156,
    'A4': 4591367792730213676,
    'A5': 'ef',
    'A6': [{'D1': [-28, -19], 'D2': 163},
            {'D1': [127, 78], 'D2': 109},
            {'D1': [-98, -51], 'D2': 73}],
    'A7': -0.6819124050202587,
    'A8': 6951378534757755150}

```

**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.dash()</code>	0
<code>o.dash()</code>	2
<code>o.dash()</code>	4
<code>o.mask()</code>	1
<code>o.dash()</code>	9
<code>o.dash()</code>	5
<code>o.mask()</code>	7
<code>o.dash()</code>	6
<code>o.mask()</code>	8

2. Пример использования класса C32:

o = C32()	
o.dash()	0
o.dash()	2
o.dash()	4
o.mask()	1
o.dash()	9
o.dash()	5
o.mask()	7
o.mask()	7
o.dash()	6
o.mask()	8

## Вариант №3

**Задача 3.1.** Реализовать разбор двоичного формата данных (в духе формата WAD игры Doom или графического формата PNG). Данные начинаются с сигнатуры 0x4f 0x55 0x42 0xfb, 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	uint16
	2	int64
	3	int32
	4	Структура В
	5	int8
	6	int16
	7	Массив int16, размер 8

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

Структура С:	1	float
	2	uint16
	3	Массив адресов (uint32) структур D, размер 2
	4	int64
	5	int32
	6	int32

Структура D:	1	Размер (uint32) и адрес (uint16) массива int8
	2	uint64
	3	Массив uint8, размер 3
	4	Размер (uint32) и адрес (uint16) массива int64

Примеры разбора двоичного формата с помощью функции f31:

### 1. Двоичные данные:

```
(b'0UB\xfb\xeb\xa9-f\x8aH\x06\xa0\xd6dN\x07\xa9dA\xf9)~\x9d\x00\x00\x00\xd4\xbb'
b'\x05?\xb4\xb4\x07,\xcd{\xee?0\x9d\x90\x95\x06\x97\xc7\xbf"^/\x93\x15Q'
b'\x17\xcd\xe2\xbf5\x0ean\xe7\xc1y\x07=\xf2\n\x06\x03\x90\xe2\x9b\xd39\x1ed'
b'ky>\xf7\xfcSo\xc0\x0bW\x8e\xb0rAZ6\x03\x00\x00\x00A\x00z10F\xc9\xc7'
b'R\x0f\xd0\xce\xe9\x03\x00\x00\x00D\x00\xcc+\x98$4\xdf<0\x96\x1f\xd8#\xda'
b'\xbd\xc4\xc6\x05\x94\x82\x03\x00\x00\x00s\x00\xf1/\x91i``\x1fc\xfe'
b'\xae\x17S\x02\x00\x00\x00v\x00\t\x1d\x1d>\x84\xfe\\\x00\x00\x00\x86'
b'\x00\x00\x00u\x9e\xad\xa4\x9a\xda\xb4\xf4\x17n2\xde\x06\xf7\xad\xa3')
```

Результат разбора:

```
{'A1': 43499,
'A2': 7266170997656151597,
'A3': 1688799054,
'A4': {'B1': 2116680001,
      'B2': {'C1': 0.40842464566230774,
            'C2': 65156,
            'C3': [{'D1': [-14, 10, 6],
                    'D2': 1104164525941141882,
                    'D3': [208, 206, 233],
                    'D4': [7214267234110312451,
                          -4580349950067574421,
                          3916514786796263179]}],
            {'D1': [-52, 43, -104],
            'D2': 18330529407276494833,
            'D3': [174, 23, 83],
            'D4': [-2873412770171243484, -9037592200963302877]}]},
      'C4': -813785274924556683,
      'C5': -567120361,
      'C6': -1548880122},
'B3': 0.5223972797393799,
'B4': 0.9526124820673858,
'B5': -0.18429643919919547},
'A5': 34,
'A6': 12126,
'A7': [5523, 5969, -7475, 21439, 24846, -6290, 31169, 15623]}
```

## 2. Двоичные данные:

```
(b'0UB\xfb\xad\x86\xa9\xc8B\xd7H\xfaVA\xb3\xab6\xf7\xbf\x7f\xef\xce\x9c\x00'
b'\x00\x00X\xc2?\xbfFL\xf2\xe9\xf6\xa6\xec\xbf`\xbb\xe2N<[\xc9\xbf\xd0\xcb'
b'\xe9\xdd\xc6H\xd0\xb1@\xd4\xe0xw\xbeB\x9c\x9a5\x020\x82\x8d=\xa2\x90\x9f'
b'\x9cmF\x81\xe80SL\xdeEUh[\xa7M\x90> \xa8G\x03\x00\x00\x00A\x00v\x81`V\x83)'
b'^\xb7\xc4\x17\xea\x03\x00\x00\x00D\x00W\xbb\x1b\xc2h\xbc\xff\xbb'
b'\xc0\x98q\xa2\x1f\xd0W\xa1\xd8\x02\x00\x00\x00s\x00i\xc2\x0b(\xec\xb7c\x1dd'
b'0T\x02\x00\x00\x00u\x00\xbc\xde\xea\xbe\xf9\x87\\\x00\x00\x00\x85\x00'
b'\x00\x00g\xft\xfd\x96\xf6\xd5\xb1\xe7;\xf3\x19\xc1\x03\xca')
```

## Результат разбора:

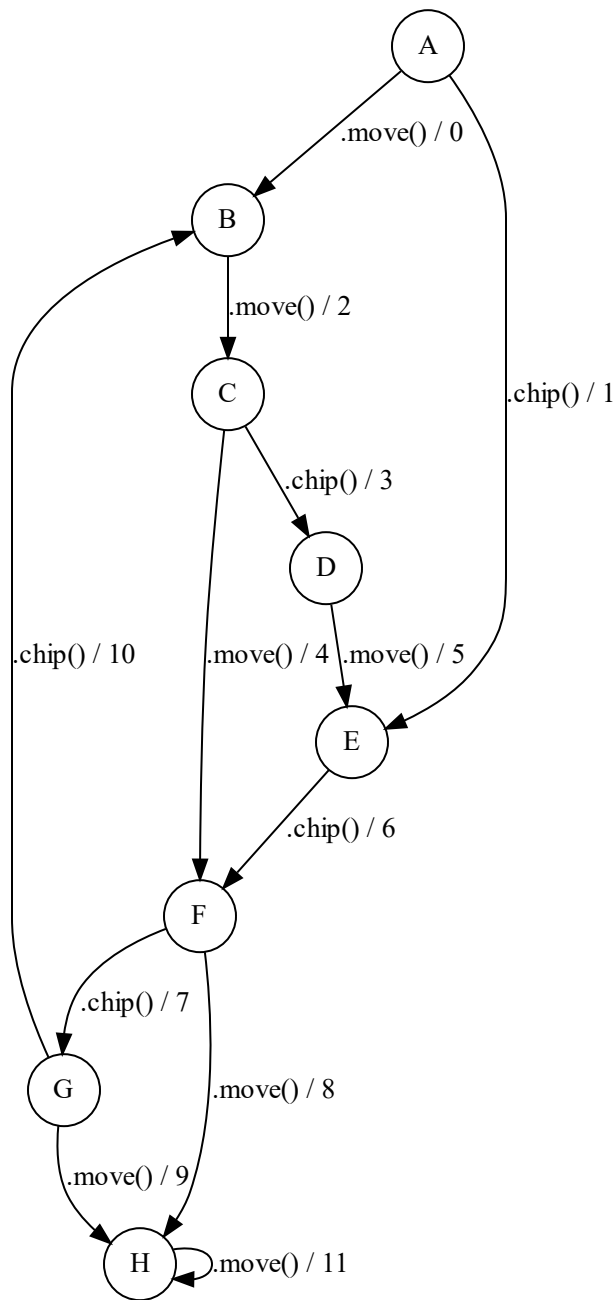
```
{'A1': 34477,
'A2': 4708225651218499753,
'A3': -147412045,
'A4': {'B1': 3471802303,
      'B2': {'C1': -0.4587305784225464,
            'C2': 34809,
            'C3': [{'D1': [48, -126, -115],
```

```

'D2': 13213044000818233718,
'D3': [196, 23, 234],
'D4': [-9131490674987720131,
       7517991974030495720,
       5163412425861343067]},
{'D1': [87, 88],
 'D2': 2117738474706289257,
 'D3': [100, 48, 84],
 'D4': [-4558769013106009157, -2836889738871148136]}],
'C4': -5632324630599109273,
'C5': 435370983,
'C6': -905742143},
'B3': -0.7490592002868652,
'B4': -0.8953814095751873,
'B5': -0.19809678890075144},
'A5': -48,
'A6': -5685,
'A7': [-14627, -12216, 16561, -7980, 30584, 17086, -25956, 565]}

```

**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.move()	0
o.chip()	RuntimeError
o.move()	2
o.move()	4
o.chip()	7
o.chip()	10
o.move()	2
o.chip()	3
o.move()	5
o.chip()	6
o.chip()	7
o.chip()	10
o.move()	2
o.move()	4
o.move()	8
o.move()	11
o.move()	11

## 2. Пример использования класса C32:

o = C32()	
o.chip()	1
o.chip()	6
o.chip()	7
o.chip()	10
o.move()	2
o.chip()	3
o.move()	5
o.chip()	6
o.chip()	7
o.chip()	10
o.move()	2
o.move()	4
o.move()	8
o.move()	11
o.move()	11

## Вариант №4

**Задача 3.1.** Реализовать разбор двоичного формата данных (в духе формата WAD игры Doom или графического формата PNG). Данные начинаются с сигнатуры 0x53 0x5a 0x59 0x4a 0x28, 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	uint64
	2	Адрес (uint16) структуры В
	3	Адрес (uint32) структуры Е
	4	float

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

Структура С:	1	uint16
	2	Массив uint8, размер 2
	3	int64

Структура D:	1	Размер (uint16) и адрес (uint32) массива int16
	2	int32
	3	uint64
	4	int64
	5	uint32

Структура Е:	1	Размер (uint32) и адрес (uint32) массива uint64
	2	int32

Примеры разбора двоичного формата с помощью функции f31:

### 1. Двоичные данные:

```
(b'SZYJ(\xb0n\xbd\xfb\t5\x16.\x00m\x00\x00\x00\x97\xbfu;F\xc4\xd4Y\xe4\x91'
b'^\x1b\x83\xffB\xd6\x10\x1c\x85\x05\xa5\x03K-\x8f\x9f\x00\x87\xae\x9f'
b'm\xeb!\xebQ\x9c(\xf9\xe9!>\x00\x17\x00#\x00/\xec\x98\x13\x1c\xb1\xa9\x17'
b'\xe6\x7f\xec\xdf\xb5\xd45\x00\x05\x00\x00\x00E\x10\x9f\xfa-\xbd(\xf2'
b'\xb5[\x9c\xed\x83t\x17\xf6\x07\x93\xee\x88\x82\xdeu\n\xdf*J!\xfb\x00\x03\x00'
b'\x00\x00;\x00\x04\x00A\xf3\\\x000\xf4\xa8Wx\xfa\x8d\x10~\x02\xf4:\xb8\xe0'
b'\x15\x8e\xf7\xd3\xd1\xbf\xean\xf2\xf7\xa9\x00\x00\x00\x03\x00'
b'\x00\x00\x7f\xa0e\xac\xc4')
```

Результат разбора:



```
{'A1': 12713307684002403886,
'A2': {'B1': 709501435,
'B2': [{'C1': 50388, 'C2': [89, 228], 'C3': -7971904036661963248},
{'C1': 7301, 'C2': [5, 165], 'C3': 237333501748217774},
{'C1': 40813, 'C2': [235, 33], 'C3': -1490238301877493442}],
'B3': [-20, -104, 19, 28],
'B4': 243,
'B5': 92,
'B6': {'D1': [-20055, 6118, 32748, -8267, -11211],
'D2': 278919725,
'D3': 13630411132976819587,
'D4': 8365425345271072898,
'D5': 3732212447}},
'A3': {'E1': [17629436918453047422, 212859648101158647, 15263191625739990953],
'E2': -1603949372},
'A4': -0.9579356908798218}
```

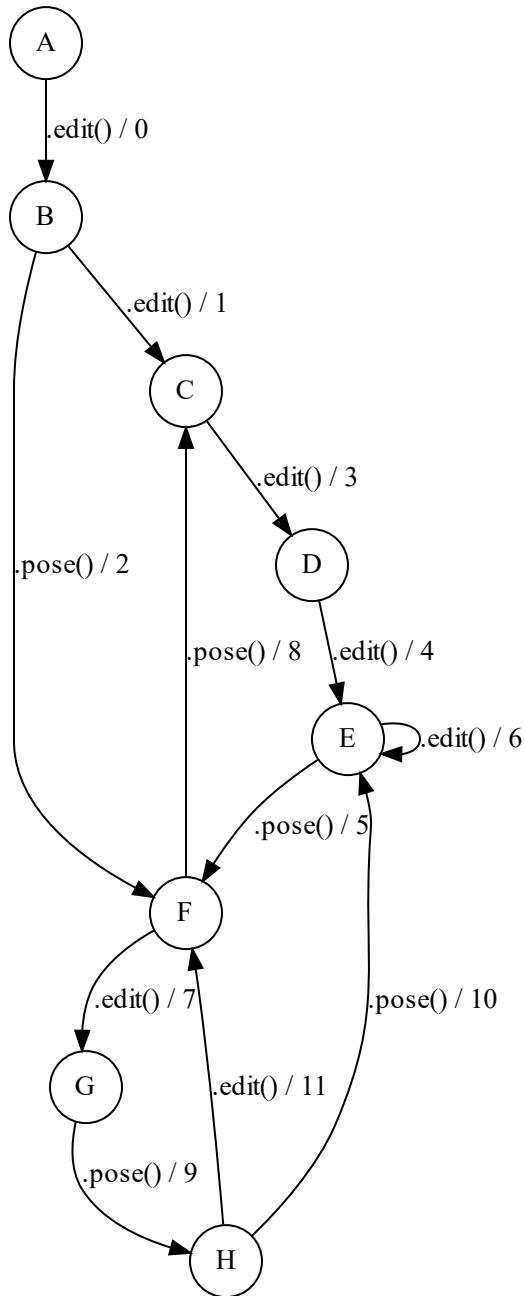
## 2. Двоичные данные:

```
(b'SZYJ(\xcak\xa2Mi\xb5\xe6\x00j\x00\x00\x00\x94\xbe\xde\x90S\xf4y\x1e\x9b\xa5'
b'\xb0=\xea\xfc\xfcIC\xb9\xfd\xc7:\x89r\xb9\xfd\xaf?R\xb2\xb5\x8d\x00\xd5)'
b'\x87\x16\x98\x16==\xe0\x00\x17\x00#\x00/\xba\xb4 \xa8E\xdc\x11\x18B0a'
b'\x00\x03\x00\x00\x00F\x94)\xde\xb3t\x88\x078\xd92\x11\xc9\n\xba'
b'\xcb\x99\xb7\x11}\xdc9\xa5:3\xf0G\xf2\xaf\x00\x03\x00\x00\x00;\x00\x05\x00A'
b'\xec*\x00L\xac\xfc88`\x8bZ\x17i\xa3de\x88\xdd\x0b^\xcc\x91u$17K|E'
b'\x00\x00\x00\x03\x00\x00\x00|\xbd\x89\xe1u')
```

## Результат разбора:

```
{'A1': 14585930271539193438,
'A2': {'B1': -263720273,
'B2': [{'C1': 62585, 'C2': [30, 155], 'C3': -6507633382074332861},
{'C1': 47613, 'C2': [199, 58], 'C3': -8542561043950382414},
{'C1': 46477, 'C2': [0, 213], 'C3': 2992385319874936288}],
'B3': [-70, -76, 32, -88, 69],
'B4': 236,
'B5': 42,
'B6': {'D1': [-9199, 6210, 20321],
'D2': -1809195341,
'D3': 8396969445975790025,
'D4': 773154147109273052,
'D5': 967129651}},
'A3': {'E1': [12463773956053800809,
11773646964351786700,
10481323754977459269],
'E2': -1115037323},
'A4': -0.4346948564052582}
```

**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.edit()          0
o.pose()          2
o.edit()          7
o.pose()          9
o.edit()          11
o.edit()          7
o.pose()          9
o.pose()          10
o.pose()          5
o.pose()          8
o.edit()          3
o.edit()          4
o.edit()          6
o.pose()          5

```

## 2. Пример использования класса C32:

```

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

```

## Вариант №5

**Задача 3.1.** Реализовать разбор двоичного формата данных (в духе формата WAD игры Doom или графического формата PNG). Данные начинаются с сигнатуры 0x56 0x53 0x59 0x47, за которой следует структура A. Порядок байт: от младшего к старшему. Адреса указаны в виде смещений от начала данных. В решении разрешено использовать модуль struct.

Структура A:	1	double
	2	uint32
	3	Массив char, размер 4
	4	uint8
	5	double
	6	uint32
	7	Адрес (uint32) структуры B
	8	uint16

Структура B:	1	Массив адресов (uint16) структур C, размер 3
	2	Адрес (uint16) структуры D
	3	double

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

Структура D:	1	uint8
	2	uint16
	3	Размер (uint32) и адрес (uint16) массива double
	4	Массив int16, размер 5
	5	uint8

Примеры разбора двоичного формата с помощью функции f31:

1. Двоичные данные:

```
(b'VSYG\xfc\x0e#\x15t\xd4\xbf\xcc\x19<\xa3svpmQP\x9((\xb4}\xe9\xbf]);'  
b'\x86w\x00\x00\x00\x17z\x0f\xdfF"\x1f\xc3\x07\x00'\x00\x00\x00\x10-M\xbf'  
b'\x9a\rn\x03\x008\x00\x00\x00X\xcf\xbf\x06\xfc\x8a\xe3\x04\x00E'  
b'\x00\x00\x00\x8f\x93\x97\xbe\x10\xa0\xc4\x8c\xc9\r\xd7\xbfXV\xa5+\xd2'  
b'[\xc7\xbf\x94\x8a\x91\x02\x00\x00\x00S\x00\xac\xad\xc2\x02a\x1e}zA27.'  
b'\x00;\x00I\x00c\x00 \x04\xbe,\x1b\xd2\xc4\xbf')
```

Результат разбора:

```
{'A1': -0.31958520145802516,  
'A2': 2738624972,  
'A3': 'svpm',  
'A4': 81,
```

```
'A5': -0.7965946945549067,
'A6': 2252024157,
'A7': {'B1': [{'C1': [15, -33, 70, 34, 92, 31, -61],
                  'C2': -0.8014688491821289},
          {'C1': [-102, 13, 110], 'C2': -0.9211325645446777},
          {'C1': [6, -4, -118, -29], 'C2': -0.2960476577281952}],
       'B2': {'D1': 148,
              'D2': 37258,
              'D3': [-0.36021651026612833, -0.18248965388436278],
              'D4': [-21076, 706, 7777, 31357, 12865],
              'D5': 55},
       'B3': -0.1626619308918018},
'A8': 31255}
```

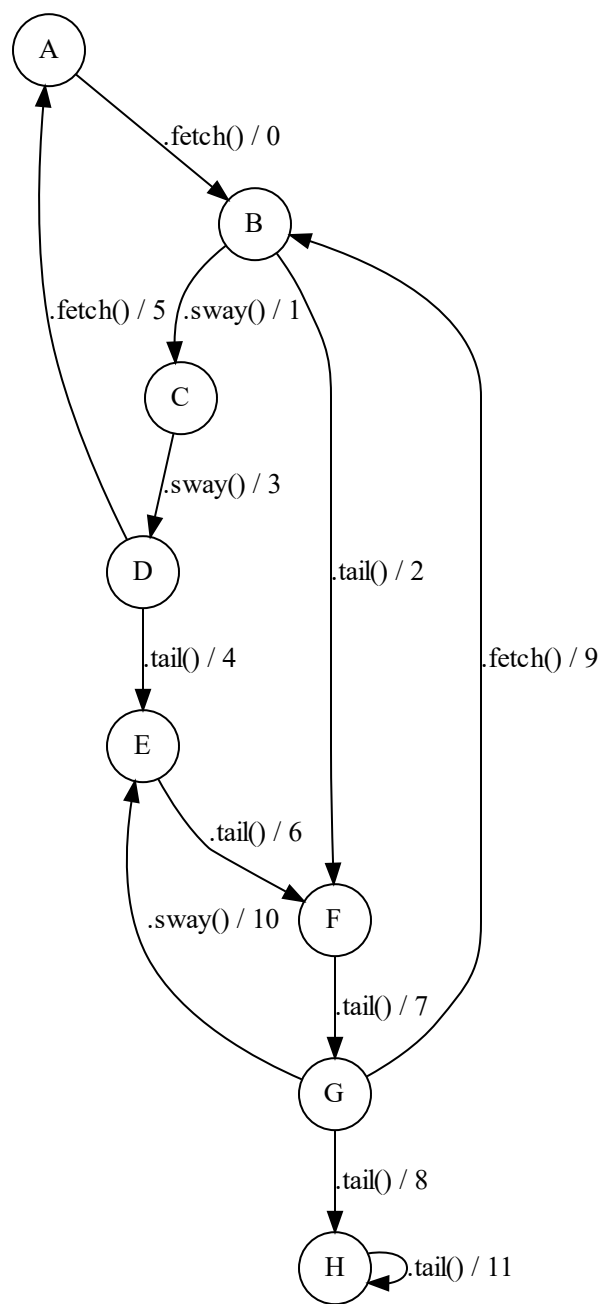
## 2. Двоичные данные:

```
(b'VSYG\xb0\xa2\x9d\xe4\xdf\xab\xb2?\x96K\x9e\xf5lft\xff\xec\x04\xd6\xc3_\xe6'
b'?\xbf 1\xb6p\x00\x00\x00\xca\xfd:x\x81\x03\x00'\x00\x00\x007\x16u\xbf"
b'\xb6\x8d\x02\x004\x00\x00\x00\xf7m\xdd\xbe\xbd\xb0\x02\x00@\x00\x00\x00'
b'\xb9\x94z\xbf<\xf8\x964\xd0\xda\xd2?\xe0X\xc6\xf5x\xaa\xbd?\x17\xfb\xa5\x02'
b'\x00\x00\x00L\x00\xdf\x97\xee\xf8\x8eS\x03\x83\xd4P@*\x006\x00B\x00\\\x00'
b'\xden\x9a8"\x06\xe0\xbf')
```

## Результат разбора:

```
{'A1': 0.07293509799422782,
'A2': 4120791958,
'A3': 'lft',
'A4': 255,
'A5': 0.699190061575234,
'A6': 3056672959,
'A7': {'B1': [{'C1': [58, 120, -127], 'C2': -0.9573702216148376},
          {'C1': [-74, -115], 'C2': -0.432479590177536},
          {'C1': [-67, -80], 'C2': -0.9788318276405334}],
       'B2': {'D1': 23,
              'D2': 42491,
              'D3': [0.29460530410495145, 0.11588245392190144],
              'D4': [-26657, -1810, 21390, -31997, 20692],
              'D5': 64},
       'B3': -0.5007487397701558},
'A8': 64970}
```

**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.fetch()	0
o.sway()	1
o.sway()	3
o.fetch()	5
o.fetch()	0
o.tail()	2
o.tail()	7
o.fetch()	9
o.tail()	2
o.fetch()	RuntimeError
o.tail()	7
o.sway()	10
o.fetch()	RuntimeError
o.tail()	6
o.tail()	7
o.tail()	8
o.tail()	11

## 2. Пример использования класса C32:

o = C32()	
o.fetch()	0
o.sway()	1
o.sway()	3
o.fetch()	5
o.fetch()	0
o.tail()	2
o.tail()	7
o.fetch()	9
o.sway()	1
o.fetch()	RuntimeError
o.sway()	3
o.tail()	4
o.tail()	6
o.tail()	7
o.tail()	8
o.tail()	11
o.tail()	11

## Вариант №6

**Задача 3.1.** Реализовать разбор двоичного формата данных (в духе формата WAD игры Doom или графического формата PNG). Данные начинаются с сигнатуры 0xb3 0x42 0x5a 0x4c, 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 char, размер 7
	3	Адрес (uint32) структуры B
	4	Размер (uint16) и адрес (uint32) массива uint16
	5	int16
	6	int8
	7	float
	8	uint32

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

Структура C:	1	int8
	2	uint16

Структура D:	1	int16
	2	uint64

Примеры разбора двоичного формата с помощью функции f31:

### 1. Двоичные данные:

```
(b'\xb3BZL`3\xf5Q\xaf\xa0\r\tqehtrxw\x00\x00\x00N\x00\x04\x00\x00\x00'
b'`\x9b\xfd\xde?t\xd8\xbc\x1c%\tA\xe4B\x15\xd6\x94\x1a\xeb\xfd\xce\xf3\xbfF'
b'1\xa1[\xfa0[(\x01\x83J\xd2\x07\x84\xa3R\xc5\xb3\xab\xc2\xac\xb7\x03\x9bm'
b'*\x93\x00\x00\x00\x04\x00\x00\x00(\x00\x00\x00\x02\x00\x00\x004\x00D'
b'\xbfL\xaab\xc0\x9c\x88\xeb')
```

Результат разбора:

```
{'A1': 6932153982640983305,
'A2': 'qehtrxw',
'A3': {'B1': [{'C1': -28, 'C2': 16917},
               {'C1': -42, 'C2': 37914},
               {'C1': -21, 'C2': 64974},
               {'C1': -13, 'C2': 48966}],
'B2': [7827638757457864705, -8986139176732503355],
'B3': {'D1': -19541, 'D2': 14027788165472660115}},
```



```
'A4': [48972, 43618, 49308, 35051],
'A5': -25603,
'A6': -34,
'A7': 0.9564321041107178,
'A8': 472189249}
```

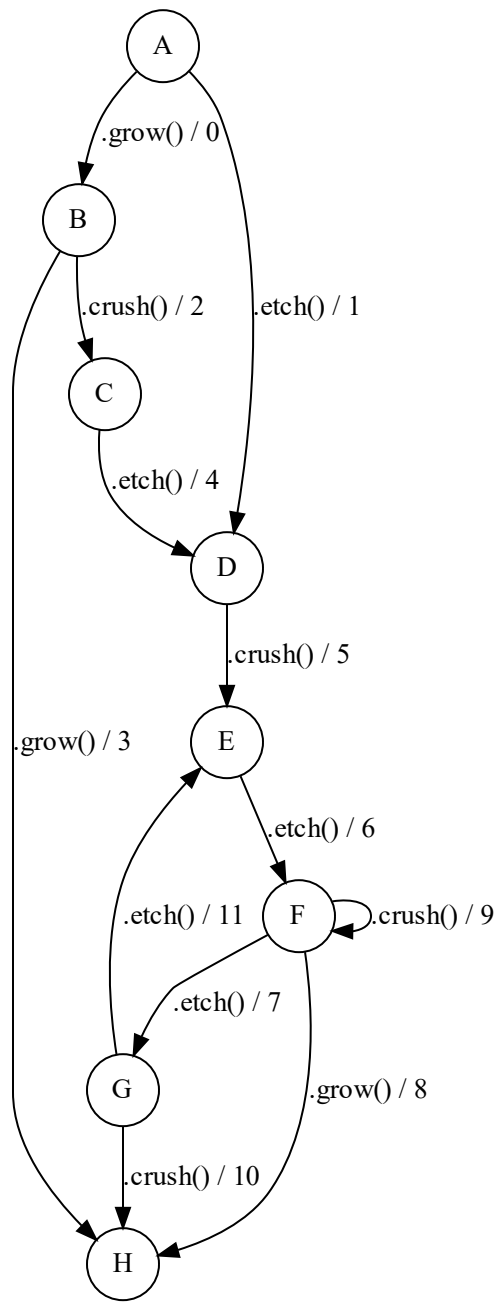
## 2. Двоичные данные:

```
(b'\xb3BZL\xb3\xab#\x19\xa1\xc1\xda\xdmj\nupdp\x00\x00\x00T\x00\x05\x00\x00\x00'
b'f7\xf8\xdf\xbf\x07\xc9\xce\x15|\x0bj.N\xb7+d\xcd\x01h\x04U\x87'
b'\xd9\xaf\x84\x8e\t\x10\xed\xca\xf8\x97|J\xe75\xeb\xba\x06\\\xda'
b'\xfc\xa2\x85\xda\x18(\x005\x93\xae\x80\x00\x00\x00\x06\x00\x00\x00('
b'\x00\x00\x00\x02\x00\x00\x00:\x00J\xdf\x9e\xf7\x83\xaf\x04\x02\xaa[')
```

## Результат разбора:

```
{'A1': 12946480146801613525,
'A2': 'mj\nupdp',
'A3': {'B1': [{'C1': 106, 'C2': 11854},
               {'C1': -73, 'C2': 11108},
               {'C1': -51, 'C2': 360},
               {'C1': -44, 'C2': 21895},
               {'C1': -39, 'C2': 44932},
               {'C1': -114, 'C2': 2320}],
        'B2': [-1311963011916044491, -1479227185186538334],
        'B3': {'D1': -31277, 'D2': 6996386002484047488}},
'A4': [57246, 32643, 44901, 46258, 43611],
'A5': 14328,
'A6': -33,
'A7': -0.5304137468338013,
'A8': 3457514507}
```

**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.grow()</code>	0
<code>o.etch()</code>	RuntimeError
<code>o.crush()</code>	2
<code>o.etch()</code>	4
<code>o.grow()</code>	RuntimeError
<code>o.crush()</code>	5
<code>o.etch()</code>	6
<code>o.crush()</code>	9
<code>o.etch()</code>	7
<code>o.etch()</code>	11
<code>o.grow()</code>	RuntimeError
<code>o.etch()</code>	6
<code>o.crush()</code>	9
<code>o.grow()</code>	8

## 2. Пример использования класса C32:

<code>o = C32()</code>	
<code>o.crush()</code>	RuntimeError
<code>o.grow()</code>	0
<code>o.crush()</code>	2
<code>o.etch()</code>	4
<code>o.crush()</code>	5
<code>o.etch()</code>	6
<code>o.crush()</code>	9
<code>o.etch()</code>	7
<code>o.etch()</code>	11
<code>o.etch()</code>	6
<code>o.etch()</code>	7
<code>o.crush()</code>	10

## Вариант №7

**Задача 3.1.** Реализовать разбор двоичного формата данных (в духе формата WAD игры Doom или графического формата PNG). Данные начинаются с сигнатуры 0xd3 0x46 0x4e 0x4f, 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	uint64
	2	float
	3	double
	4	uint16
	5	int8
	6	Структура В
	7	uint32

Структура В:	1	Структура С
	2	float
	3	int8
	4	int16
	5	Массив int32, размер 3
	6	Размер (uint16) и адрес (uint16) массива int8
	7	uint8
	8	Размер (uint16) и адрес (uint32) массива int16

Структура С:	1	uint64
	2	Массив структур D, размер 2

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

Примеры разбора двоичного формата с помощью функции f31:

### 1. Двоичные данные:

```
(b'\xd3FN0\xe9JU\xe9L\xd0\x80\x90\xbe2\x11\xa5?\xcc\x99<`\x89\xde\x98'  
b'u\x83\xd7\x8c\xa0\x8f\xb1\x06\xbe"\x83\xbf\xe8\x9f\xb0\x03\xb6F\xc2\x00'  
b'\x00\x00\x03\x00s:\xee\x8f[\x19Jr\xa6M\xbf\xe0|\xc9\xe2\x88\x04\xb6\x00\x00'  
b'\x00\x03\x00\x7f\xf6\x9c\xf6\xaa\xbeR\x88\\\x9b>\xa0\xc4-\x94\xb1\xdfbb\x11D'  
b'\xf0\x19J\x03\xb70\x1c\x9f\x00\x06\x00\x8bc\x00\x03\x00\x00\x91y'  
b'\xe78\xe0\xe7\xcd\x00\xa7(\xbd03\x9a8\xc2\xa8\x11\x84\x81\x84\xfd'  
b'\x81\xdf}\xcf\xe1\n\x98\xf0\xc9j\xc0\xbf\x82;\xcc\xcdI\xf6\x88')
```

Результат разбора:

```
{'A1': 16815691044175773840,
'A2': -0.1738954335451126,
'A3': 0.2234263869647648,
'A4': 30083,
'A5': -41,
'A6': {'B1': {'C1': 10133257152068723331,
              'C2': [{'D1': -0.7694931099415145,
                      'D2': [-405995353, 683503411, -1707556184],
                      'D3': 4246489120086061734,
                      'D4': 77},
                    {'D1': -0.5152329849662667,
                      'D2': [293896580, -41820291, -807335272],
                      'D3': -676394630814463908,
                      'D4': -101}]}},
'B2': 0.31399670243263245,
'B3': -108,
'B4': -20001,
'B5': [1650594116, -266778109, -1221583713],
'B6': [-16, -55, 106, -64, -65, -126],
'B7': 99,
'B8': [15308, -12983, -2424]},
'A7': 2045196512}
```

## 2. Двоичные данные:

```
(b'\xd3FN0\x03\xe3\xd2\xa8\xde\xabi\x0e?\r\x90\x7f?\xe2p\xab\x1b\t\xa7n'
b'\x17q,\x7f\xac\x82R\xd7qe\xb1\xbf\xe65o\xc5\x1a{\xcc\x00\x00\x02\x00'
b's\xda\xb1G\xcbT@P\xc4u\xbf\xeb\x9c \xbe\x92i\xfc\x00\x00\x04\x00{'
b'\xe3\xe0\x0e\x008\xc6\x10\xef\xdd?\x04\xec\xe7\x14q\x1b^\x1f)_z\xc5\x80V'
b'\x97\x1a\xd1\x8b\x00\x06\x00\x8bX\x00\x02\x00\x00\x00\x91YyL}?\x191+'
b'\xb6@\xae\xdd\xebG\xf2\xe10k\x01\x9f\x06\xa8b\xbf*\xfd\xbbC\xbb\xe8a\x10'
b'\xba5\x9c\xef\xbc')
```

## Результат разбора:

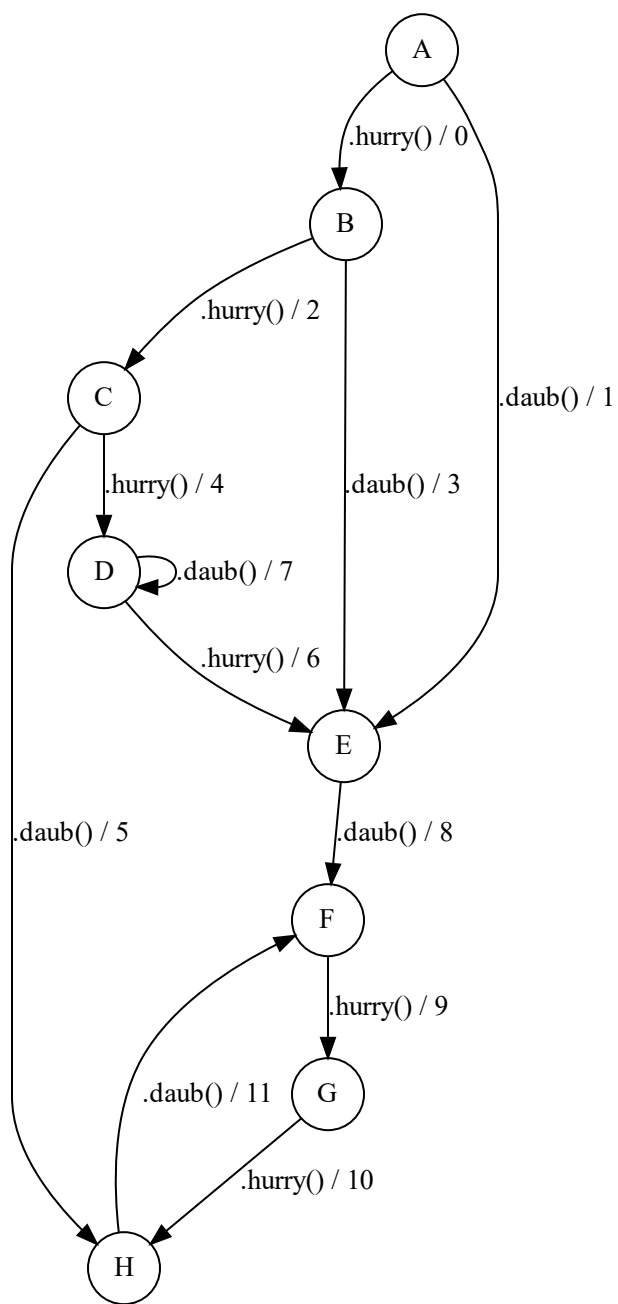
```
{'A1': 280299224559216910,
'A2': 0.5529860854148865,
'A3': 0.576253464515714,
'A4': 6001,
'A5': 44,
'A6': {'B1': {'C1': 9199871431124542897,
              'C2': [{'D1': -0.6940230226208484,
                      'D2': [1060051249, 733364398],
                      'D3': -2688288563946041148,
                      'D4': 117},
                    {'D1': -0.8628085825069429,
```

```

        'D2': [-571783182, -514888959, -1626953630, -1087701573],
        'D3': -2026604438201429777,
        'D4': -35}}},
    'B2': 0.5192398428916931,
    'B3': 20,
    'B4': 28955,
    'B5': [1579100511, 2059763798, -1759850101],
    'B6': [67, -69, -24, 97, 16, -70],
    'B7': 88,
    'B8': [13724, -4164]},
    'A7': 1501121661}

```

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



1. Пример использования класса C32:

o = C32()	
o.hurry()	0
o.hurry()	2
o.hurry()	4
o.daub()	7
o.daub()	7
o.hurry()	6
o.daub()	8
o.hurry()	9
o.hurry()	10
o.daub()	11
o.hurry()	9
o.hurry()	10
o.daub()	11

## 2. Пример использования класса C32:

o = C32()	
o.hurry()	0
o.hurry()	2
o.hurry()	4
o.daub()	7
o.hurry()	6
o.daub()	8
o.hurry()	9
o.hurry()	10
o.hurry()	RuntimeError
o.daub()	11
o.hurry()	9
o.daub()	RuntimeError
o.hurry()	10
o.daub()	11
o.hurry()	9
o.hurry()	10
o.daub()	11
o.hurry()	9



## Вариант №8

**Задача 3.1.** Реализовать разбор двоичного формата данных (в духе формата WAD игры Doom или графического формата PNG). Данные начинаются с сигнатуры 0x5a 0x47 0x4c 0x52 0x63, 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	int16
	3	int16
	4	Структура B
	5	int8
	6	Размер (uint16) и адрес (uint16) массива uint8
	7	Массив float, размер 3
	8	Массив int8, размер 4

Структура B:	1	Массив адресов (uint32) структур C, размер 4
	2	uint64
	3	uint32
	4	Структура D
	5	uint32
	6	Массив int64, размер 4
	7	int8

Структура C:	1	int8
	2	uint16

Структура D:	1	int64
	2	float

Примеры разбора двоичного формата с помощью функции f31:

1. Двоичные данные:

```
(b'ZGLRc:\xac\x94\xc5\xb6\x1e,\xca\x7f\x15\xfc\xd3\x00\x00\x00s\x00\x00\x00'
b'v\x00\x00\x00y\x00\x00\x00|\x91\xe6\xceL\xd5n\x19,\xf8\x01:S\xb9\xd2\x11'
b'3\x8a\xcf&J\xbe\x14\xec\xb4\xe8\xe0\xf0\xb1\x86\x02\x15\x10\x83\x1a0'
b"\xf6\xe4\xc4Cw\x8b\xb7\xcb#\r\x1c5o\x96$Wu'd\xa6c\tj\xba7\xca\xd4\x00"
b'\x03\x00\x7f=\x1c\\\x9a>S\x96*\xbe\x7f*\xfc\xddU!\xd4\x93\xc7\x18K\xa7'
b'\xbf(\x96;\xabP\xbc\x0f\x0eT')
```

Результат разбора:

```
{'A1': 4227917727078952138,
'A2': 32533,
'A3': -813,
```

```

'A4': {'B1': [{'C1': -109, 'C2': 50968},
               {'C1': 75, 'C2': 42943},
               {'C1': 40, 'C2': 38459},
               {'C1': -85, 'C2': 20668}],
       'B2': 10513317209536600364,
       'B3': 4160830035,
       'B4': {'D1': -5056960518513809846, 'D2': -0.1454342007637024},
       'B5': 3907055793,
       'B6': [-8790440362010595082,
               -1962369494341399773,
               944688775217567605,
               2838576609440217655],
       'B7': -54},
'A5': -44,
'A6': [15, 14, 84],
'A7': [0.03817424923181534, 0.2066275179386139, -0.2491874098777771],
'A8': [-35, 85, 33, -44]}

```

## 2. Двоичные данные:

```

(b'ZGLRc\xde\xe9\xbf\xf6\xe0\x8a>\xc3\x1a1:\x9b\x00\x00\x00s\x00\x00\x00'
 b'v\x00\x00\x00y\x00\x00\x00|\xe5\xe0\x05>hE8\xd2P\x99\x1d\x06I\x1b\x06/@5*'
 b'\xa7?#\x84\x97\xc2r>\xc6\xa6@F#\x0b>\x18@\xa5n\xaa\x81`pM\xd7z\xe1D^i1\x9b'
 b'kHx\\*$\xef\x06\x86\xd4\x15\x00\x03\x00\x7f\xbf\x01b\x16\xbfG(\xd9?'
 b'B\xf3\xe6\x15,zJ\xf7\x8b\xf4\xf1\x048\xa19\xfd\x96\xa3U4\xf1\xd8')

```

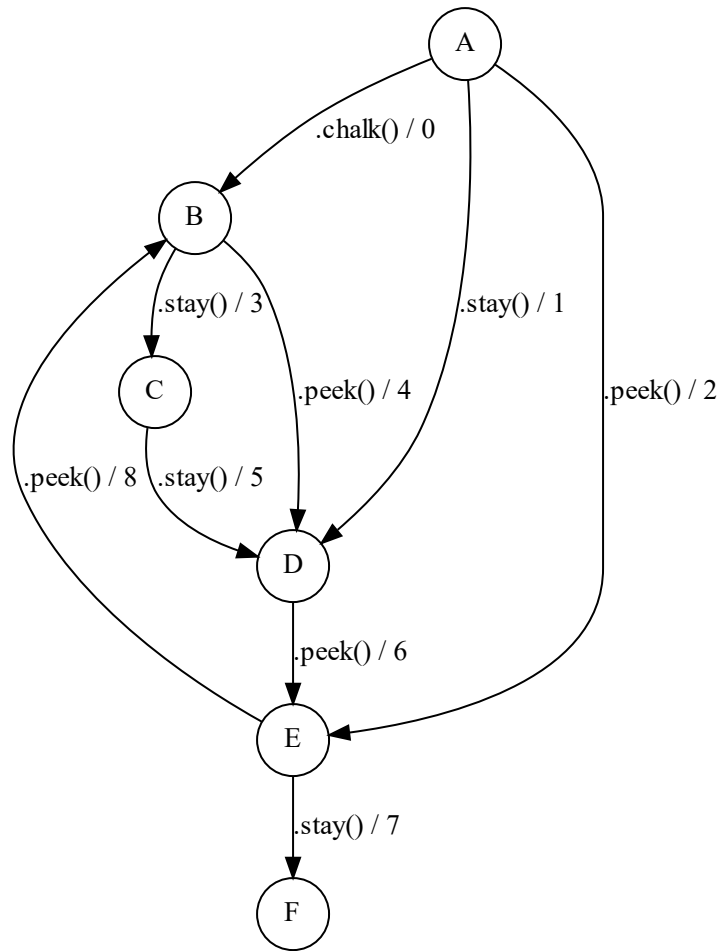
## Результат разбора:

```

{'A1': 16062580613043601091,
 'A2': 6764,
 'A3': 15003,
 'A4': {'B1': [{'C1': -9, 'C2': 35828},
               {'C1': -15, 'C2': 1080},
               {'C1': -95, 'C2': 14845},
               {'C1': -106, 'C2': 41813}],
       'B2': 16564245195064162514,
       'B3': 1352211718,
       'B4': {'D1': 5267810989150382759, 'D2': 0.6387419104576111},
       'B5': 3262267078,
       'B6': [-6467091948577613760,
               -6526091337367794217,
               8854433514673445739,
               5222025104014116486],
       'B7': -44},
 'A5': 21,
 'A6': [52, 241, 216],
 'A7': [-0.5054029226303101, -0.7779670357704163, 0.7615340948104858],
 'A8': [21, 44, 122, 74]}

```

**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.stay()</code>	1
<code>o.stay()</code>	RuntimeError
<code>o.peak()</code>	6
<code>o.peak()</code>	8
<code>o.stay()</code>	3
<code>o.stay()</code>	5
<code>o.peak()</code>	6
<code>o.peak()</code>	8
<code>o.peak()</code>	4
<code>o.peak()</code>	6
<code>o.stay()</code>	7

## 2. Пример использования класса C32:

<code>o = C32()</code>	
<code>o.chalk()</code>	0
<code>o.peek()</code>	4
<code>o.chalk()</code>	RuntimeError
<code>o.peek()</code>	6
<code>o.chalk()</code>	RuntimeError
<code>o.peek()</code>	8
<code>o.chalk()</code>	RuntimeError
<code>o.stay()</code>	3
<code>o.stay()</code>	5
<code>o.peek()</code>	6
<code>o.stay()</code>	7

## Вариант №9

**Задача 3.1.** Реализовать разбор двоичного формата данных (в духе формата WAD игры Doom или графического формата PNG). Данные начинаются с сигнатуры 0x2a 0x5a 0x51 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	Структура B
	2	uint8
	3	uint8
	4	Массив структур D, размер 4
	5	uint32

Структура B:	1	uint8
	2	Адрес (uint16) структуры C

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

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

Примеры разбора двоичного формата с помощью функции f31:

### 1. Двоичные данные:

```
(b'*ZQIQK\x00\xaae\xfd\x94\xd0@\xceA\xbfk\xa3\x08\x00U\x00\x00\x00\xe7u\x94B'
b'\xb4\xcf>2\xcd\x03\x00]\x00\x00\x00i\xa5\x91Xa~?\x91\x87\x06\x00`\x00'
b'\x00\x00\xc0*\xb6\xee\xe8$\xbf\x0c\x96\x04\x00f\x00\x00\x00\x88&\xde'
b'\x14z1\xd3\x92\xb9\n\x02\x00\x00\x00I\x00\x07\xa5t\xc2#\xd0\xbf8=\xa4\xe0'
b'\x05\xc3\xbe\xdf\xe95\xda\xd1!\xf0')
```

Результат разбора:

```
{'A1': {'B1': 81, 'B2': {'C1': 179933907, 'C2': 'z1'}}},
'A2': 170,
'A3': 101,
'A4': [{'D1': [-3, -108, -48],
'D2': -0.7570533752441406,
'D3': -23701,
'D4': [7, -91, 116, -62, 35, -48, -65, 56]},
{'D1': [-25, 117, -108],
'D2': 0.4056721329689026,
'D3': -13006,
'D4': [61, -92, -32]}],
```

```
{'D1': [105, -91, -111],
  'D2': 0.9936728477478027,
  'D3': -30831,
  'D4': [5, -61, -66, -33, -23, 53]},
{'D1': [-64, 42, -74],
  'D2': -0.6441792249679565,
  'D3': -27124,
  'D4': [-38, -47, 33, -16]}},
'A5': 350103176}
```

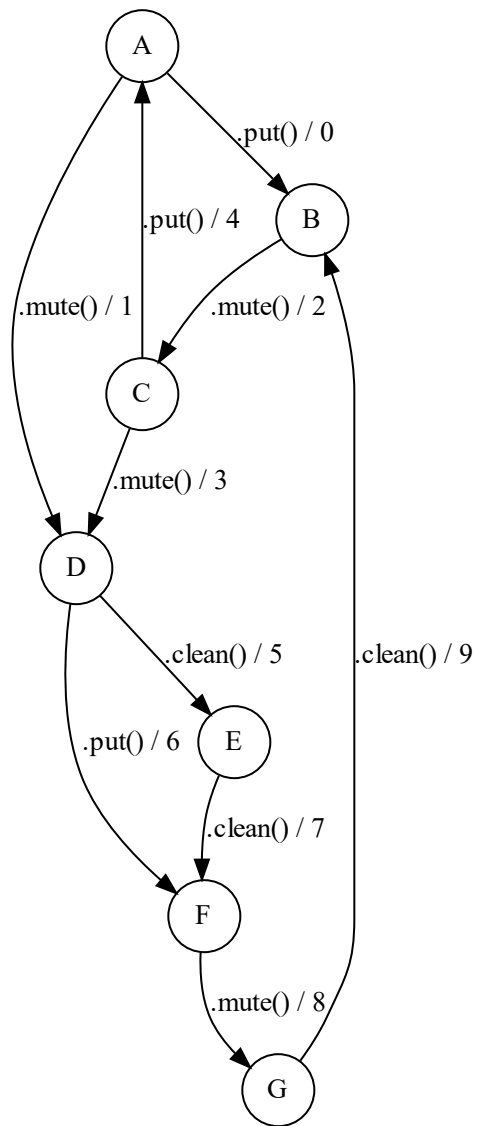
## 2. Двоичные данные:

```
(b'*ZQI\nK\x00\xb82\xa2\xf9\x02\x9e\xc2A?~\x1b\x06\x00U\x00\x00\x00N\x86\xfao'
b'\x12L?\x1d\x0b\x06\x00[\x00\x00\x00\xa8\x15\x00\xe6\x1d\x0b\xbf\xef\xa2'
b'\x05\x00a\x00\x00\x00\x97\xb6\xf5\x1ayK\xbf\xdf\xaf\x04\x00f\x00\x00'
b'\x00\x86m\xa2\x8bck\x14\x0c\x05m\x02\x00\x00\x00I\x00\xeb\x9a\x93'
b'F\xa5\x85\xd5+\xad\xe1.9j\x89\x8f\xc6\x8f:\xfa\xf8\x8c')
```

## Результат разбора:

```
{'A1': {'B1': 10, 'B2': {'C1': 1829047316, 'C2': 'ck'}},
  'A2': 184,
  'A3': 50,
  'A4': [{'D1': [-94, -7, 2],
    'D2': 0.7568758726119995,
    'D3': 7038,
    'D4': [-21, -102, -109, 70, -91, -123]},
  {'D1': [78, -122, -6],
    'D2': 0.7971562743186951,
    'D3': 2845,
    'D4': [-43, 43, -83, -31, 46, 57]},
  {'D1': [-88, 21, 0],
    'D2': -0.5434249639511108,
    'D3': -23825,
    'D4': [106, -119, -113, -58, -113]},
  {'D1': [-105, -74, -11],
    'D2': -0.7948166131973267,
    'D3': -20513,
    'D4': [58, -6, -8, -116]}]},
'A5': 2342677894}
```

**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.mute()</code>	1
<code>o.put()</code>	6
<code>o.mute()</code>	8
<code>o.clean()</code>	9
<code>o.mute()</code>	2
<code>o.mute()</code>	3
<code>o.mute()</code>	RuntimeError
<code>o.clean()</code>	5
<code>o.clean()</code>	7
<code>o.mute()</code>	8
<code>o.clean()</code>	9
<code>o.put()</code>	RuntimeError
<code>o.mute()</code>	2
<code>o.put()</code>	4
<code>o.put()</code>	0
<code>o.mute()</code>	2

## 2. Пример использования класса C32:

<code>o = C32()</code>	
<code>o.put()</code>	0
<code>o.mute()</code>	2
<code>o.mute()</code>	3
<code>o.mute()</code>	RuntimeError
<code>o.clean()</code>	5
<code>o.put()</code>	RuntimeError
<code>o.clean()</code>	7
<code>o.mute()</code>	8
<code>o.clean()</code>	9
<code>o.mute()</code>	2
<code>o.put()</code>	4
<code>o.mute()</code>	1
<code>o.put()</code>	6



## Вариант №10

**Задача 3.1.** Реализовать разбор двоичного формата данных (в духе формата WAD игры Doom или графического формата PNG). Данные начинаются с сигнатуры 0x43 0x44 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	Структура B
	2	Размер (uint32) и адрес (uint32) массива структур C
	3	int32
	4	float
	5	uint64
	6	Размер (uint32) и адрес (uint16) массива int16

Структура B:	1	int64
	2	int16
	3	Массив char, размер 7
	4	float
	5	uint8
	6	int64
	7	float

Структура C:	1	Массив double, размер 4
	2	Адрес (uint32) структуры D

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

Примеры разбора двоичного формата с помощью функции f31:

### 1. Двоичные данные:

```
(b'CDG\xe2\xc9\x17\x00Q\x8a\x0e\xe5\xc1\xae\v\bmdfi\xbe\xe1\xd0\xe7\x98\xc2(\x85'
b'\xb9\xc1\xe1\xeen>\xa4\xf3\x15\x00\x00\x00\x02\x00\x00\x00[(/\xef'
b'\xbf\xbfX\xded\xf1Y\x10\xf4\n\xe8e\xf\x00\x00\x00\x02\x00\xa3\x9a'
b'\xd3\x10\xc2\xf5\xf6\x00\x02\x00\x00\x00C\xbc\xf8\x8c\x88\x81'
b'\x18\x00\x02\x00\x00\x000\xbf\xe1\xe3\xef\xd2\x1fBH?\xe1/\xb1\xff1\xa7N?'
b'\xa2n3(\x89\x94\xe0\xbf\xe6\x13\xad\xc1\xff\x14\x98\x00\x00\x00E?\xdbur\xe3'
b"\x0c@L?\xe4s\xe4: '2<?\xc8\x16\xc0i\xa1\xb8\xc0?\xaca;\xf4\x95\xc6 \x00"
b'\x00\x00Q\xb0\xe6\xf7xaa')
```

Результат разбора:

```
{'A1': {'B1': -2105126060683555099,
        'B2': -15954,
        'B3': 'v\bmdfi',
```

```

'B4': -0.44104692339897156,
'B5': 152,
'B6': -4456164798414852498,
'B7': 0.3221670687198639},
'A2': [{'C1': [-0.5590743164012304,
               0.537072180200292,
               0.03599700803605699,
               -0.6899021901123374],
        'C2': {'D1': 281212406, 'D2': [154, 211]}}},
{'C1': [0.4290435044540615,
        0.6391469131281791,
        0.18819432409094894,
        0.055429338079999946],
 'C2': {'D1': 2357756184, 'D2': [188, 248]}}},
'A3': 674230207,
'A4': -0.8471434116363525,
'A5': 17390950076408947983,
'A6': [-20250, -2134]}

```

## 2. Двоичные данные:

```

(b'CDG\xc7\xa3"\xc0\xba\xda\x03^\x04bsqngbo\xbf;\x0f\x8e\x10c\x96\xe7'
 b'[\xe7#\x1d\x89=\x14\x1ai\x00\x00\x00\x02\x00\x00\x00\\\x8f\xb4\xce-?M\xb9'
 b'\x94\xe1\xd5\xb2\xeb\xdf#\x9b\xbd\x00\x00\x00\x04\x00\xa4\xb1\x00\x0b}\x15'
 b'\x1b\x97\x00\x03\x00\x00\x00C%6\xac\xd3\xc5*\x00\x02\x00\x00\x00P?\xe6I\xaf'
 b'[q\\\x8c\xbf\x19\t\xc5\x95\xbf\x00?\xd5\xe3\x00\xe0\x89\xa8\xa0?\xe3\x1b2'
 b'}\x0f\xc2\n\x00\x00\x00F\xbf\xe4\xc0\xd9\x85r\xa8\xc8\xbf\xa9\xdc\xc2'
 b'\x17\xa2\xeb\xc0?\xe6F\x0f!\xe9j^\xc2\x98_\xa8\xd5NX\x00\x00\x00RR\xc5\xc79'
 b'\xe6\xee=6')

```

## Результат разбора:

```

{'A1': {'B1': -4061361607765468669,
        'B2': 24068,
        'B3': 'bsqngbo',
        'B4': -0.7307060956954956,
        'B5': 16,
        'B6': 7176177438167342473,
        'B7': 0.0361579991877079},
 'A2': [{'C1': [0.6964947496407077,
                 -0.004662550135762222,
                 0.3419801895853798,
                 0.5970699732407081],
        'C2': {'D1': 2098535319, 'D2': [177, 0, 11]}}},
{'C1': [-0.6485412222293911,
        -0.05051237619905846,

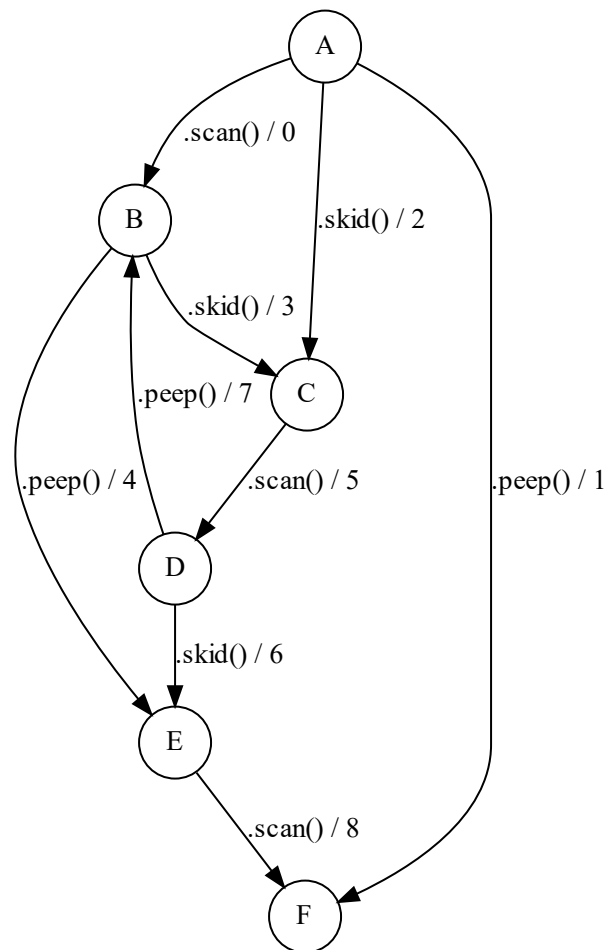
```

```

0.6960521375979842,
0.14527506140676638],
'C2': {'D1': 2899559722, 'D2': [37, 54]}},
'A3': -1883976147,
'A4': 0.8036129474639893,
'A5': 16273109554703866813,
'A6': [21189, -14535, -6418, 15670]}

```

**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.skid()</code>	2
<code>o.scan()</code>	5
<code>o.scan()</code>	RuntimeError
<code>o.peep()</code>	7
<code>o.skid()</code>	3
<code>o.scan()</code>	5
<code>o.skid()</code>	6
<code>o.scan()</code>	8

## 2. Пример использования класса C32:

<code>o = C32()</code>	
<code>o.scan()</code>	0
<code>o.skid()</code>	3
<code>o.skid()</code>	RuntimeError
<code>o.scan()</code>	5
<code>o.peep()</code>	7
<code>o.scan()</code>	RuntimeError
<code>o.peep()</code>	4
<code>o.scan()</code>	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 0x3e 0x42 0x51 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 3
	2	uint8
	3	double
	4	Массив адресов (uint16) структур B, размер 2
	5	Структура D
	6	uint16
	7	float
	8	uint32

Структура B:	1	Структура C
	2	uint8

Структура C:	1	float
	2	int64

Структура D:	1	Массив int32, размер 8
	2	Массив int16, размер 6
	3	uint8
	4	uint32
	5	int8
	6	int8

Примеры разбора двоичного формата с помощью функции f31:

1. Двоичные данные:

```
(b'>BQFotvi\x12\xf1\xda\xc1\x0b\xac\xec?Q\x00^\x00^\x81w\t\x9d9\x01\x9d'
b'F\xe2\xa2\xe77\xe2\x94\x9aB2{\x14\x9a\xde!\x02\xe3T\xdf\x84\x9f\xf7\xbc\xec'
b'\xad\xe0\xda\x90\x08\xa3\xb1\x1f\x93E\x1c\x079\x8e\xe4\x18Q\xd6\xff\xb7'
b'\xdf\xca\x91\x06\xbf\xd9\xe8\xd5$\xe1\xb4S\xbe\xd8\x1fT\xa6\x19\xdc\xe'
b'\xb3\xeb\x8f\xea\x0f=>\xec\xee-e\xb86\x16\x99')
```

Результат разбора:

```
{'A1': 'otv',
'A2': 105,
'A3': 0.8960017000422644,
'A4': [{'B1': {'C1': -0.20674468576908112, 'C2': -5508223291504713768},
'B2': 235},
```

```
{'B1': {'C1': 0.03513580188155174, 'C2': 1600669462278630462},
  'B2': 153}},
'A5': {'D1': [158826846,
  -1660864099,
  -408755642,
  -1701518793,
  343618114,
  35774106,
  -2065738525,
  -323160161],
  'D2': [-8019, -28454, -23800, 8113, 17811, 1820],
  'D3': 57,
  'D4': 1360585870,
  'D5': -42,
  'D6': -1},
'A6': 57271,
'A7': -0.5256620645523071,
'A8': 617998553}
```

## 2. Двоичные данные:

```
(b'>BQFtay{\x82F\xda\x95\x96\xd4\xef?Q\x00^\x00\x8d\xfe\x0cv\xacX\x15\xa9'
b'[\xe5y\xe1\xe3\xa3\x92\xa5\xa7\xd7\r1\xfb3"\xd4T!As\' \xa30\x90P\x1f\x82'
b'\xcb\x13\x9f\xa3p9\xc2\xd8\x9c\x0f\xd7\x0bo\xda\xfb7U\xbf\x86\x87K\xbfjv<'
b'M'\xf3\xe7>\x99xB\x14\xc6\xbd\x93{e\xfb1%\xac\xbe\xccG\x0e\x9a|\xc8\xef?#')

```

## Результат разбора:

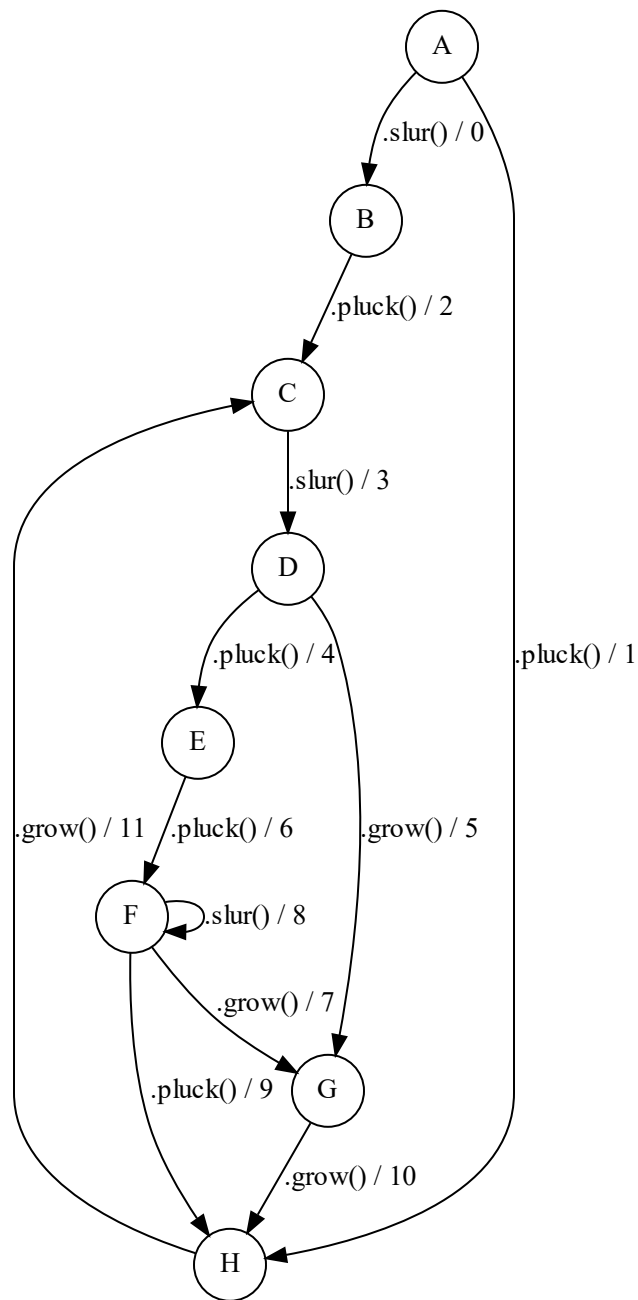
```
{'A1': 'tay',
  'A2': 123,
  'A3': 0.9947007109460204,
  'A4': [{'B1': {'C1': 0.45302867889404297, 'C2': 8904669546682677401},
    'B2': 101},
    {'B1': {'C1': -0.3362269699573517, 'C2': 4607121381309433804},
    'B2': 35}],
  'A5': {'D1': [1980563085,
    -1458218836,
    -512105125,
    -1517116445,
    -681269439,
    586378253,
    1092703444,
    815998835],
    'D2': [20624, -32225, 5067, -23649, 14704, -10046],
    'D3': 156,
    'D4': 1863046927,
```

```

'D5': -38,
'D6': -9},
'A6': 48981,
'A7': -0.7950366735458374,
'A8': 1295808106}

```

**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.grow()	RuntimeError
o.slur()	0
o.pluck()	2
o.slur()	3
o.slur()	RuntimeError
o.pluck()	4
o.pluck()	6
o.slur()	8
o.pluck()	9
o.grow()	11
o.slur()	3
o.pluck()	4
o.pluck()	6
o.grow()	7
o.grow()	10
o.grow()	11
o.slur()	3
o.slur()	RuntimeError
o.grow()	5

## 2. Пример использования класса C32:

o = C32()	
o.slur()	0
o.pluck()	2
o.slur()	3
o.grow()	5
o.grow()	10
o.grow()	11
o.slur()	3
o.slur()	RuntimeError
o.pluck()	4
o.pluck()	6
o.slur()	8
o.pluck()	9
o.grow()	11
o.slur()	3
o.pluck()	4
o.slur()	RuntimeError
o.pluck()	6
o.grow()	7



## Вариант №12

**Задача 3.1.** Реализовать разбор двоичного формата данных (в духе формата WAD игры Doom или графического формата PNG). Данные начинаются с сигнатуры 0x47 0x52 0x4c 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	uint64
	3	int8
	4	Массив uint8, размер 6
	5	float
	6	double
	7	Адрес (uint32) структуры D

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

Структура C:	1	Размер (uint32) и адрес (uint32) массива int16
	2	Массив int16, размер 3

Структура D:	1	Массив int32, размер 3
	2	int32

Примеры разбора двоичного формата с помощью функции f31:

1. Двоичные данные:

```
(b'GRLX\xd2\xf1)\xf0\x89B\xe0?\xbchA\x00\x00\x00W\x00\x00\x00m\x00'
b'\x00\x00\x83\x00\x00\x00\xca\xb9\xd63\xe5+\xff\x03:\xdd:D\x7f.\x08\r\x91'
b'>\x88\xcbt\xd9\xa0\xeb\xca?\x91\x00\x00\x00\xfc\xcb\x82\x12\x02\x00\x00'
b'\x00=\x00\x00\x00~\xa5\x15\x86z\xd6\xd7\xcb\xf5L\xa35\x94g\x04\x00\x00\x00'
b'\x00\x00\x00\x1bN\xf7\xb9$\x19\x8aq\x80\x1d\x805\x82\xf3\x04\x00\x00'
b'\x00e\x00\x00\x00~\x0fe\x94\xbc\xa88\x86\x81\x13\x83\x84PC\x04\x00\x00\x00{'
b"\x00\x00\x00\xec\xbaK\xf8'\x12\xce|\x11\x8c\x02\x80`W\x84t\xc6o\xa5\xb2\xcd"
b'\xc1')
```

Результат разбора:

```
{ 'A1': { 'B1': 0.5081224146556969,
          'B2': 26812,
          'B3': [{ 'C1': [-13316, 4738], 'C2': [-23170, -31211, -10630]},
                  { 'C1': [-13353, 19701, 13731, 26516],
                    'C2': [19995, -17929, 6436]},
                  { 'C1': [29066, 7552, 21376, -3198],
                    'C2': [3966, -27547, -22340]}],
  }
```

```

        {'C1': [-31176, 4993, -31613, 17232],
         'C2': [-17684, -1973, 4647]}]},
'A2': 18387041752832195018,
'A3': 3,
'A4': [58, 221, 58, 68, 127, 46],
'A5': 0.28330254554748535,
'A6': 0.21031580560665142,
'A7': {'D1': [-1945010994, 1465942018, 1875276932], 'D2': -1043483995}}

```

## 2. Двоичные данные:

```

(b'GRLX\x80}\xbfPW\xe1\xe1?5\xa1G\x00\x00\x00[\x00\x00\x00q\x00\x00\x00\x85\x00'
 b"\x00\x00>\x03\xbf\x91y<?\x0bAb\x00d3fa\x80}\xc4\xbe\xe0\x02\xba'^W\xb3"
 b'? \x93\x00\x00\x00\xf2\x94/xp\xb05~\xe6\xfd\x05\x00\x00\x00=\x00\x00\x00\xae'
 b"\x8d\xb2\xc6\x88\xbfX\xc f*'\{ \xa4\x03\x00\x00\x00U\x00\x00\x00\xf3"
 b'W\x14\x18\x9c%\xeaTLJ\xc9\x97\xec\x f4\x04\x00\x00\x00i\x00\x00'
 b'\x00\xf1\xd9\x9aJ\x89v\xc9\xc1\x9a=\t\xe9\x03\x00\x00\x00\x7f\x00\x00'
 b'\x00qu\x98\x1a\xa3\t\x87\x90tU 1\xd4\x9e\xef\xdc=\xc9\x92w=\xcc')

```

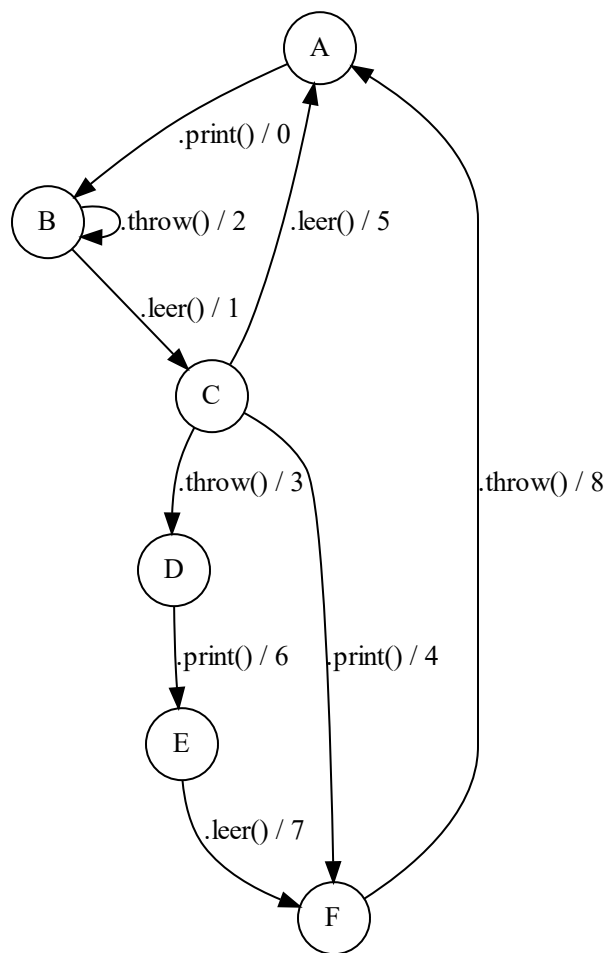
## Результат разбора:

```

{'A1': {'B1': 0.5587574555501504,
        'B2': 41269,
        'B3': [{'C1': [-27406, 30767, -20368, 32339, -538],
                  'C2': [-29266, -14670, -16504]}],
         {'C1': [-12456, 10026, -23429], 'C2': [22515, 6164, 9628]},
         {'C1': [21738, 19020, -26679, -2836],
          'C2': [-9743, 19098, 30345]},
         {'C1': [-15927, 15770, -5879], 'C2': [30065, 6808, 2467]}]},
'A2': 810432950783902526,
'A3': 65,
'A4': [98, 0, 100, 51, 102, 97],
'A5': -0.3837699890136719,
'A6': 0.07555187673461772,
'A7': {'D1': [1433702535, -1630245856, -918692625], 'D2': -868386926}}

```

**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.print()           0
o.leer()            1
o.leer()            5
o.print()           0
o.throw()           2
o.leer()            1
o.throw()           3
o.print()           6
o.leer()            7
o.throw()           8
o.print()           0
  
```

2. Пример использования класса C32:

o = C32()	
o.print()	0
o.throw()	2
o.throw()	2
o.leer()	1
o.print()	4
o.throw()	8
o.print()	0
o.throw()	2
o.leer()	1
o.throw()	3
o.print()	6
o.leer()	7

## Вариант №13

**Задача 3.1.** Реализовать разбор двоичного формата данных (в духе формата WAD игры Doom или графического формата PNG). Данные начинаются с сигнатуры 0xdf 0x53 0x55 0x51 0x44, 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	Структура Е

Структура В:	1	float
	2	int64
	3	Размер (uint16) и адрес (uint16) массива адресов (uint16) структур С
	4	int32
	5	Размер (uint16) и адрес (uint32) массива int8
	6	int32
	7	int8

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

Структура D:	1	int16
	2	uint8

Структура Е:	1	Размер (uint16) и адрес (uint32) массива uint8
	2	int8
	3	int16
	4	double

Примеры разбора двоичного формата с помощью функции f31:

1. Двоичные данные:

```
(b'\xdfSUQD1\x00\x00\x00\x04\x00\x8b\x00\x00\x00\x04\xcch\xd0\x00\x11\x8b\x8dr'  
b'\xc8\xbf\xd0z\x8aD\x9ckg\x921\xb4\x1a\x00\x00\x00$t\tda\xe4\x0e\x1f\x80\xb9'  
b'Y\xef\xb9xj}\nH\x93\xd8\xc4Z\x05\x00\x1d\x00\x18/\xba\x9c-h\x1c\x00'  
b'\xd6\x07@\x00\x00\x00\x92\da\x16\x0f\x81\xea\xb7\x1d\xf3B\xff\xffV\xd0'  
b'\xde\xb2\xc9G\x9eJ\x05\x00C\x00"\x00H\x00\xc9\x93\xff\x99)\xbe|ks\xe4'  
b'V\xb0\x07\xc4\x02\x00f\x00\xcf\xe3\x94\\\x02\x00j\x00\x00\x00\x89\x0b'  
b'\x0b%\x8e@z)\x88')
```

Результат разбора:

```
{'A1': {'B1': -0.16562651097774506,
        'B2': -4321291430192256132,
        'B3': [{'C1': 46129,
                  'C2': {'D1': 31440, 'D2': 138},
                  'C3': [36, 9, -38, -28, 14, 31, -128],
                  'C4': 185,
                  'C5': 5191099360908799833,
                  'C6': 1522849939,
                  'C7': [68, 156, 107, 103, 146]},
                {'C1': 2006,
                  'C2': {'D1': 12056, 'D2': 186},
                  'C3': [-110, -38, 22, 15, -127, -22, -73],
                  'C4': 29,
                  'C5': -5557775818047929613,
                  'C6': 1570654153,
                  'C7': [156, 45, 104, 28, 0]}]},
        'B4': 1553261519,
        'B5': [-55, -109],
        'B6': 621480841,
        'B7': -114},
{'A2': {'E1': [64, 122, 41, 136],
        'E2': 4,
        'E3': 26828,
        'E4': -0.1909958771739242}}
```

## 2. Двоичные данные:

```
(b'\xdfSUQDk\x00\x00\x00\x04\x00\x8a\x00\x00\x00j\xd2\xda\xd0;\xaa\xb9\xb6%'
b'\xcd?\xe1\xb2\x07E\x90\x88\xe9H\x1a\x00\x00\x00T}1`\xbe\x81\xaei\xd83.0\x18W'
b'\x08\x1e\xc2\xa5\xda\x03\x00\x1d\x00J\x1f\xb3\xdc\n}\xbf\xc3a\xa3'
b'>\x00\x00\x00\x06\xc2\xe6\xb4\xc1\xe0\xa1\xba\x1f5\x88t&\xe1c\x82'
b'\x8a:\x84\xe0\x05\x00A\x00 \x00F\x00\xe3>\xce\xdd\x04w\xbe\xe7\xd1M\xe7\x00'
b'\x04\xb3c\x02\x00d\x00oq\xa5/\x03\x00h\x00\x00\x00cc&\xac\x82%\x83\xfc\')
```

## Результат разбора:

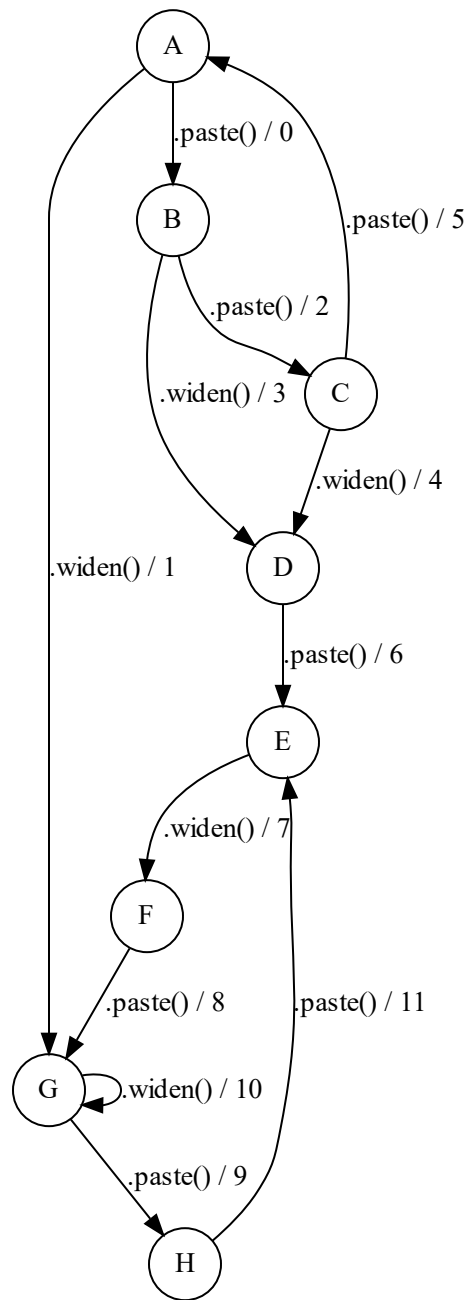
```
{'A1': {'B1': -0.24122948944568634,
        'B2': 7184090232513221095,
        'B3': [{'C1': 18665,
                  'C2': {'D1': -19743, 'D2': 7},
                  'C3': [84, 125, 49, 96, -66, -127, -82],
                  'C4': 105,
                  'C5': 2164075382350689240,
                  'C6': 2010817986,
                  'C7': [69, 144, 136]},
                {'C1': 41825,
```

```

        'C2': {'D1': 8010, 'D2': 179},
        'C3': [6, -62, -26, -76, -63, -32, -95],
        'C4': 186,
        'C5': -9051143270804474593,
        'C6': -528205174,
        'C7': [220, 10, 125, 191, 195]}],
    'B4': 799371631,
    'B5': [-29, 62, -50],
    'B6': -1406770333,
    'B7': -126},
    'A2': {'E1': [37, 131, 252, 97],
        'E2': 106,
        'E3': -9518,
        'E4': 0.22771343293831459}}

```

**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.paste()	0
o.paste()	2
o.paste()	5
o.paste()	0
o.widen()	3
o.paste()	6
o.widen()	7
o.paste()	8
o.widen()	10
o.paste()	9
o.paste()	11
o.widen()	7
o.paste()	8
o.widen()	10
o.paste()	9

## 2. Пример использования класса C32:

o = C32()	
o.paste()	0
o.paste()	2
o.paste()	5
o.paste()	0
o.paste()	2
o.widen()	4
o.paste()	6
o.widen()	7
o.paste()	8
o.widen()	10
o.paste()	9
o.widen()	RuntimeError
o.paste()	11
o.paste()	RuntimeError
o.widen()	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 0x4a 0x46 0x57 0x4e, за которой следует структура A. Порядок байт: от старшего к младшему. Адреса указаны в виде смещений от начала данных. В решении разрешено использовать модуль struct.

Структура A:	1	Массив uint32, размер 4
	2	Структура B
	3	uint8
	4	Структура E

Структура B:	1	Массив структур C, размер 2
	2	int16
	3	int64
	4	int16
	5	uint64
	6	Структура D
	7	uint64
	8	uint64

Структура C:	1	uint16
	2	Массив uint8, размер 4

Структура D:	1	uint64
	2	uint16
	3	Массив int64, размер 4

Структура E:	1	uint16
	2	Массив float, размер 3

Примеры разбора двоичного формата с помощью функции f31:

1. Двоичные данные:

```
(b'JFWN\xf5\x17\x99\xc7\xf6\xfa\xcc\xd3\xfb\xc4E!\xc6[\xe2\x12\xef\xc8'
b'\xc0\xf0e?\x13\xce\x889\xb1&+\x00rE\\\xd8\x12\xb3M\x10\xe0Y\xec\x90'
b'\xb40\xcd<\x07\xa4>\x95=\xb3#\x80\x9eJ\xb4\x88}\x1d.\xdb\xb9\\\xda\x8a'
b'\x1b=X\xb8\x9a\x93\xf3j\r*\xa7\xe9$\x18\x93a\xacs\x99\xc4\xb5p\xbb:'
b'v\xb8\x8dX\x94\x80\xf0?\xe2\x04\xf2\xa6\x9e"\xe7~\xf7\xbfU9Z>\xfd\xf1'
b'\xde?\x1f\xd8\xf6')
```

Результат разбора:

```
{'A1': [4111964615, 4143631571, 4223944044, 4162963035],
'A2': {'B1': [{'C1': 57874, 'C2': [239, 200, 192, 240]},
```

```

        {'C1': 25919, 'C2': [19, 206, 136, 57]}],
'B2': -20186,
'B3': 3098491135192470195,
'B4': 19728,
'B5': 16166212443667680572,
'B6': {'D1': 550633865152242560,
        'D2': 40522,
        'D3': [-5437958985756198564,
                -2699315076359742829,
                -906897898104871912,
                -7826785064892189328]}},
'B7': 13491226168759194752,
'B8': 17311804003514228258},
'A3': 231,
'A4': {'E1': 32503,
        'E2': [-0.8329063653945923, 0.4959859251976013, 0.6244043111801147]}}

```

## 2. Двоичные данные:

```

(b'JFWNg^xf6\x03\x9a\x05D(u\xd4\x0eJ\x97\xbeUiUfloe\xea\xc5\x14\xacWx`'
b'\x01\xfe\xbf\xe5\xf7\xcb\xb3\xd2dE\xb92\x0c\xa8/\xf0\xf6.\n\xa8\x0c\xd8\xfdw'
b'\xb7\xaep0G)_\xd1\x9c\xca\xe9\xfe\x98RT\xed\xe3\xe6D\xccI^\x8c\xd2'
b'\x82\xfe\xbe\xd4\xfd\xfaI\xa3\x04\x87\x94C\x85|\xdd\xdb\x85S\xe8'
b'\xd4\xef\x11\xe3\xee*\x96\x92\x96\xaf+\xbeX?C!\x08=\xc5g\xae?\x1f\n#')

```

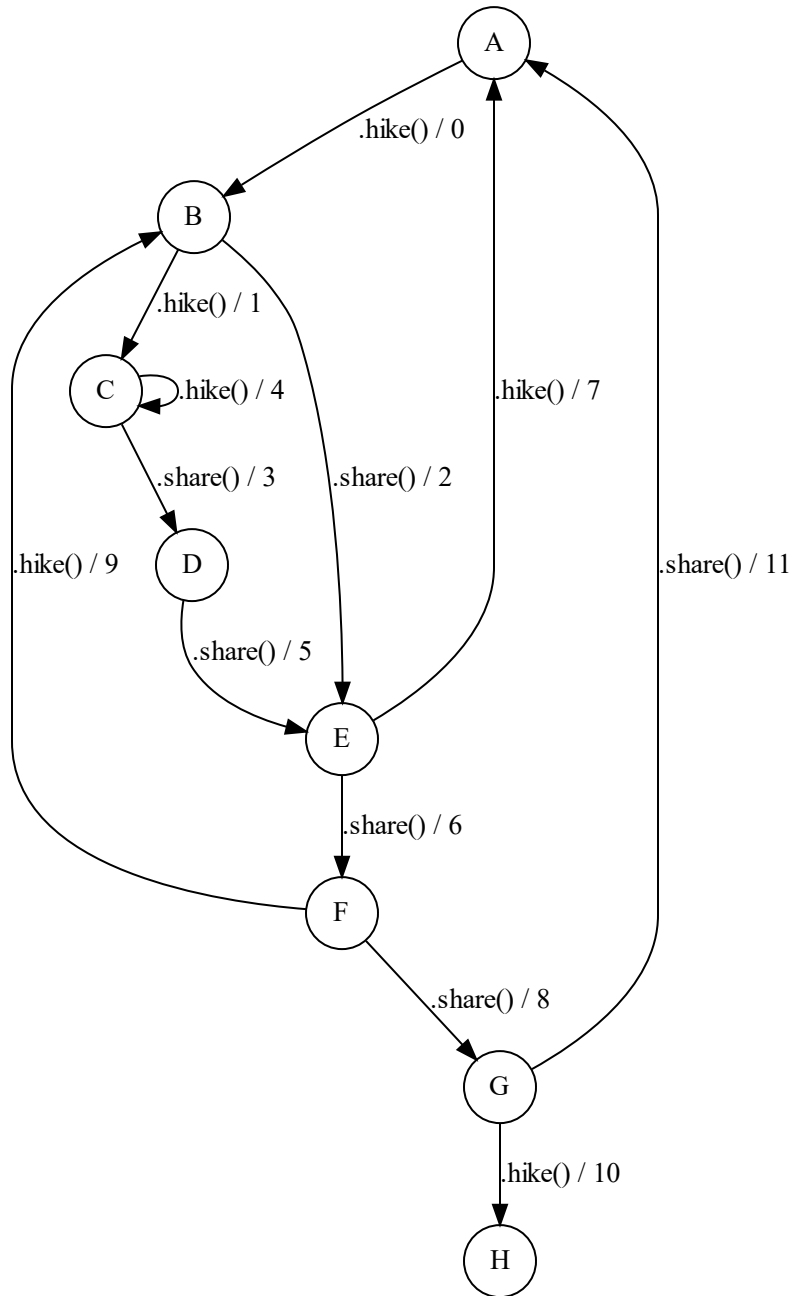
## Результат разбора:

```

{'A1': [1734276611, 2584036392, 1976831562, 2545833321],
'A2': {'B1': [{'C1': 21862, 'C2': [108, 111, 101, 234]},
              {'C1': 50452, 'C2': [172, 87, 120, 96]}],
        'B2': 510,
        'B3': -4619013388531243963,
        'B4': -18126,
        'B5': 912031636511394472,
        'B6': {'D1': 925768414049235023,
                'D2': 18217,
                'D3': [6904472099058784338,
                        6119798046802594142,
                        -8299427132655469215,
                        -843964215305419899]}},
        'B7': 8997588996241020143,
        'B8': 1289135785040320175},
'A3': 43,
'A4': {'E1': 48728,
        'E2': [0.7622227668762207, 0.09638915956020355, 0.6212484240531921]}}

```

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

## 2. Пример использования класса C32:

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

## Вариант №15

**Задача 3.1.** Реализовать разбор двоичного формата данных (в духе формата WAD игры Doom или графического формата PNG). Данные начинаются с сигнатуры 0x53 0x46 0x4e 0x42 0xff, 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	int16
	2	int32
	3	Структура B
	4	Размер (uint32) и адрес (uint32) массива структур D
	5	Массив float, размер 3
	6	int16
	7	Массив char, размер 8
	8	Адрес (uint32) структуры E

Структура B:	1	double
	2	int16
	3	uint64
	4	Структура C
	5	double
	6	int16
	7	double

Структура C:	1	uint64
	2	uint32
	3	int64
	4	uint64

Структура D:	1	int8
	2	int32

Структура E:	1	uint64
	2	int32
	3	uint64
	4	Массив int8, размер 7
	5	uint64
	6	uint16
	7	int64

Примеры разбора двоичного формата с помощью функции f31:

1. Двоичные данные:

```
(b"SFNB\xff}\x0c?|\x91|\xbf\xee'\xcc,A\xae\xea\xfe\x10r6ih}\xa4\xfb"
b'\n\xad\xfe\x9e\xbd\x08\xca\xa0\x019\xf8+P\xe6\x16Xf\xda.\x0f\xbc\xc7|'
```

```

b'\x06\xd9\x8d\x96v\xbf\xefo)\xfc\xa7e1_g?\xe4@\x98\xdf\x10\x15\xca\x00'
b'\x00\x00\x03\x00\x00\x00m?S#\x15>Z\xf3*\xbf\x15\xbaT0~dbbepopt\x00\x00\x00'
b'| \xce\x89\x89\xe9;\xb27\xaa:U)\x97k@\x84\xab>k\xa3$\xa7\xfc`\xb3\xf0\xe5y'
b'\x89D\xf4\x80rK:\xd3\n\xf6\xeb\xeeV\x88\xc2\xdf\x87>\xd3Z\xec\xfd\xe3'
b'\x80\xc1P\xb5I\xf2T{k'})

```

Результат разбора:

```

{'A1': 32012,
 'A2': 1065128316,
 'A3': {'B1': -0.9423580994020686,
        'B2': -496,
        'B3': 8229881266571639562,
        'B4': {'C1': 12537632654050773664,
                 'C2': 20576299,
                 'C3': 5829371336616193551,
                 'C4': 13602977568357455478},
        'B5': -0.9823198256155314,
        'B6': 24423,
        'B7': 0.6328853947349498},
 'A4': [{'D1': -50, 'D2': -1987450565},
         {'D1': -78, 'D2': 933902933},
         {'D1': 41, 'D2': -1754578812}],
 'A5': [0.8247540593147278, 0.21381822228431702, -0.5848743915557861],
 'A6': 12414,
 'A7': 'dbbepopt',
 'A8': {'E1': 12339418377480567904,
        'E2': -1276058247,
        'E3': 9891299514122975955,
        'E4': [10, -10, -21, 114, -18, 86, -120],
        'E5': 14042090867068169469,
        'E6': 58240,
        'E7': -4516911097049744533}}

```

## 2. Двоичные данные:

```

(b'SFNB\xff\x08\xe4\x84;"\x04\xbf\xee\xe2\x96fw\x93d\x8c)\x84\x1a~'
b'\x94\xe1\xa4\x87#\x00\xf79t\x16\x04&\xd8\xfd\xd4\x06\x07U\xd3\xe0'
b'\x10>\x8f\xc7a,\x1bd>y\xe8\xdfP\xbf\xc4X\xc7i\xd4\xccx"\xc9?\xe5\xde\x8fA'
b"'\xc00\xc6\x00\x00\x03\x00\x00\x00m?' '\x0c?\# \x1f\x18\xbe\xcc\xff\xf4$"
b'\xfafaliayja\x00\x00\x00|\x8c~\xd4\xa1<\xdc\x9d\xd9x7\xb7\x85I\xbfN'
b'a\xdc\xda^\x0f\x86[p\x0f^p?Z\xed\x17x7\xf0\x11\xcf@\xd4\xbar\x04\x0f\xdd\x02'
b'P1}\x04\x98\xc8\x1b\xa0UN\xb1\xfe\xf73\x06\x17\xaa')

```

Результат разбора:

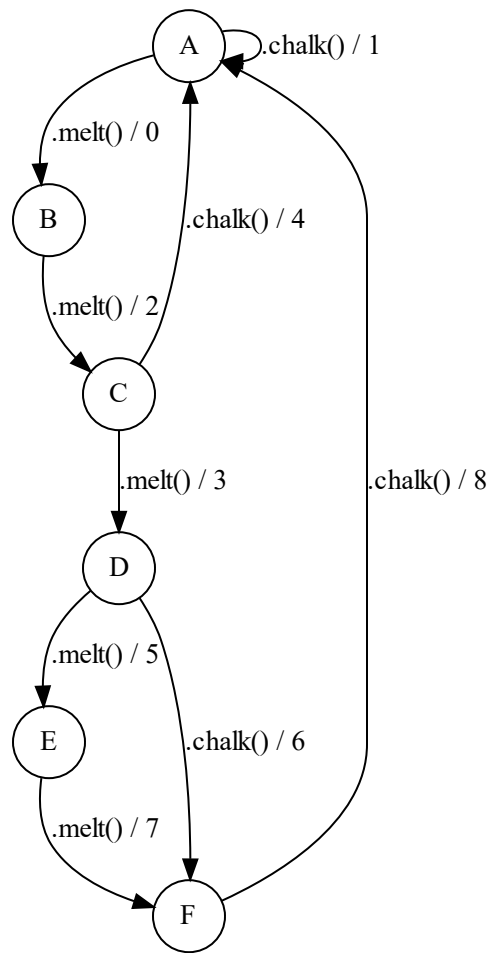
```

{'A1': 2276,
 'A2': -2076499452,
 'A3': {'B1': -0.9651596070585637,
        'B2': -29655,
        'B3': 9519059940306880291,
        'B4': {'C1': 69587489995892440,
                'C2': 4258530823,
                'C3': 6184533073683531617,
                'C4': 3178244181536071504},
        'B5': -0.15895931880513436,
        'B6': 8905,
        'B7': 0.6834179188808995},
 'A4': [{'D1': -116, 'D2': 2127864124},
        {'D1': -36, 'D2': -1646692297},
        {'D1': -73, 'D2': -2058764466}],
 'A5': [0.6529395580291748, 0.6371932029724121, -0.40039026737213135],
 'A6': 9466,
 'A7': 'faliayja',
 'A8': {'E1': 7051751214077598576,
        'E2': -262246337,
        'E3': 6551918837995934159,
        'E4': [64, -44, -70, 114, 4, 15, -35],
        'E5': 166687599230502939,
        'E6': 41045,
        'E7': 5670593743502579626}}

```

**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.chalk()           1
o.chalk()           1
o.melt()            0
o.melt()            2
o.chalk()           4
o.chalk()           1
o.melt()            0
o.melt()            2
o.melt()            3
o.melt()            5
o.melt()            7
o.chalk()           8

```

2. Пример использования класса C32:

o = C32()	
o.melt()	0
o.melt()	2
o.melt()	3
o.melt()	5
o.melt()	7
o.chalk()	8
o.melt()	0
o.melt()	2
o.chalk()	4
o.chalk()	1
o.chalk()	1



```
{'A1': [{'B1': 0.33469947691189494,
        'B2': {'C1': -0.0031294687433223167, 'C2': 3461316657},
        'B3': 59,
        'B4': 12744},
        {'B1': -0.9389459285473671,
        'B2': {'C1': 0.11003344545166271, 'C2': 3206041185},
        'B3': 162,
        'B4': 1648}],
'A2': {'D1': 3313,
        'D2': 3612528929,
        'D3': 51,
        'D4': 'ntd',
        'D5': {'E1': -0.606879633476834,
        'E2': 5782244651549030447,
        'E3': [0.27981582283973694, 0.36636823415756226],
        'E4': 93,
        'E5': -1857614192,
        'E6': 14165901149722362667,
        'E7': [104, 185, 40]},
        'D6': [107, -35, -19, 2, 41],
        'D7': -16623}}
```

## 2. Двоичные данные:

```
(b"LBU\x00\x02\x007\x00}\xbf\xbb['\xf12\xe8\x10?\xea\xbf\xe5\xd7\xc4["
b'\x06V\x89\x9f\r\x91\x1a\x8a?\xdc\xf9\xee\xf9-\x87\xf8?\xe4k\xc0>B<\xbc'
b'\x0f\xa2}e=\x02\xe2\x00\x00\x00\t\x00\x00\x00 ?l3\x0e>\xb2\xea\xec7'
b'\xec\x82\xd2\xf6\xef\xbf\xb0\x91\xf7+]\x9c \xf67r\x95\xf6M\xa7'
b'\xd7\x00\x02\x00?\x92\xaf\xa8\xa1Dv1\nIy{\x0e\xa6\x00\x00\x00\x06\x00\x00'
b'\x00G\xaa\x07\x90?\xfa\xfa\xae\x999\x93\x95^\x83ksi\x00\x00\x00M\x00'
b'\x00\x00\x07\x00\x00\x00v\x99^')
```

## Результат разбора:

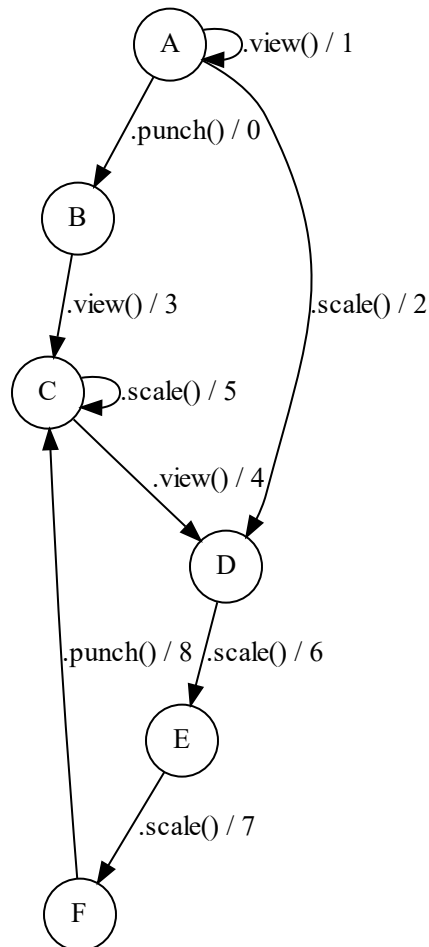
```
{'A1': [{'B1': -0.10685968054431805,
        'B2': {'C1': 0.8359250272941103, 'C2': 1451859725},
        'B3': 145,
        'B4': 6794},
        {'B1': 0.45275472959303853,
        'B2': {'C1': 0.6381531921378136, 'C2': 262307173},
        'B3': 61,
        'B4': 738}],
'A2': {'D1': 44697,
        'D2': 965973342,
        'D3': 131,
        'D4': 'ksi',
```

```

'D5': {'E1': -0.06472725685817737,
      'E2': 17741775245457336279,
      'E3': [0.9226540327072144, 0.349448561668396],
      'E4': 146,
      'E5': -1349869244,
      'E6': 8516599681021316774,
      'E7': [55, 236, 130, 210, 246, 239]},
'D6': [-86, 7, -112, 63, -6, -6, 106],
'D7': -26274}}

```

**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.view()</code>	1
<code>o.punch()</code>	0
<code>o.view()</code>	3
<code>o.scale()</code>	5
<code>o.scale()</code>	5
<code>o.view()</code>	4
<code>o.view()</code>	RuntimeError
<code>o.scale()</code>	6
<code>o.scale()</code>	7
<code>o.punch()</code>	8

## 2. Пример использования класса C32:

<code>o = C32()</code>	
<code>o.view()</code>	1
<code>o.punch()</code>	0
<code>o.view()</code>	3
<code>o.scale()</code>	5
<code>o.view()</code>	4
<code>o.scale()</code>	6
<code>o.scale()</code>	7
<code>o.punch()</code>	8
<code>o.punch()</code>	RuntimeError
<code>o.view()</code>	4

## Вариант №17

**Задача 3.1.** Реализовать разбор двоичного формата данных (в духе формата WAD игры Doom или графического формата PNG). Данные начинаются с сигнатуры 0x51 0x52 0x48 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	float
	2	Массив char, размер 8
	3	Адрес (uint16) структуры B
	4	uint64
	5	Массив адресов (uint32) структур D, размер 4
	6	int32
	7	uint64
	8	Размер (uint16) и адрес (uint16) массива uint16

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

Структура C:	1	int64
	2	int64
	3	float
	4	uint64

Структура D:	1	int16
	2	Массив int8, размер 8

Примеры разбора двоичного формата с помощью функции f31:

1. Двоичные данные:

```
(b'QRHI6\xd7\xcb>jrpornhqV\x00\xca\xb17\xc9\xe4Zk#j\x00\x00\x00t\x00'
b'\x00\x00~\x00\x00\x00\x88\x00\x00\x00\xee\x84\xff\x8b\x01\x9b'
b'\x88\x84\x80\x9e\xc0G\x03\x00\x92\x00\xf3\xf4\xd8\xf5d\xbane\x84\xef'
b'\xd7\xb1\xe4@\x95Y\x995\xcb>g\x7f\xfc\xbf\xd3\x0f\xb9\xf7Ne{\xb8j;'
b'$\x8e\xe3\x1e\x1a\xa6:\x00\x00\x00$\x8e\x17\xc8\xb9\x15:\xc0\x0f"k\xaec\x85'
b'6TB\x8cm\xf9\xd61\x070\x0e+k\x14?DF\xce3\xc8\xee\xa7j\xd9\xb2\xff2e'
b"\x8d\xc4\xfb7'\xf2\xf2\x91")
```

Результат разбора:

```
{'A1': 0.3981263041496277,
'A2': 'jrpornhq',
'A3': {'B1': 3095094606,
```

```

'B2': -6477831151327888534,
'B3': {'C1': 7308984188053026035,
      'C2': 6455136991885913988,
      'C3': 0.39689329266548157,
      'C4': 17850316000193183591},
'B4': 3356986916},
'A4': 2552233552510431690,
'A5': [{'D1': 5561, 'D2': [58, -64, 15, 34, 107, -82, 99, -123]},
      {'D1': 21558, 'D2': [66, -116, 109, -7, -42, 49, 7, 79]},
      {'D1': 11022, 'D2': [107, 20, 63, 68, 70, -50, 51, -56]},
      {'D1': -22546, 'D2': [106, -39, -78, -1, 50, 101, -115, -60]}],
'A6': -1946188562,
'A7': 5170306647037876993,
'A8': [14331, 61991, 37362]}

```

## 2. Двоичные данные:

```

(b'QRHItn\x82\xbemoivdpizV\x00\x10\xca\xc0\x96\xec.s\xf7j\x00\x00\x00t\x00'
 b'\x00\x00~\x00\x00\x00\x88\x00\x00\x00AU;$\xe1\xb6k\x14\x8b\xf6\x8bd\x05\x00'
 b'\x92\x00\x8b\x82\x9bW\n\xb0\x01\x88vz\xa6/H\xf76gUt\x1f>\x00\x9b\xc1\xa1C4'
 b'HA3B{\xf3\xc5\xe0Z\xa2\xd1\xf0\xf4|:\x00\x00\x00E\xe2\xe7\xe5L\xe9\xe0e\x87h_'
 b'\x10\x0f\xb1"rnvG\x98+\x0f\xfa+\x1bY\xa8\x90\xa7\xaf\xd0\\\x812+'
 b'\x95\x0f\xc5\x17\x87\xaf\xaa0<\xd5\xc5<\xb7U\xcc\x80\x8ct9\x7f')

```

## Результат разбора:

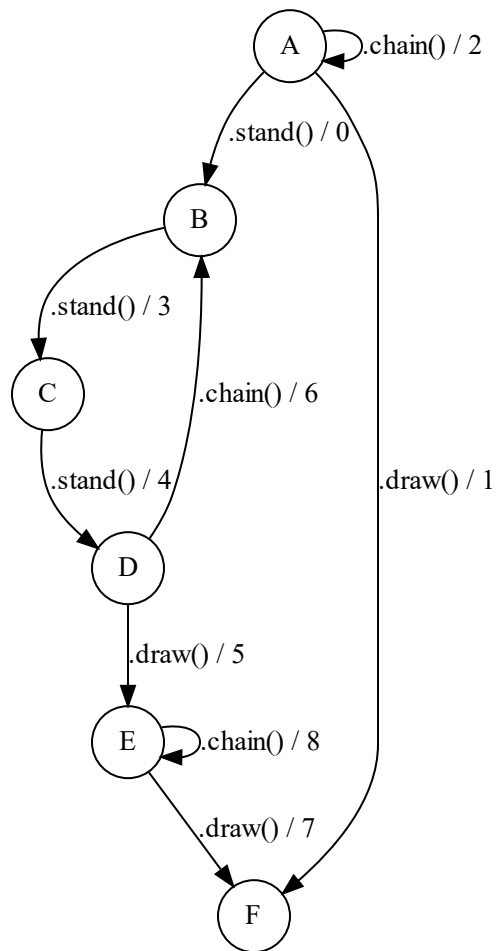
```

{'A1': -0.2547489404678345,
 'A2': 'moivdpiz',
 'A3': {'B1': 4084941363,
      'B2': 9004086338183160005,
      'B3': {'C1': -8646436251108670837,
            'C2': 7437403724058098294,
            'C3': 0.15571720898151398,
            'C4': 4704067275869756160},
      'B4': 3857179205},
 'A4': 17830646943366302224,
 'A5': [{'D1': -25012, 'D2': [14, -121, 104, 95, 16, 15, -79, 34]},
      {'D1': 28274, 'D2': [118, 71, -104, 43, 15, -6, 43, 27]},
      {'D1': -22439, 'D2': [-112, -89, -81, -48, 92, -127, 50, 43]},
      {'D1': 3989, 'D2': [-59, 23, -121, -81, -86, 79, 60, -43]}],
 'A6': 607868225,
 'A7': 7245155502759065313,
 'A8': [15557, 21943, 32972, 29836, 32569]}

```

**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()           2
o.stand()           0
o.stand()           3
o.chain()           RuntimeError
o.stand()           4
o.chain()           6
o.stand()           3
o.stand()           4
o.draw()            5
o.chain()           8
o.draw()            7

```

2. Пример использования класса C32:

o = C32()	
o.chain()	2
o.stand()	0
o.stand()	3
o.stand()	4
o.chain()	6
o.chain()	RuntimeError
o.stand()	3
o.chain()	RuntimeError
o.stand()	4
o.draw()	5
o.chain()	8
o.chain()	8
o.draw()	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 0x44 0x4d 0x45 0x41, 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	uint64
	2	int8
	3	Адрес (uint32) структуры В
	4	Структура D
	5	uint16
	6	uint64
	7	Адрес (uint16) структуры E

Структура В:	1	Размер (uint16) и адрес (uint32) массива char
	2	Размер (uint32) и адрес (uint16) массива адресов (uint16) структур С
	3	uint8
	4	Размер (uint32) и адрес (uint16) массива float
	5	float
	6	uint8
	7	int64
	8	int32

Структура С:	1	int32
	2	uint16

Структура D:	1	float
	2	int16
	3	uint16

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

Примеры разбора двоичного формата с помощью функции f31:

### 1. Двоичные данные:

```
(b'DMEA\xa8\xd0\xdd\x08E\x07K\x9b\xb6\x00\x00\x000?jd5\xf6C0\xb4\x19p\xd0'  
b'zw.\xaf\xd49\x8e\x00\x9fnx\x9dUV\x82\x11\xdaIK\xf3d\xc4\x94\xb7\xb0\xe8*','
```

```

b"7\x00'\x00-\x003\xbe\xb9:\xca\xbf\x03\x9aT\xbfU\xc3\x10\xbd]a\xcf\x00"
b'\x02\x00\x00\x00%\x00\x00\x00\x03\x009f\x00\x00\x00\x04\x00?>\xd3!\x0c\x9f/'
b':q\x1f\x9f=en\xcc\x1b\xb2B\xa4#\xa5.\xd7o\xd6\xfc;\x1d\xdb\xbc0\n#r*'
b'\xe9)\% \xfe\x0d\xdc\x8c\x9f3hlFB\x0c1j\x85\x92S\x16\xae\x170\x1e\xe0'
b'\xf5\x91\x15a\xe4\xd5`\xbbb\x1cf\x0c8hj\x9f2\x88\x9f\xbf\xe2>\x99'
b'\xe5\x0b\xac\x00\x07\x00\x00\x00s\x00\x02\x00\x00\x00\x8f\xad\xc2')

```

## Результат разбора:

```

{'A1': 12164465621115292571,
 'A2': -74,
 'A3': {'B1': 'nx',
        'B2': [{'C1': -1655351678, 'C2': 4570},
                 {'C1': 1229714276, 'C2': 50324},
                 {'C1': -1213142998, 'C2': 11319}],
        'B3': 102,
        'B4': [-0.3617766499519348,
                 -0.5140736103057861,
                 -0.8350076675415039,
                 -0.05404835566878319],
        'B5': 0.4123615026473999,
        'B6': 249,
        'B7': 3403156849061356910,
        'B8': -870600126},
 'A4': {'D1': 0.9155915379524231, 'D2': -2493, 'D3': 20404},
 'A5': 6512,
 'A6': 15022450549449832846,
 'A7': {'E1': 7053997527801816776,
        'E2': 1751839368,
        'E3': 40895,
        'E4': -7618,
        'E5': -1713042516,
        'E6': [2753799470,
                 3614430972,
                 991812540,
                 805970802,
                 719923493,
                 4274021596,
                 2364762220],
        'E7': [7368664603583206166, 12544548196245147925],
        'E8': -21054}}

```

## 2. Двоичные данные:

```

(b'DMEA')\xf3 1\x08\xe0\xae\x10\x1c\x00\x00\x00G>\xd0\x0f#\x95\x01\xdf'\xb0w>'
b'N{\xcdj\xd9\xdf\x99\x00\x93cf\x1f!J\x81\xba\\\xde7\x1218z1\xc2\x0c\xaa\x11'

```

```

b"\xfa\x00'\x00-\x003>\x87$\xb7\xbe\x88B\x99\x00\x02\x00\x00\x00%\x00\x00\x00"
b'\x03\x009\xc3\x00\x00\x00\x02\x00?\xbe\xccZu\x1bQ\x81\xba\x18\xc0'
b'\x84\xc4\xa4\xdb\xbdF$\x7f\xac\xf6\x9d-\x9f\xb1K\x94\xbe\x82~\x1b'
b'\x1d\xb3\xb6\xc9\x06\x1d\xa8\xd8\xf5DCm\xceY\xc9\t1\x11\x11y*\xd3Vh'
b'l\xb0\x06,A\xe1c\x8bU\x01\x91\xf4\xd1\x1c\tD\x1b\xb7\xe3\xe1\xf8\xf2T\x00'
b'\x06\x00\x00\x00k\x00\x02\x00\x00\x00\x8361')

```

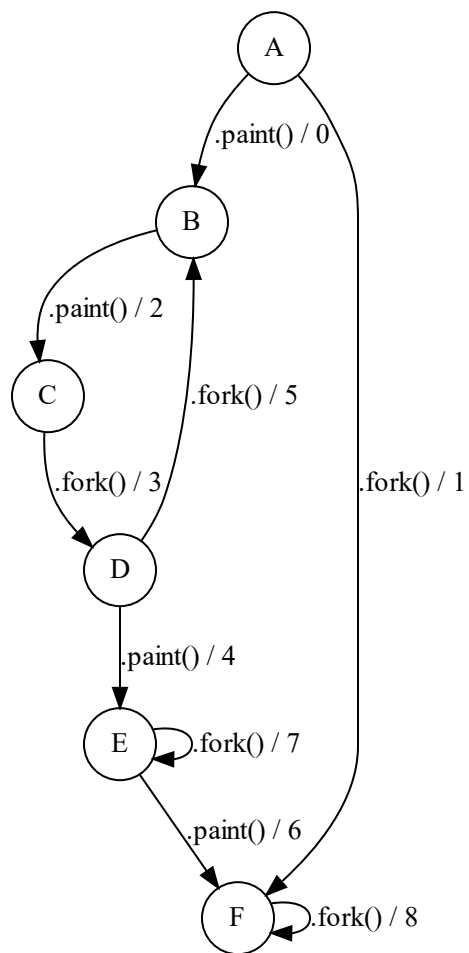
Результат разбора:

```

{'A1': 9075633322459180560,
 'A2': 28,
 'A3': {'B1': 'cf',
        'B2': [{'C1': 522275457, 'C2': 47708},
                 {'C1': -566816207, 'C2': 14458},
                 {'C1': 1824656554, 'C2': 4602}],
        'B3': 195,
        'B4': [0.26395198702812195, -0.2661330997943878],
        'B5': -0.39912763237953186,
        'B6': 27,
        'B7': 5873180004539745444,
        'B8': -608352732},
 'A4': {'D1': 0.4063654839992523, 'D2': -27391, 'D3': 57184},
 'A5': 45175,
 'A6': 4489662000726138777,
 'A7': {'E1': 3189077828810637713,
        'E2': 4107344905,
        'E3': 17435,
        'E4': -18461,
        'E5': -503778732,
        'E6': [2142041757,
                 765440331,
                 2495513214,
                 454931382,
                 3372621224,
                 3639952451],
        'E7': [7912360315314049297, 8731023195683729414],
        'E8': 13873}}

```

**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.paint()</code>	0
<code>o.paint()</code>	2
<code>o.fork()</code>	3
<code>o.fork()</code>	5
<code>o.paint()</code>	2
<code>o.fork()</code>	3
<code>o.paint()</code>	4
<code>o.fork()</code>	7
<code>o.fork()</code>	7
<code>o.paint()</code>	6
<code>o.fork()</code>	8

2. Пример использования класса C32:

o = C32()	
o.paint()	0
o.paint()	2
o.fork()	3
o.fork()	5
o.paint()	2
o.fork()	3
o.paint()	4
o.fork()	7
o.paint()	6
o.fork()	8
o.fork()	8

## Вариант №19

**Задача 3.1.** Реализовать разбор двоичного формата данных (в духе формата WAD игры Doom или графического формата PNG). Данные начинаются с сигнатуры 0x43 0x5a 0x43, 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	Размер (uint32) и адрес (uint32) массива char
	3	uint32
	4	Структура B
	5	int8
	6	Структура E
	7	uint8

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

Структура C:	1	int16
	2	float

Структура D:	1	int16
	2	uint8
	3	Размер (uint16) и адрес (uint16) массива double
	4	Массив uint8, размер 7

Структура E:	1	uint16
	2	int64
	3	uint16
	4	uint16
	5	uint64
	6	uint64

Примеры разбора двоичного формата с помощью функции f31:

1. Двоичные данные:

```
(b' CZC1\xd7\xd0\xef\xd6Y\xd6$\x00\x00\x00\x03\x00\x00\x00CD\xd7\xc1\x9b\x00'
 b'\x00\x00\x04\x00\x00\x00^\x00\x00\x00~\xe2\x87\xbd\xe8\xa1-\xc6\xa0d\xe1+w4'
 b'\x8a\x90\xf3\xe8\x1a\x86\xbct\x9f\x16\xbffK\x89\xad\xbb1\dcxonf\xbe\x08'
 b'\xbf\x11K\xe2\xc2\x89\xbd\x96Q\x00\xe7w?\x01\xfcq\x93\x1a\xbf~v\xa2\x00\x00'
 b'\x00F\x00\x00\x00L\x00\x00\x00R\x00\x00\x00X?\xd9\x97>\xca\xd4)\x80\xbf\xe2'
 b'\xb3\x9aI/\x15\x1a|\xf8\xcb\x00\x02\x00n\x00\xe5WL\x83:e')
```

Результат разбора:



```
{'A1': 7842967004600981028,
'A2': 'onf',
'A3': 1154990491,
'A4': {'B1': [{'C1': -16888, 'C2': -0.5675641298294067},
               {'C1': -15735, 'C2': -0.07339668273925781},
               {'C1': -6281, 'C2': 0.5077582001686096},
               {'C1': -27878, 'C2': -0.9939976930618286}],
'B2': {'D1': 31992,
       'D2': 203,
       'D3': [0.3998562794446556, -0.5844241551755289],
       'D4': [0, 229, 87, 76, 131, 58, 101]}},
'A5': -30,
'A6': {'E1': 34749,
       'E2': -1684014454542114517,
       'E3': 30516,
       'E4': 35472,
       'E5': 17575326711803059990,
       'E6': 13791793962675974620},
'A7': 120}
```

## 2. Двоичные данные:

```
(b"CZC\x91\xcek\x90\x18\x04'2\x00\x00\x00\x02\x00\x00\x00C\xc9\x11\x08\x15\x00"
b"\x00\x00\x03\x00\x00\x00W\x00\x00\x00s\xee\x19\xa4\xf1\xe6\x9bm'E"
b'\x18\xb5\x9e\x7f"\xeb\xc4\t\xa2\x98\xb6\xd4]\xdbJ\xca\xb9\xa5\xfd'
b'\xc8\xe4\xb5jz\xce\xff\xbd\xc4)\x9a\x9f9>\xe8\xc0|\x80?\x126+\x00'
b'\x00\x00E\x00\x00\x00K\x00\x00\x00Q?\xe4\xc0\xb7\xe5\xdc<\x86\xbf\xea;y<'
b'\xbcw\xa8\x98\x95\xbc\x00\x02\x00c\x88\xb3\xce\xbc\x99!V')
```

### Результат разбора:

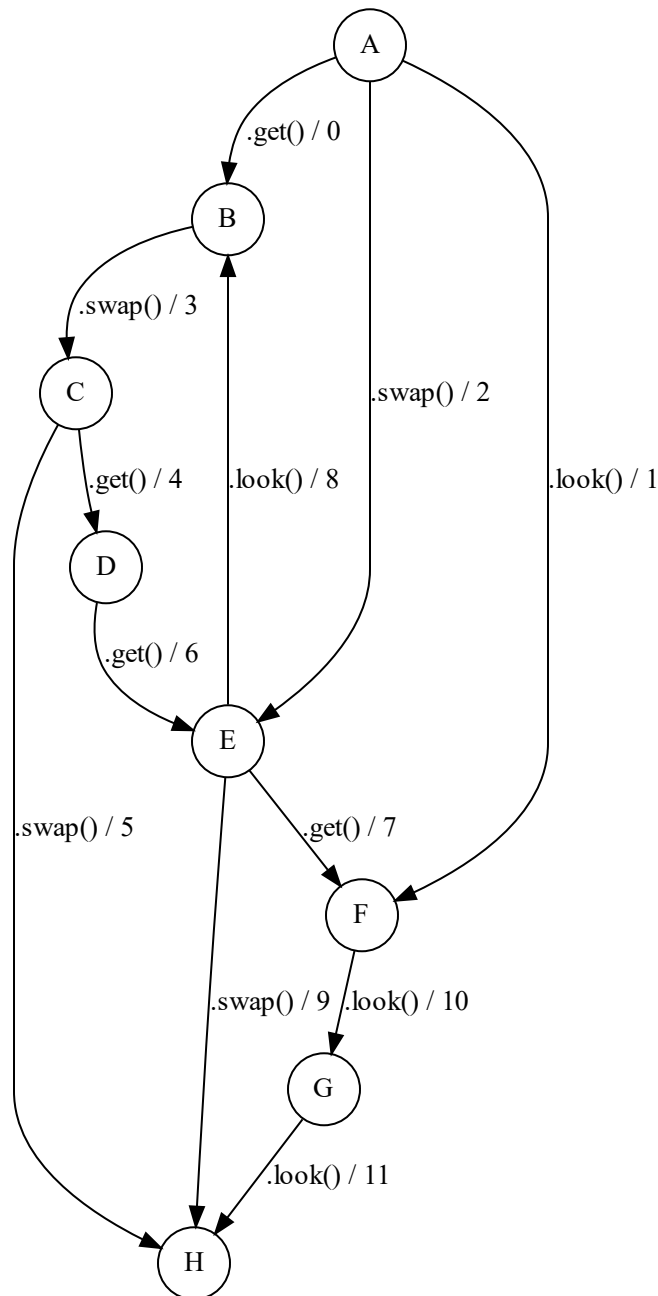
```
{'A1': 10506453247324333874,
'A2': 'jz',
'A3': 3373336597,
'A4': {'B1': [{'C1': -12545, 'C2': -0.09578247368335724},
               {'C1': -24775, 'C2': 0.4545912742614746},
               {'C1': 31872, 'C2': 0.5711390376091003}],
'B2': {'D1': -26475,
       'D2': 188,
       'D3': [0.6485251893481212, -0.8197599588636324],
       'D4': [136, 179, 206, 188, 153, 33, 86]}},
'A5': -18,
'A6': {'E1': 6564,
       'E2': -1015953772812887883,
       'E3': 40575,
       'E4': 8939,
```

```

'E5': 14144691123615355997,
'E6': 15801665141285439716},
'A7': 181}

```

**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.swap()	2
o.look()	8
o.swap()	3
o.get()	4
o.get()	6
o.get()	7
o.look()	10
o.look()	11

## 2. Пример использования класса C32:

o = C32()	
o.get()	0
o.swap()	3
o.get()	4
o.get()	6
o.look()	8
o.swap()	3
o.get()	4
o.get()	6
o.look()	8
o.swap()	3
o.get()	4
o.get()	6
o.get()	7
o.look()	10
o.look()	11

## Вариант №20

**Задача 3.1.** Реализовать разбор двоичного формата данных (в духе формата WAD игры Doom или графического формата PNG). Данные начинаются с сигнатуры 0x4e 0x55 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	uint8
	2	Размер (uint32) и адрес (uint16) массива char
	3	int64
	4	Размер (uint16) и адрес (uint32) массива char
	5	Структура B
	6	Массив структур D, размер 3
	7	uint16

Структура B:	1	Структура C
	2	int16

Структура C:	1	int32
	2	uint16
	3	double
	4	uint16
	5	int16

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

Примеры разбора двоичного формата с помощью функции f31:

1. Двоичные данные:

```
(b'NUX*\x07\x00\x00\x00F\x00xD0\x97\xbf\xeb\x12>\x04\x00M\x00\x00\x00'
b'\xb2\xe7\xaa\xbb\xec\xa4|o\\Y\x8e\xd5\xed\xbf\xa4\x0fj>\xa5~\xbcL\xd5\xc8'
b'\x02\x00Q\x00\x1cZ7\xb6\x03\x00a\x00\xba\xd3a\xdd\x03\x00y\x00a\xf2egcuje'
b'etcdh$mt\xf5\xba\x8b)V0E\xcb?\xea\xf5\xfa\x8d\x82\x99\x95\x94t0E,\x17D2'
b'!\xf4\xd4\xfeL\x97>j\xba\x82\x0f\xfeU1\xc0\x85\xd1}\x9c\xebf3T{'
b'\xa1\xe3\x15\x14\x1a1\x9c\xae\xf5[8?\\')
```

Результат разбора:

```
{'A1': 42,
'A2': 'egcuje',
'A3': 4472896588040127608,
'A4': 'tcdh',
```

```

'A5': {'B1': {'C1': -1146427470,
              'C2': 42220,
              'C3': -0.9323188538869398,
              'C4': 4004,
              'C5': 15978},
       'B2': 32421},
'A6': [{'D1': 19644,
        'D2': -43,
        'D3': -56,
        'D4': [6208647196405165348, -8215984183742610128]},
        {'D1': 23068,
        'D2': 55,
        'D3': -74,
        'D4': [3190043273189497218, 5548105935961080855, 6196407191456071319]},
        {'D1': 54202,
        'D2': 97,
        'D3': -35,
        'D4': [7416193275595702321,
                1879151011931313203,
                6647093542637182060]}],
'A7': 62049}

```

## 2. Двоичные данные:

```

(b'NUX\xff\x02\x00\x00\x00F\x00\x94\x7f\xc0\x075\xa8\xbd\xbd\x03\x00'
 b'H\x00\x00\x00\xa9\xc0w\x8e\xfb\xe1P\xca\xc7\xcf>\x\dc\xbfL\x1dK\x1R\x1f0'
 b'\xa6\xa9I\x93\x02\x00K\x00\x0d&\xb0\x02\x00[\x00\xd9\x83d\xae\x02\x00k\x00'
 b'a\x8avpfmg\xbf28k\xb42\xc6\x1b\x8f\xd7/\x01\x8f\x9c\n\x92\xe2>z\xee\x16'
 b'\xbd\xcd\xeb[i\x14\xd2\xa2^{\xdf\xc9\x8b,Zn\x9cLKy\x16z\x1d#\xa5\x9b\xbb')

```

### Результат разбора:

```

{'A1': 255,
 'A2': 'vp',
 'A3': -4774475084226068588,
 'A4': 'fmg',
 'A5': {'B1': {'C1': -1904754519,
              'C2': 57851,
              'C3': -0.44483919421251006,
              'C4': 7500,
              'C5': -3765},
       'B2': -4014},
'A6': [{'D1': 43430,
        'D2': 73,
        'D3': -109,
        'D4': [2001342834887111359, -7923348456390797425]},

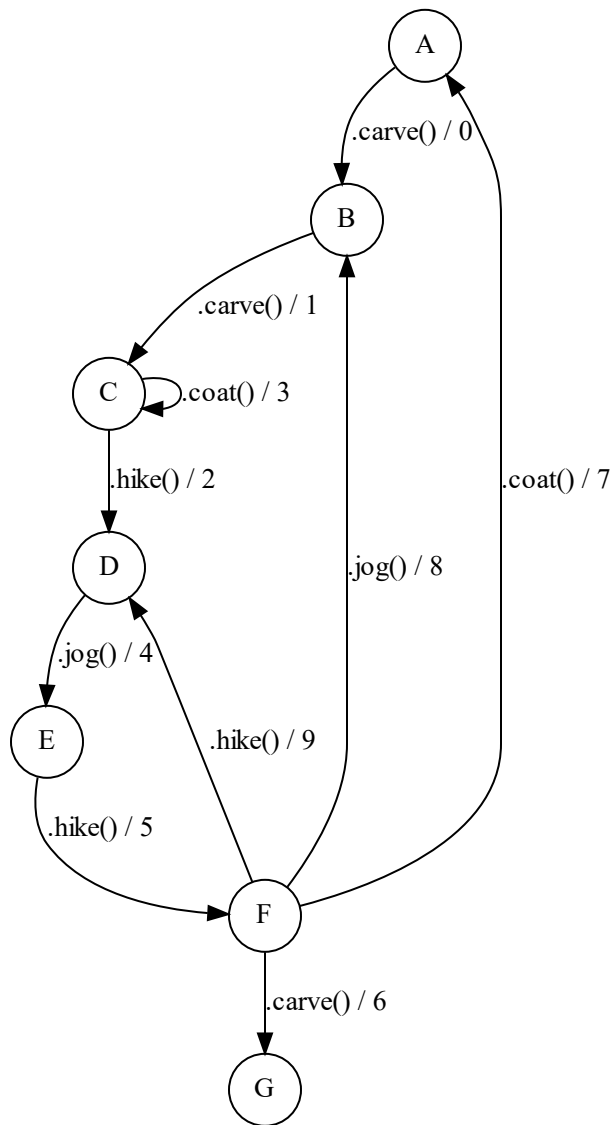
```

```

{'D1': 25840,
 'D2': 38,
 'D3': -80,
 'D4': [-1455299198382883102, -2343175127713945253]},
{'D1': 33753,
 'D2': 100,
 'D3': -82,
 'D4': [5425883648847809481, -4928163796989897095]}],
'A7': 35425}

```

**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.jog()	RuntimeError
o.carve()	0
o.carve()	1
o.coat()	3
o.hike()	2
o.jog()	4
o.hike()	5
o.jog()	8
o.carve()	1
o.jog()	RuntimeError
o.hike()	2
o.jog()	4
o.hike()	5
o.hike()	9

## 2. Пример использования класса C32:

o = C32()	
o.carve()	0
o.carve()	1
o.coat()	3
o.coat()	3
o.hike()	2
o.jog()	4
o.jog()	RuntimeError
o.hike()	5
o.jog()	8
o.carve()	1
o.hike()	2
o.jog()	4
o.hike()	5
o.carve()	6

## Вариант №21

**Задача 3.1.** Реализовать разбор двоичного формата данных (в духе формата WAD игры Doom или графического формата PNG). Данные начинаются с сигнатуры 0x4b 0x5a 0x5a 0x5a, 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	uint64
	3	uint16
	4	uint32
	5	Массив структур B, размер 4
	6	Структура C
	7	Структура D
	8	uint16

Структура B:	1	Массив char, размер 8
	2	float

Структура C:	1	int8
	2	uint64
	3	int64
	4	int64
	5	Массив uint8, размер 6
	6	Размер (uint16) и адрес (uint16) массива uint8
	7	uint64
	8	uint16

Структура D:	1	uint16
	2	uint32

Примеры разбора двоичного формата с помощью функции f31:

1. Двоичные данные:

```
(b"KZZZj\x99\xd02\x9b3\xfd50\xea='8\xa38v\xec\x9afhcymbn1?w]\xe8dhyh1tnd\xbey"
b's\xfavqoainqt\xbf{\x8c\xdewhoybzu1;\xae"\xd7\x8eA\t\xe0\xb1\x18\xb6-E\x98'
b'R\xb0{C\xc6\xe9\xcf(\xca,\xe4\xe8a\xae\x1f\x14\x92\xff\r\xe7\x00\x05\x00'
b' {\x7f\xacK\x13\xdc\xb2Aw0\xf7\x8a-\xd7\r{N\x03\x04\xc0D\xe48\x9d' )
```

Результат разбора:

```
{'A1': 1788465202,
'A2': 11183560833955937575,
'A3': 14499,
'A4': 947317914,
```



```
'A5': [{ 'B1': 'fhcymbnl', 'B2': 0.9662766456604004},
        { 'B1': 'dhyhltnl', 'B2': -0.24360647797584534},
        { 'B1': 'wqoainqt', 'B2': -0.9826182126998901},
        { 'B1': 'whoybzul', 'B2': 0.005314211826771498}],
'A6': { 'C1': -114,
        'C2': 4686523938484137285,
        'C3': -7470714788389656113,
        'C4': 2939211065464480174,
        'C5': [31, 20, 146, 255, 13, 231],
        'C6': [192, 68, 228, 56, 157],
        'C7': 9199810687490212215,
        'C8': 12535},
'A7': { 'D1': 35373, 'D2': 3607984974},
'A8': 772}
```

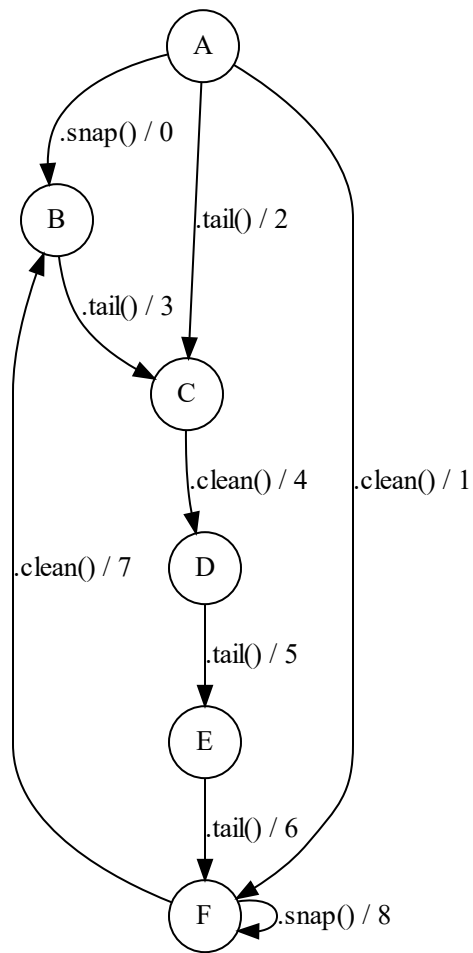
## 2. Двоичные данные:

```
(b'KZZZB\x9eJ\x8a\x87P\xfdY\x0e/\xc7\x90\x18\xeah\x3n\x8byejntpam\xbe\x91'
b'\xc6\xc7pubidnlg?.M\xa4jkuyvpli\xbe\x80\x87xbpbghzmh?-\x1a\xfa\xc8\xe4'
b'\xb5\x07!m\xea\x92=\xb2\xa6D\xd9.\x93\xa9\xd4_\x1b\xc1uou\x90\xd9\xe'
b'\xder-\x8d$\x00\x04\x00{\x99\xf3?"Xvr\xdeF\xf7]\xbem(u\xed\x84\xaa'
b';\xbbt')
```

### Результат разбора:

```
{ 'A1': 1117670026,
  'A2': 9750571752189052816,
  'A3': 6378,
  'A4': 1759735435,
  'A5': [{ 'B1': 'yejntpam', 'B2': -0.28471967577934265},
          { 'B1': 'pubidnlg', 'B2': 0.6808722019195557},
          { 'B1': 'jkuyvpli', 'B2': -0.2510335445404053},
          { 'B1': 'bpbghzmh', 'B2': 0.6761928796768188}],
  'A6': { 'C1': -56,
          'C2': 16480086251591602749,
          'C3': -5573691789244388908,
          'C4': 6853283968099651801,
          'C5': [142, 222, 114, 45, 141, 36],
          'C6': [170, 59, 187, 116],
          'C7': 11093279723889259230,
          'C8': 18167},
  'A7': { 'D1': 23998, 'D2': 1831368173},
  'A8': 46724}
```

**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.tail()</code>	2
<code>o.snap()</code>	RuntimeError
<code>o.clean()</code>	4
<code>o.clean()</code>	RuntimeError
<code>o.tail()</code>	5
<code>o.tail()</code>	6
<code>o.tail()</code>	RuntimeError
<code>o.snap()</code>	8
<code>o.clean()</code>	7
<code>o.snap()</code>	RuntimeError
<code>o.tail()</code>	3

2. Пример использования класса C32:

<code>o = C32()</code>	
<code>o.tail()</code>	2
<code>o.tail()</code>	RuntimeError
<code>o.clean()</code>	4
<code>o.tail()</code>	5
<code>o.clean()</code>	RuntimeError
<code>o.tail()</code>	6
<code>o.snap()</code>	8
<code>o.tail()</code>	RuntimeError
<code>o.clean()</code>	7
<code>o.tail()</code>	3

## Вариант №22

**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 0x4e, 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	uint64
	2	Структура В

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

Структура С:	1	int8
	2	uint64
	3	Размер (uint16) и адрес (uint32) массива структур D

Структура D:	1	Массив uint32, размер 5
	2	int64
	3	Размер (uint16) и адрес (uint32) массива uint8
	4	Массив double, размер 2
	5	uint32

Структура Е:	1	uint16
	2	int16
	3	float
	4	uint32
	5	double
	6	int16

Примеры разбора двоичного формата с помощью функции f31:

1. Двоичные данные:

```
(b"IBCn\xa0\xb1\xb8\xcf\xae\x1d\x87\xf4\x81\x00\x95\x07'\x00\x00\x00"
b'\xa4\xd1\x9fIL\x97$\x98ykR\xa6\xa6J x\x0fo\xe0\x91\xfc\xc0\x98\xbc'
b'\xfa\x95\x8c]\x0bF\xe9L\x98f\xad\xbd"\x9a\xfb.\xe4\xcb\x91\xa9'
b'\xc5\xf7\xe5\xd1C\x00\x05\x00\x00\x00!\xbf|00*\x85W\x00?\xe9F\x0c\xa1'
b'\x99\xed\xfb0_\x14\xa9XE\xb7\tgC\x80\xb2M\xef\xd9\x99\x16\xaf\xcf\xbb2\xd8'
b'!\xd2\x88\xbf\x85Z\xcc\xb8K+\x16\x00\x03\x00\x00\x00&?\xb1t\x0c\xd6\xbb\xf3'
b'@\xbf\xe8H-\x8b\x1c\x9d<\xf0s\xe8\xcf\xb3\x88r\xf5\xbc\x1e\xb7'
b'\xcb\xfe\x00\x02\x00\x00\x00)L\xa9'\x8e\xbf=xn\xfd:\x92*\xbf\xe5KSNF\xa1T'
b'\x832')
```

Результат разбора:

```
{'A1': 11579239319065167860,
 'A2': {'B1': 129,
        'B2': {'C1': -77,
                 'C2': 9832191124821232638,
                 'C3': [{'D1': [3231235322,
                                2509004043,
                                1189694616,
                                1722662178,
                                2600021732],
                        'D2': -3778051944654450365,
                        'D3': [74, 32, 120, 15, 111],
                        'D4': [-0.006882008783881011, 0.7898009449267516],
                        'D5': 1595189592},
                  {'D1': [1169623399,
                          1132507725,
                          4024015126,
                          2949626674,
                          3626095240],
                  'D2': -4646207605251036394,
                  'D3': [224, 145, 252],
                  'D4': [0.06817703478368475, -0.7588107792879906],
                  'D5': 4034128079}]},
 'B3': 1831,
 'B4': {'E1': 19625,
        'E2': 24718,
        'E3': -0.7401188611984253,
        'E4': 4248474154,
        'E5': -0.6654449967218548,
        'E6': -31950},
 'B5': [3516877132, 2535757945, 1800578726]}
```

## 2. Двоичные данные:

```
(b' IBCN\xad\xf8\x12\x98/\xd4\xf1\xd4\x06\x00\x92\xba\x89\x00\x00\x00'
 b'\xa1\xb2\xba\xd7.\xa3\xde\xffe\x9b\x1cE\x9a\x01|\x99\x0e\x82\xb80'
 b'\xceN\x95\xbf6U3\xbc2fb\xab\xa9\xcd\x17\x19\x8c\xe6\x84\xba\xf6\xda\xa4\xc4'
 b'\xb7\xa6\x00\x03\x00\x00\x00!?\xd8\xe8\x1e;\x104H\xbf\xd9\xf51'
 b'\xe4\xce\xe1\x9cd\xc4\x1bY\xaa\x10\xf3$\t'\x1cg\x90\x16\x17\x98"\xc3v'
 b'I\xc8\xbc\aa\x00\x9e\xb9\xba\x00\x02\x00\x00\x00$?\xe2\x93g\xc7\x06'
 b'\xc5\xa4\xbf\xee\x9d\xe8\xf8U\xc4\xee\x87WB\xd4\xa6\x9b\xbb\x12\xca&'
 b'\x82\x98\xff\x00\x02\x00\x00\x00&\xda\xcei\xb4\xbe:\xc1]\x10\xac\xc2'
 b'*?\xe1\xc77Q0\xd4\xb8r\x8c')
```

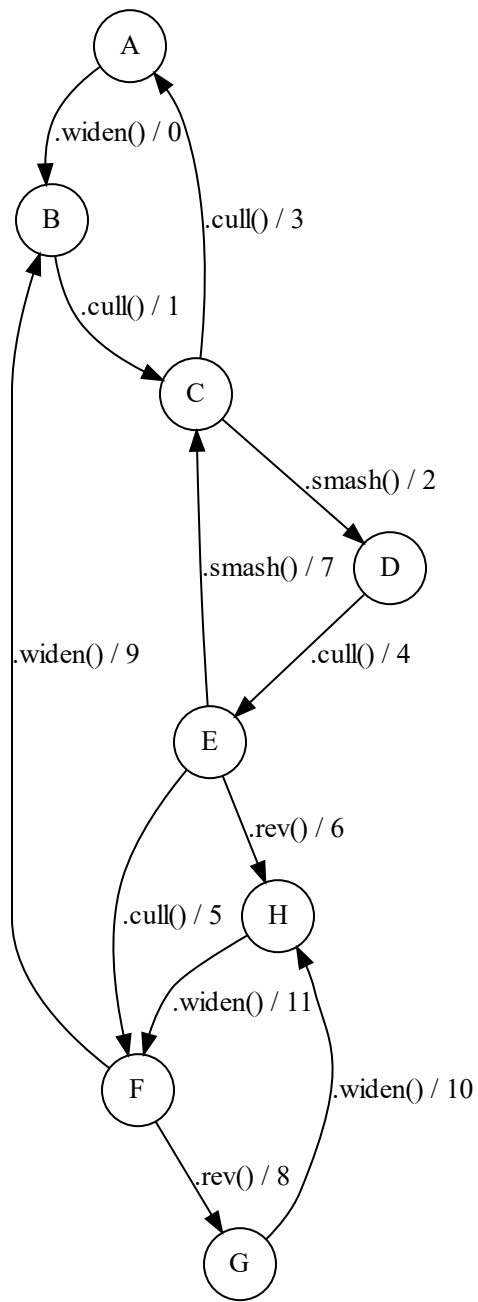
Результат разбора:

```

{'A1': 12535790007632589268,
 'A2': {'B1': 6,
        'B2': {'C1': -90,
                 'C2': 11221583555962509567,
                 'C3': [{'D1': [3092237902,
                                2512336469,
                                867971686,
                                1655417293,
                                387550438],
                          'D2': -8882515896107223130,
                          'D3': [1, 124, 153],
                          'D4': [0.38916736381435557, -0.4055905088858027],
                          'D5': 1690573657},
                  {'D1': [2853237540,
                          153558119,
                          2417366936,
                          576701302,
                          1237892193],
                    'D2': 6683141383724644794,
                    'D3': [14, 130],
                    'D4': [0.5804938208804526, -0.9567761278432536],
                    'D5': 2270642900}}]},
 'B3': 47753,
 'B4': {'E1': 56014,
        'E2': 27060,
        'E3': -0.18237824738025665,
        'E4': 279757354,
        'E5': 0.5555683696863047,
        'E6': 29324},
 'B5': [2998589230, 2749300581, 2602321306]}}

```

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



1. Пример использования класса C32:

o = C32()	
o.widen()	0
o.widen()	RuntimeError
o.cull()	1
o.smash()	2
o.cull()	4
o.cull()	5
o.rev()	8
o.widen()	10
o.rev()	RuntimeError
o.widen()	11
o.widen()	9
o.cull()	1
o.cull()	3
o.rev()	RuntimeError
o.widen()	0
o.cull()	1
o.smash()	2
o.cull()	4
o.smash()	7

## 2. Пример использования класса C32:

o = C32()	
o.widen()	0
o.cull()	1
o.cull()	3
o.widen()	0
o.cull()	1
o.smash()	2
o.cull()	4
o.cull()	5
o.rev()	8
o.widen()	10
o.widen()	11
o.widen()	9
o.cull()	1
o.smash()	2
o.cull()	4
o.smash()	7



## Вариант №23

**Задача 3.1.** Реализовать разбор двоичного формата данных (в духе формата WAD игры Doom или графического формата PNG). Данные начинаются с сигнатуры 0x9 0x48 0x52 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	Структура C
	3	uint32
	4	Массив char, размер 4
	5	Размер (uint32) и адрес (uint32) массива структур D
	6	Размер (uint32) и адрес (uint16) массива float
	7	double
	8	Массив uint8, размер 5

Структура B:	1	uint32
	2	double
	3	int8

Структура C:	1	uint16
	2	uint64
	3	double
	4	uint64
	5	int8

Структура D:	1	uint8
	2	int32
	3	int16
	4	uint16
	5	Массив uint16, размер 7

Примеры разбора двоичного формата с помощью функции f31:

### 1. Двоичные данные:

```
(b'\tHRW\x00\x00\x00F\xa4\xa1\xe2\xd5\xfe\x190~\xb9\xe7?\xd5g\xf5y\x91'
b'p\xc0\x1d\x0e\xf7\xfc9@$\x04\x85\xa6\xc9f`uyqo\x00\x00\x00\x02\x00'
b'\x00\x00S\x00\x00\x00\x05\x00\x81?\xe6\x11\xf1\xf5\xd9U\x9e\xd0\x8b\x01'
b'\xf7C8\\\x12z?\xe7y\x013\xae\xf0\xea\x96\xd9\x90N\xe1K\xa1}\x9d9y\x81pE'
b'\xa6\x02S\xacF\x13\xee\\\xab\x15\xa820\x14\xddV\x1c\xd5\x17C\x90Hq\r'
b'\xf7\xf1\xb6\xc1\xa4\xc1;\xb0*\xbfpq\xf7\xbfNe>;\x84\x1b=\xa2\xe2'
b'\xeb\xbd\x9e\xe9E')
```

Результат разбора:

```
{'A1': {'B1': 945558138, 'B2': 0.7335210809174246, 'B3': -106},
'A2': {'C1': 42145,
'C2': 16345249807272425959,
'C3': 0.3344701468647706,
'C4': 2093883539438183428,
'C5': -123},
'A3': 2798216800,
'A4': 'uyqo',
'A5': [{'D1': 217,
'D2': -1873878709,
'D3': -24195,
'D4': 40249,
'D5': [31105, 28741, 42498, 21420, 17939, 61020, 43797]}],
{'D1': 168,
'D2': 844043485,
'D3': 22044,
'D4': 54551,
'D5': [17296, 18545, 3575, 61878, 49572, 49467, 45098]}],
'A6': [-0.939086377620697,
-0.8996337056159973,
0.18312112987041473,
0.0795343741774559,
-0.0775933638215065],
'A7': 0.6896905709970047,
'A8': [208, 139, 1, 247, 67]}
```

## 2. Двоичные данные:

```
(b'\tHRW\x00\x00\x00F\x86\x8fG\x10F\xc30y,\xf3?\xe7@\x81\x07W\xe8\x9et\xa9'
b'Z_\x19\xce\x00[\x94\x8a\Fxvoi\x00\x00\x00\x02\x00\x00\x00S\x00'
b'\x00\x00\x05\x00\x81\xbf\xdc}\x078\xae\xdc\\\x82\xe0\xc5d\x87J\xa4'
b'\xf2B\xbf\xec\x84\x90R\\\xa3^l]\x92\x0c^\x7f\x13$\xd4\xe5+4%\xe2Y\x98aH'
b'\xe4\xb1uP\x13\x11\xc7h~\x9dZ\x86\xbfm\x8b\\R\xe9p\xa3c\xde\xd9\xdb'
b'\xb7\xb0q\xeb\xde\xbe\xda\xff\xff\xbfz6\x05>S\xb6_\xbfv\rT\xbf\x14"\xa9')
```

## Результат разбора:

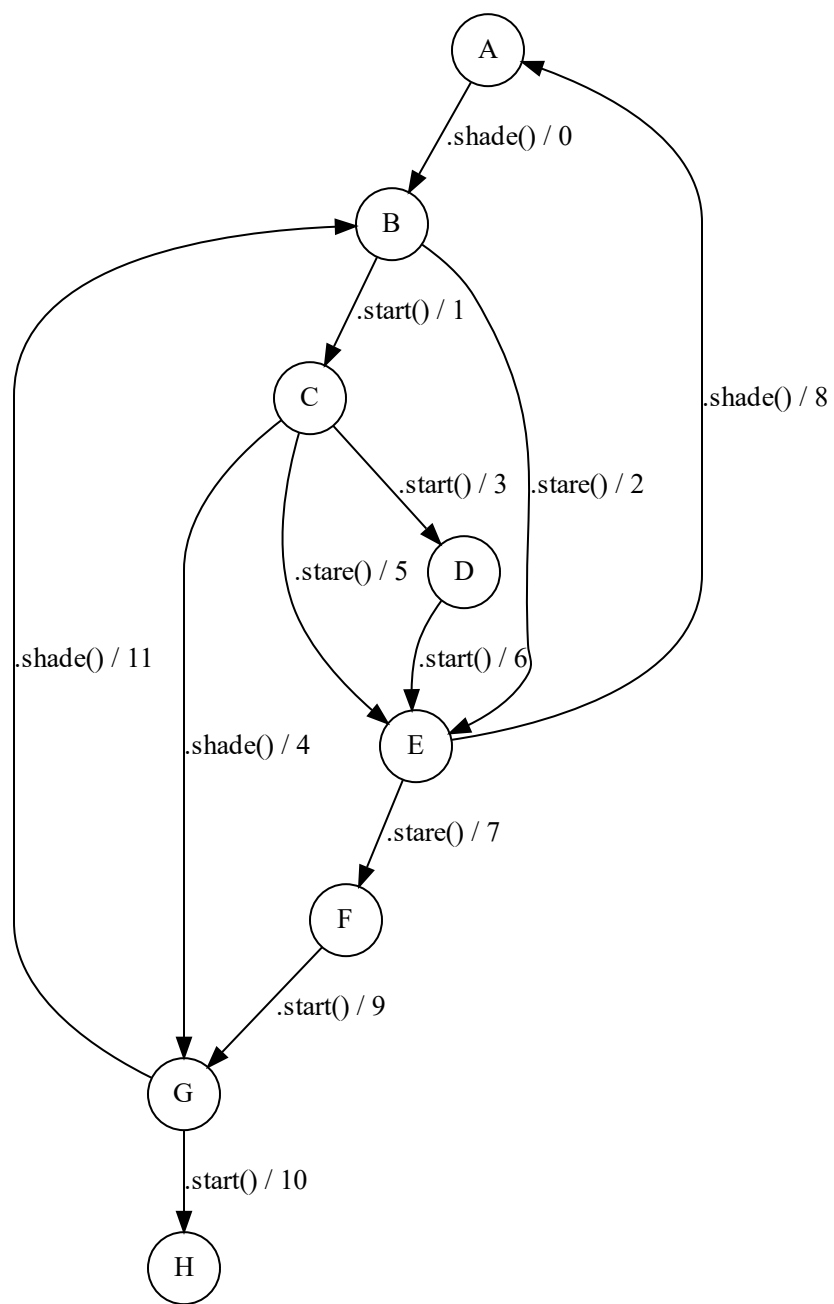
```
{'A1': {'B1': 1252323906, 'B2': -0.8911820992117148, 'B3': 108},
'A2': {'C1': 34447,
'C2': 5120670580466068723,
'C3': 0.7266240256715439,
'C4': 8406349543965110363,
'C5': 95},
'A3': 2492099654,
'A4': 'xvoi',
'A5': [{'D1': 93,
```

```

'D2': -1844683137,
'D3': 4900,
'D4': 54501,
'D5': [11060, 9698, 22936, 24904, 58545, 30032, 4881]},
{'D1': 199,
'D2': 1753128282,
'D3': -31041,
'D4': 28043,
'D5': [23634, 59760, 41827, 57049, 56247, 45169, 60382]}},
'A6': [-0.4277343451976776,
-0.9773867726325989,
0.20675037801265717,
-0.9611408710479736,
-0.578653872013092],
'A7': -0.4451311162515077,
'A8': [130, 224, 197, 100, 135]}

```

**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.shade()          0
o.start()          1
o.stare()          5
o.shade()          8
o.shade()          0
o.stare()          2
o.stare()          7
o.start()          9
o.shade()          11
o.start()          1
o.start()          3
o.start()          6
o.stare()          7
o.start()          9
o.start()          10

```

## 2. Пример использования класса C32:

```

o = C32()
o.shade()          0
o.stare()          2
o.stare()          7
o.stare()          RuntimeError
o.start()          9
o.shade()          11
o.start()          1
o.start()          3
o.start()          6
o.shade()          8
o.start()          RuntimeError
o.shade()          0
o.shade()          RuntimeError
o.start()          1
o.shade()          4
o.start()          10

```

## Вариант №24

**Задача 3.1.** Реализовать разбор двоичного формата данных (в духе формата WAD игры Doom или графического формата PNG). Данные начинаются с сигнатуры 0x44 0x5a 0x51, за которой следует структура A. Порядок байт: от старшего к младшему. Адреса указаны в виде смещений от начала данных. В решении разрешено использовать модуль struct.

Структура A:	1	uint8
	2	uint64
	3	double
	4	Адрес (uint32) структуры B
	5	Массив адресов (uint32) структур C, размер 2
	6	uint8

Структура B:	1	double
	2	uint16

Структура C:	1	double
	2	Массив char, размер 5
	3	uint64
	4	int32
	5	uint16
	6	Адрес (uint16) структуры D
	7	uint16
	8	int16

Структура D:	1	uint64
	2	Размер (uint32) и адрес (uint16) массива int16
	3	Размер (uint32) и адрес (uint16) массива uint8
	4	int16
	5	uint8
	6	uint8

Примеры разбора двоичного формата с помощью функции f31:

1. Двоичные данные:

```
(b'DZQ\xfb\xba\x9c(\xe3j\x0cD\xef\xbf\xe9p1\x074\x84Z\x00\x00\x00!\x00\x00\x00M'
b'\x00\x00\x00\x95\x05?\xd8R.\x18\xdeT\xd8\x9a0\x89x}\xa4\xc5=\xb7#PK\x04d'
b'\xcf\x05\xd5Q\xea\x00\x00\x00\x02\x00+\x00\x00\x00\x06\x00/r\xb3b'
b'0\xbf\xe5\x9f0\x96\xb3-\xc0yamfn%\x0fkR\xf6\x10\xaeZ\x90d\xa2\x89\x8c\xbc'
b'\x005\xe2\x1b#\xe85\n]\xda\x99\xcaB\x86_\xf0\xe5`\x82J\xfe\xce\x1c\xab\xea'
b'g\xd3Q\xca\xa7\x00\x00\x00\x04\x00n\x00\x00\x00\x07\x00v\xfc)k\x03?\xc73'
b"\x0ezb\xa5\x08bjhln'd\xb1c`\xe1\xd5\x85\xdaDb\xa5\x15F\x00}\xec\xa8\xb9J")
```

Результат разбора:

```
{'A1': 251,
'A2': 13446667544623334639,
'A3': -0.7949452534413795,
'A4': {'B1': 0.38001587322431374, 'B2': 55450},
'A5': [{'C1': -0.6756823485715628,
'C2': 'yamfn',
'C3': 2670471108113772122,
'C4': -1872452983,
'C5': 36028,
'C6': {'D1': 5405556293070311914,
'D2': [12425, 30845],
'D3': [164, 197, 61, 183, 35, 80],
'D4': 29363,
'D5': 98,
'D6': 48},
'C7': 57883,
'C8': 9093},
{'C1': 0.18124562240453934,
'C2': 'bjhl1n',
'C3': 2838588705535546757,
'C4': -633052507,
'C5': 5446,
'C6': {'D1': 2066002585727388327,
'D2': [2653, -9575, -13758, -31137],
'D3': [240, 229, 96, 130, 74, 254, 206],
'D4': -983,
'D5': 107,
'D6': 3},
'C7': 60584,
'C8': -18102}],
'A6': 5}
```

## 2. Двоичные данные:

```
(b'DZQ\x95e\x7f\xee\x80\x8d\xfb\x0e\x93?\xe9\xec\x10\x0c\xc7c~\x00\x00\x00!'
b'\x00\x00\x00R\x00\x00\x00\x94\x8e\xbf\xe72<\xc8f:4\\$\xce\xdf\xb8\xb8\xb8'
b't<\x94\xec\xe1y\xbc\x0e\xf18\x1e\x87\xebI%\x14\x9a\x13\x00\x00\x00\x06\x00+'
b'\x00\x00\x00\x03\x007Fm\x19\xf3?\xa4\xa2\xf8\x8a\x88\xc0esebk\xd6'
b'\xe7\xce\xf9\x91\x8d5\xac\x08:\xd0\xdb\xee\xad\x00:Za\xedK\xd6\xee\xd2\n0'
b'\x12\xa4\xa3\xe0q\xc5\x00\x12\x07\xd0\xa2v\x00\x00\x00\x03\x00s\x00\x00'
b'\x00\x03\x00ym\x90K*\xbf\xea\x8dt=\xae\x08\xa2pmqs1\xde\xe1(\xc6\xb6A\xfd'
b'\x91\x0e5kh\x8f\x9a\x00|~\xfd\x91\xa0')
```

Результат разбора:

```
{'A1': 149,
```

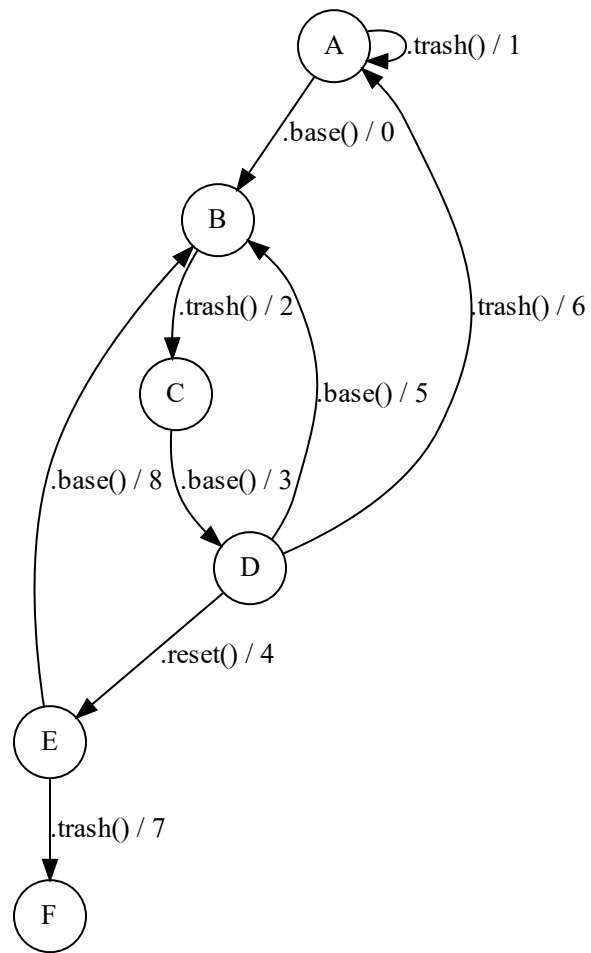
```

'A2': 7313826555778240147,
'A3': 0.8100662469470163,
'A4': {'B1': -0.7248824991273239, 'B2': 23588},
'A5': [{'C1': 0.040305869004229944,
        'C2': 'esebk',
        'C3': 15485573415020869036,
        'C4': 138072283,
        'C5': 61101,
        'C6': {'D1': 2199985642381023763,
                'D2': [-12577, -18248, -18316, 15508, -4895, 31164],
                'D3': [14, 241, 56],
                'D4': 18029,
                'D5': 25,
                'D6': 243},
        'C7': 23137,
        'C8': -4789},
        {'C1': -0.8297673420604676,
        'C2': 'pmqs1',
        'C3': 16060162580106313105,
        'C4': 238381928,
        'C5': 36762,
        'C6': {'D1': 8197958774138380918,
                'D2': [-10514, -11766, 20242],
                'D3': [164, 163, 224],
                'D4': 28048,
                'D5': 75,
                'D6': 42},
        'C7': 32509,
        'C8': -28256}],
'A6': 142}

```

**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.trash()</code>	1
<code>o.base()</code>	0
<code>o.reset()</code>	RuntimeError
<code>o.trash()</code>	2
<code>o.trash()</code>	RuntimeError
<code>o.base()</code>	3
<code>o.base()</code>	5
<code>o.trash()</code>	2
<code>o.base()</code>	3
<code>o.reset()</code>	4
<code>o.base()</code>	8
<code>o.trash()</code>	2
<code>o.base()</code>	3
<code>o.trash()</code>	6

## 2. Пример использования класса C32:

```
o = C32()
o.base()          0
o.trash()         2
o.base()          3
o.reset()         4
o.base()          8
o.trash()         2
o.trash()         RuntimeError
o.base()          3
o.base()          5
o.trash()         2
o.base()          3
o.trash()         6
o.reset()         RuntimeError
o.trash()         1
```

## Вариант №25

**Задача 3.1.** Реализовать разбор двоичного формата данных (в духе формата WAD игры Doom или графического формата PNG). Данные начинаются с сигнатуры 0xfc 0x4d 0x50 0x4e 0x4b, 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 (uint32) и адрес (uint32) массива char
	3	int32
	4	Структура B
	5	uint8
	6	int64
	7	Размер (uint16) и адрес (uint16) массива структур D
	8	int32

Структура B:	1	uint32
	2	uint64
	3	Структура C

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

Структура D:	1	uint16
	2	int64
	3	Массив uint32, размер 4

Примеры разбора двоичного формата с помощью функции f31:

1. Двоичные данные:

```
(b'\xfcMPNK\xf3\x00\x00\x00\x04\x00\x00\x00M\x0b\x01\xdaC\x9d\xab'
b"\x1c\x11\x13\xffG\xbe\xb1\x1a\xca=\xee\x17\xcf\xef\xe1?\xea\xa6\x99\x15'^n"
b'\xbf\xb0\x1c\x01\x984H ?\xe7\x9f\x86L\x90rt\xcc\xd4\xc4\xb7\xc3\xef\xa0\x0c'
b'\xe2\x00\x02\x00Q\xea/\t0annp\xfc\xd6k\x90\x12u\xd2\xc7;1g\x9b\xfbgY'
b'\x9c\xd6\xb6u\xa5\xa9j[\xe2,gj0\x859\x17p\xe3\xc3D\x96\xdd\xcf\x88'
b'\xc0\x8e\x8c\xd5\x85\xce\xd2\xfd"E\xf7\x95k')
```

Результат разбора:

```
{'A1': -13,
'A2': 'annp',
'A3': 184670787,
'A4': {'B1': 2645236753,
```

```

'B2': 1440949290122529341,
'B3': {'C1': 238,
       'C2': 1712836591,
       'C3': 225,
       'C4': [0.8328366673621994,
              -0.06292734113620257,
              0.7382232184787214]}},
'A5': 204,
'A6': -3115162990066791198,
'A7': [{'D1': 64726,
        'D2': 7750715255961369393,
        'D3': [1738275687, 1503450806, 1973791082, 1541549159]},
        {'D1': 27215,
        'D2': -8847014219367627626,
        'D3': [3721365696, 2391594373, 3469933858, 1173853547]}],
'A8': -366016177}

```

## 2. Двоичные данные:

```

(b'\xfcmPNKe\x00\x00\x00\x04\x00\x00\x00M\x1cD\xb7\xeb\xa7\xa8a\x82T\xf3\xd7'
b'_b\x0c\x94\xbd\\\x98\xfe*\xb\x1b\xf16\xbf\xe3\xdd\x9dK_\xa2\xdc?\xd4\x83\xa4'
b'%\xf7^L\xbf\xc6k\x9e\xc4\x9f\xb50\xe0\x17\xae\x83\xa3K\x1d\xde'
b'\x82\x00\x02\x00Q#m\x98RotxiS\x9cu\xc31\xb3!\xbE\xa4h\x83z\xea\xa9'
b'\x13\x97\xca\xe8yB\xcd\xd9{w\x95\x03\xb8\xef\x03\x85rU\xf6J\x89\xce\x1b_'
b'k\xbc\xa7\xa3\x92v\xd8\xee}|\xd1H\xa1')

```

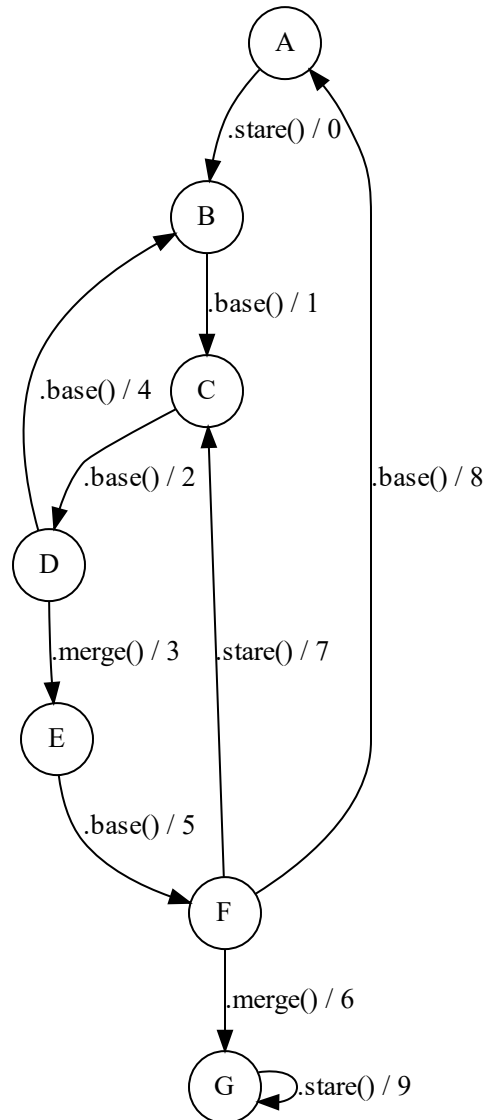
## Результат разбора:

```

{'A1': 101,
 'A2': 'otxi',
 'A3': 474265579,
 'A4': {'B1': 2810872404,
        'B2': 17570617345931787612,
        'B3': {'C1': 152,
                'C2': -30794767,
                'C3': 54,
                'C4': [-0.6208025428905688,
                       0.3205347414745787,
                       -0.17515930748724307]}},
 'A5': 224,
 'A6': 1706446046183153282,
 'A7': [{'D1': 21404,
        'D2': 8485680768320488868,
        'D3': [1753447146, 2836633546, 3900261069, 3648747413]},
        {'D1': 907,
        'D2': -1223987947599672695,
        'D3': [3457900395, 3165102994, 1993928317, 2094090401]}],
 'A8': 594384978}

```

**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.base()	RuntimeError
o.stare()	0
o.merge()	RuntimeError
o.base()	1
o.merge()	RuntimeError
o.base()	2
o.base()	4
o.base()	1
o.base()	2
o.merge()	3
o.base()	5
o.stare()	7
o.stare()	RuntimeError
o.base()	2
o.merge()	3
o.base()	5
o.merge()	6
o.stare()	9

## 2. Пример использования класса C32:

o = C32()	
o.stare()	0
o.base()	1
o.base()	2
o.merge()	3
o.base()	5
o.stare()	7
o.base()	2
o.base()	4
o.base()	1
o.base()	2
o.merge()	3
o.base()	5
o.merge()	6
o.merge()	RuntimeError
o.stare()	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 0xb6 0x56 0x4e 0x46 0x4c, за которой следует структура A. Порядок байт: от старшего к младшему. Адреса указаны в виде смещений от начала данных. В решении разрешено использовать модуль struct.

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

Структура B:	1	uint16
	2	uint32

Структура C:	1	int64
	2	Структура D

Структура D:	1	Массив uint32, размер 7
	2	int16
	3	Массив uint64, размер 6
	4	Массив uint16, размер 4

Структура E:	1	uint8
	2	uint8
	3	float

Примеры разбора двоичного формата с помощью функции f31:

1. Двоичные данные:

```
(b'\xb6VNFL\x00w\x00}ewty\xdf\xe3\x7f\x83\x10v8j#\xa7\x16~\xe1\x9f\xa2'
b'\x98U\x8c\x9e*\xe1\x05]\xdc\xa3u\xc7\x160\x8e!\x07"BT\xd6j%\xb7jh\xd8\xc7'
b'\xd6n,0\x86\xb0\xe8\xaa\xa_\xc9Ns\xfb\xe93\xfd\xde\xcc\x00|\xbf\xf8\x19'
b'\xb9&\x8c\xb9,\xcb\xab\x07\x96\x1e\x04\xebI\xc5\xa5\x0bW[kS4{_\x7f+\xf7\xf7'
b'\xfa\xb0\xf0\x00\x00\x00\x83>L(\j\xec^\xf7\xbd\x1f\xf8I\xbb\x0e\x95;7\x1b'
b'\xff?~U\xa8')
```

Результат разбора:

```
{'A1': [{'B1': 60510, 'B2': 4156366840}, {'B1': 18875, 'B2': 244661047}],
'A2': 'ewty',
'A3': {'C1': -2313865582644676502,
'C2': {'D1': [598152830,
```

```

3785335448,
1435278890,
3775225308,
2742404886,
1334714631,
574772438],
'D2': 27173,
'D3': [13216491336747085356,
3496676674039865289,
5653138936298331852,
35113969407408422,
10140185289260766750,
354458108905084763],
'D4': [27475, 13435, 24447, 11255]}},
'A4': 2147135728,
'A5': {'E1': 27, 'E2': 255, 'E3': 0.9934945106506348},
'A6': 0.19937272369861603}

```

## 2. Двоичные данные:

```

(b'\xb6VNFL\x00w\x00}mupm\xd6\xfb\x81\xb1$\x8e/gio\x7fu\xfa\xfa\xfa\xac\tt\xeac'
b'\x7f\xd9f0D\xa7&t\x1c\xad\x06\xff#\xb9M\x9f\xfa\xfb5\xcfi\xb5o0\x1e'
b'\xc3"\x9f\x03\x87\xee\x9b\n\xe9(2\x80\xa3\x86*\x13\xae0mY\x95T\xa1\x97'
b'\xf6\x93J<\xf2\x02$\xe4FU\xaa\xe2\\\x88\x80\xc0@\x96pK9\xb8\xd3i'
b'\xa0\xaa\xeb\xfd\x16U\x01\x00\x00\x00\x83\xbfD\xbe;\t,\x1c<\xcb8Q\x05\xd2'
b'\x9b\x08\\\x98\n>\xcb\x83=')

```

## Результат разбора:

```

{'A1': [{'B1': 2348, 'B2': 473746232}, {'B1': 20741, 'B2': 3533375580}],
'A2': 'mupm',
'A3': {'C1': -2955626132616106137,
'C2': {'D1': [1768914805,
4210404361,
1961517951,
3647352900,
2804315164,
2902916899,
3108872185],
'D2': -18993,
'D3': [7617116730376266399,
254434253640116274,
9269399973335543917,
6455158693921461066,
4391574944111547818,
16311062036991743600],

```

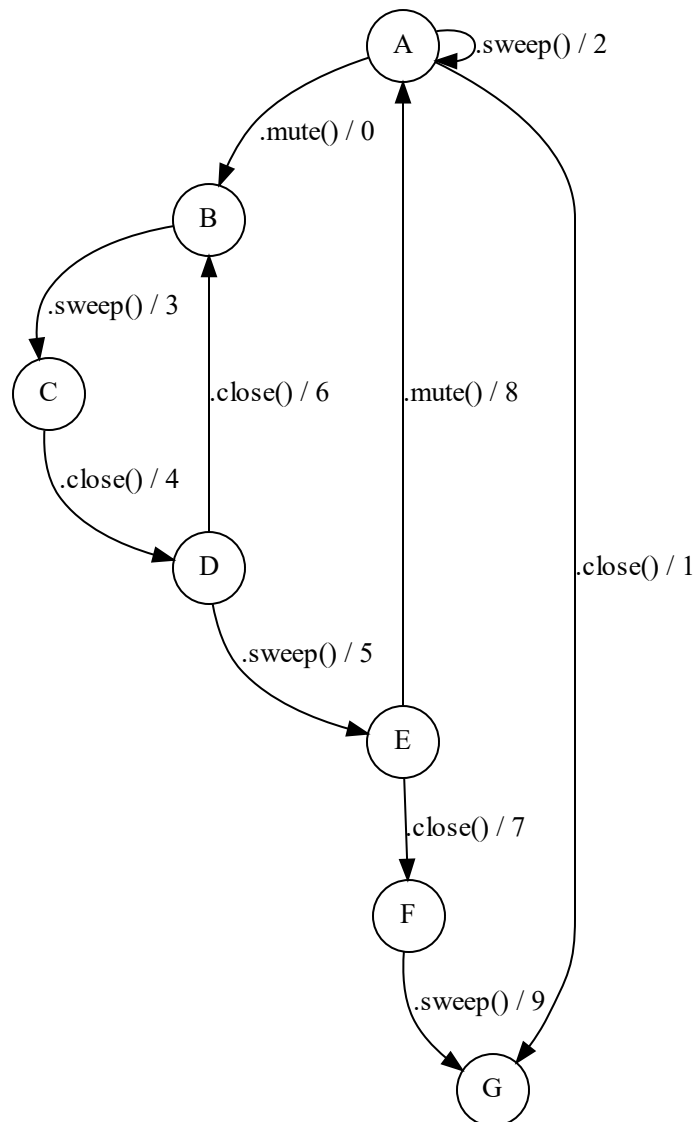


```

'D4': [19257, 47315, 27040, 43755]}},
'A4': -48868095,
'A5': {'E1': 152, 'E2': 10, 'E3': 0.3974856436252594},
'A6': -0.7685276865959167}

```

**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.sweep()          2
o.mute()           0
o.sweep()          3
o.close()          4
o.close()          6
o.sweep()          3
o.close()          4
o.sweep()          5
o.mute()           8
o.close()          1

```

## 2. Пример использования класса C32:

```

o = C32()
o.mute()           0
o.sweep()          3
o.close()          4
o.mute()           RuntimeError
o.close()          6
o.sweep()          3
o.close()          4
o.sweep()          5
o.mute()           8
o.sweep()          2
o.close()          1

```

## Вариант №27

**Задача 3.1.** Реализовать разбор двоичного формата данных (в духе формата WAD игры Doom или графического формата PNG). Данные начинаются с сигнатуры 0x58 0x50 0x5a 0xe5, за которой следует структура А. Порядок байт: от старшего к младшему. Адреса указаны в виде смещений от начала данных. В решении разрешено использовать модуль struct.

Структура А:	1	Размер (uint32) и адрес (uint16) массива char
	2	Размер (uint32) и адрес (uint32) массива адресов (uint32) структур В
	3	int32
	4	Адрес (uint32) структуры С

Структура В:	1	uint32
	2	uint16
	3	uint32

Структура С:	1	float
	2	uint8
	3	Структура D
	4	Массив int64, размер 2
	5	uint8
	6	float
	7	Структура E

Структура D:	1	uint16
	2	double
	3	uint32
	4	uint64
	5	int16

Структура E:	1	Размер (uint32) и адрес (uint32) массива int16
	2	int8

Примеры разбора двоичного формата с помощью функции f31:

1. Двоичные данные:

```
(b'XPZ\xe5\x00\x00\x00\x02\x00\x1a\x00\x00\x00\x02\x00\x00\x00>V'  
b'\xfc\xb5\x00\x00\x00<dn-\x93L\x9a\xc1\x08\xac+ju\x9d\xd1\xa7\xa0\xc7\x1f'  
b'U\x9fo\x9e\x00\x00\x00\x1c\x00\x00\x00&\x1f\x1f\xef\xb3\xbfA\x99\x94'  
b'\xd2\x8fw?\xba\xf3$I7j\xb0s\xf0F\xd45\x89\xa0IQV\x16\xc2M\x1a\xc9E\xa5'  
b'N\x10\x14P\xf5\x15\xba\xf9\xc9ph\x93\xbc>>\xed\xf5\x00\x00\x00\x02\x00\x00'  
b'\x008$')
```

Результат разбора:

```
{'A1': 'dn',
'A2': [{'B1': 764628122, 'B2': 49416, 'B3': 2888526453},
{'B1': 2647762848, 'B2': 50975, 'B3': 1436512158}],
'A3': 1045888181,
'A4': {'C1': -0.7562496662139893,
'C2': 210,
'C3': {'D1': 36695,
'D2': 0.10527254855643764,
'D3': 1945126612,
'D4': 3857790792577193666,
'D5': 19738},
'C4': [-3943564143997202187, 1565838464034313148],
'C5': 62,
'C6': 0.24309523403644562,
'C7': {'E1': [8177, -4173], 'E2': 36}}}
```

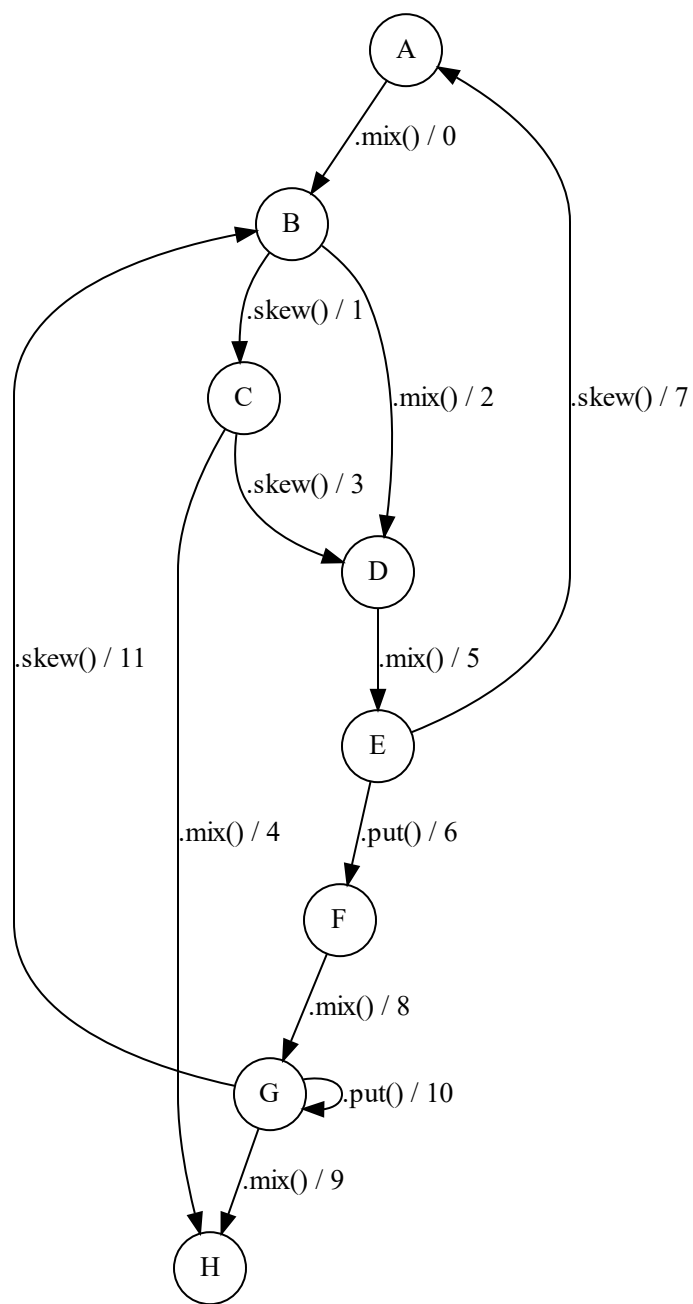
## 2. Двоичные данные:

```
(b'XPZ\xe5\x00\x00\x00\x02\x00\x1a\x00\x00\x00\x03\x00\x00:\x17\xb7'
b'\x0f\x0e\x00\x00\x00JwuH\x85I\xc8\xd5\xde\xf1\xa5X\xde\x1c\xb4\xa95"R'
b'T\xeb\x04\x9f\xdfF\x11\xd7\xb2\xcaU\x08\x80m\x00\x00\x00\x1c\x00\x00'
b'\x00&\x00\x00\x0001R\x15\x89\xbc\xa;0\x0c\xc0\x96\xbf\xe1\x054\x87-\x8e'
b'\x0eAu]\x0e\x96\xd7I%\xa3w\x0e\x10\x84\x1d\xf0u\x90\xb0\xfe7\x7fr\x9e'
b'\xfa\x14\xd0\x1a\x06\xfd\x8b\xa0?f!\xdf\x00\x00\x00\x02\x00\x00\x00F^')
```

## Результат разбора:

```
{'A1': 'wu',
'A2': [{'B1': 1216694728, 'B2': 54750, 'B3': 4054145246},
{'B1': 481601845, 'B2': 8786, 'B3': 1424688287},
{'B1': 3745911255, 'B2': 45770, 'B3': 1426620525}],
'A3': 397872910,
'A4': {'C1': -0.018827108666300774,
'C2': 12,
'C3': {'D1': 49302,
'D2': -0.5318853988832772,
'D3': 1098210574,
'D4': 10869236651687087632,
'D5': -31715},
'C4': [-1119829842478006414, -6991252587516854901],
'C5': 160,
'C6': 0.8989543318748474,
'C7': {'E1': [12626, 5513], 'E2': 94}}}
```

**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.mix()	0
o.skew()	1
o.skew()	3
o.mix()	5
o.put()	6
o.mix()	8
o.put()	10
o.skew()	11
o.mix()	2
o.mix()	5
o.skew()	7
o.mix()	0
o.skew()	1
o.mix()	4

## 2. Пример использования класса C32:

o = C32()	
o.mix()	0
o.skew()	1
o.put()	RuntimeError
o.skew()	3
o.skew()	RuntimeError
o.mix()	5
o.put()	6
o.mix()	8
o.skew()	11
o.put()	RuntimeError
o.mix()	2
o.mix()	5
o.put()	6
o.mix()	8
o.put()	10
o.mix()	9

## Вариант №28

**Задача 3.1.** Реализовать разбор двоичного формата данных (в духе формата WAD игры Doom или графического формата PNG). Данные начинаются с сигнатуры 0x55 0x56 0x55 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	uint8
	2	Размер (uint16) и адрес (uint16) массива адресов (uint16) структур В
	3	float
	4	Адрес (uint32) структуры С
	5	int16
	6	Структура F

Структура В:	1	int8
	2	int16
	3	int8

Структура С:	1	Структура D
	2	int32
	3	Адрес (uint16) структуры Е

Структура D:	1	Размер (uint32) и адрес (uint16) массива int16
	2	Массив uint16, размер 3
	3	uint16

Структура Е:	1	uint16
	2	uint64
	3	int8

Структура F:	1	int16
	2	Размер (uint16) и адрес (uint16) массива float
	3	Массив float, размер 3

Примеры разбора двоичного формата с помощью функции f31:

### 1. Двоичные данные:

```
(b'UVUXL\x04\x005\x00\x82>\x9d\xbdN\x00\x00\x00V\xb9"\xa9\x04\x00b\x00\xea\xc9}'
b'? \x16\x9c ?\xbe\xaaN?v\xc4\x12\x84n\xc2\xf1\xeaz\xe5\x8d\x90\xb5\xee\xe9'
b'\x9b%\x00)\x00-\x001\x00\xd6\xbb#\n\xba\xa1\x0b\xba\x95\x91\xe9Y\xda\xfb\x80'
b'7T\x03\x00\x00\x00=\x00\xb4\x91\xb6\xbe\xf4%\xb8Xooo3C\x00\x8ai\x9d\xbd\x086'
b'7\xbf\x07\xdd*?N\xd8*?' )
```

Результат разбора:

```
{'A1': 76,
'A2': [{'B1': 118, 'B2': 4804, 'B3': -124},
        {'B1': 110, 'B2': -3646, 'B3': -22},
        {'B1': 122, 'B2': -29211, 'B3': -112},
        {'B1': -75, 'B2': -5650, 'B3': -101}],
'A3': -0.07677938044071198,
'A4': {'C1': {'D1': [-17450, 2595, -24134],
               'D2': [37300, 48822, 9716],
               'D3': 22712},
        'C2': 862941039,
        'C3': {'E1': 47627, 'E2': 3999473384334922133, 'E3': 84}},
'A5': -18090,
'A6': {'F1': -22238,
        'F2': [-0.07686145603656769,
                -0.7156682014465332,
                0.6674351096153259,
                0.6673630475997925],
        'F3': [0.9913622140884399, 0.6273816823959351, 0.8072928190231323]}}
```

## 2. Двоичные данные:

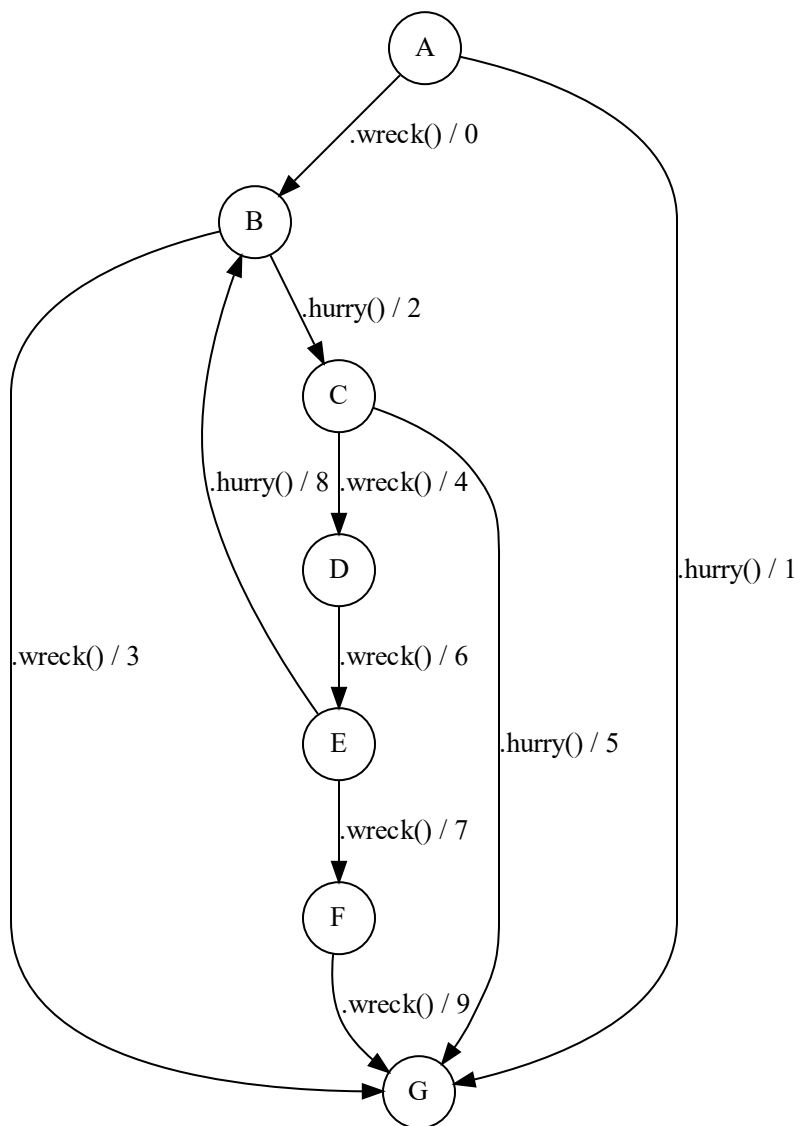
```
(b'UVUX\xcc\x03\x001\x00/k\x01\xbfJ\x00\x00\x00@\xf0\x91b\x02\x00^\x00?\x1b)'
b'\xbfhS\xb0\xbe\x8eFt?\xfc@\xabm\xc83\x8ce\xb2\x10M\xc7%\x00)\x00-\x00\x0e'
b'K\xd2\xba\xf0\x9f\xfc\x8c\x9a\x0c\xc1k\xa6\x06\xa7\xf1b\xcc\x81\x04\x00'
b'\x00\x007\x00\x98\x8fy-\xce\xd0#>\xe2ad-?\x00!\xf6\x05?\x9bw[\xbf')'
```

## Результат разбора:

```
{'A1': 204,
'A2': [{'B1': -4, 'B2': -21696, 'B3': 109},
        {'B1': -56, 'B2': -29645, 'B3': 101},
        {'B1': -78, 'B2': 19728, 'B3': -57}],
'A3': -0.5055417418479919,
'A4': {'C1': {'D1': [19214, -17710, -24592, -29444],
               'D2': [36760, 11641, 53454],
               'D3': 15907},
        'C2': 761553378,
        'C3': {'E1': 3226, 'E2': 14727599431128345537, 'E3': -127}},
'A5': -4032,
'A6': {'F1': 25233,
        'F2': [0.5232868790626526, -0.8568055033683777],
        'F3': [-0.6605719923973083, -0.34438633918762207, 0.9542015790939331]}}
```

**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.wreck()           0
o.hurry()           2
o.wreck()           4
o.hurry()           RuntimeError
o.wreck()           6
o.hurry()           8
o.hurry()           2
o.wreck()           4
o.wreck()           6
o.wreck()           7
o.wreck()           9
  
```

## 2. Пример использования класса C32:

o = C32()	
o.wreck()	0
o.hurry()	2
o.wreck()	4
o.wreck()	6
o.hurry()	8
o.hurry()	2
o.wreck()	4
o.wreck()	6
o.wreck()	7
o.wreck()	9

## Вариант №29

**Задача 3.1.** Реализовать разбор двоичного формата данных (в духе формата WAD игры Doom или графического формата PNG). Данные начинаются с сигнатуры 0x4a 0x56 0x52 0x50 0xb7, 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 3
	2	Массив адресов (uint16) структур B, размер 3
	3	Структура C
	4	int16
	5	uint32
	6	uint32
	7	Структура D
	8	uint32

Структура B:	1	uint64
	2	int32
	3	int16

Структура C:	1	uint64
	2	uint8
	3	Размер (uint16) и адрес (uint16) массива int16

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

Примеры разбора двоичного формата с помощью функции f31:

### 1. Двоичные данные:

```
(b'JVVR\b7ihmp\x00~\x00\x8c\x00\xc8(\x83\x8cq\xf7R\x88\t\x02\x00\x9a\x00\xeb'
b'\xa5W\xbd3\x12\xf2\x1e\x06\xcb\x02\x12\x0e\xedQ\xa9\x9d\xef\xbf\x04\xed'
b'\xafoVl\xee?J\x15\xae\x98\xc4E\xe3\xbf\x84\x1e*\xa9P\xa1\xdc\xbf@]<*\x8e'
b'\xb7\xbf\xe6\x17\xe2\xfe\x92\xee\xbf>z\xca\x87\x7f\x8b\xe2\xbf\x90\xa9'
b'\xf1\xc7[\x87\xc6\xbf\x02\x00\x00\x00\x9e\x004e\x1e\xec\xbf\xcf)\x8avU\x0c,'
b'\xce\x04\x92:\f\b0\xd8\x19\x17R?\xdb\xbc\x89\x9fG\xb2\xc0H\xbe\xa9\xb1\xda'
b'\xd8\xb8i3S\xa4\x8c\xd9\x14oyr\xd7z\xe3\\\xf5\xf9')
```

Результат разбора:

```
{ 'A1': 'ihm',
  'A2': [ { 'B1': 3174005805001854897, 'B2': 982648014, 'B3': 26204 },
          { 'B1': 13608540320837523632, 'B2': -1303928951, 'B3': 18624 },
          { 'B1': 3704695410010466750, 'B2': -645094317, 'B3': 28436 } ],
```

```
'A3': {'C1': 9823185804309244104, 'C2': 9, 'C3': [29305, 31447]},
'A4': -23061,
'A5': 305380695,
'A6': 3406175986,
'A7': {'D1': 2,
      'D2': [-0.9879957771412953,
              0.950724809779246,
              -0.6022665960341922,
              -0.4473458772194834,
              -0.09201766043803605,
              -0.9554434882163576,
              -0.5795285846869047,
              -0.17600581419840156],
      'D3': [23779, 63989]},
'A8': 3961414964}
```

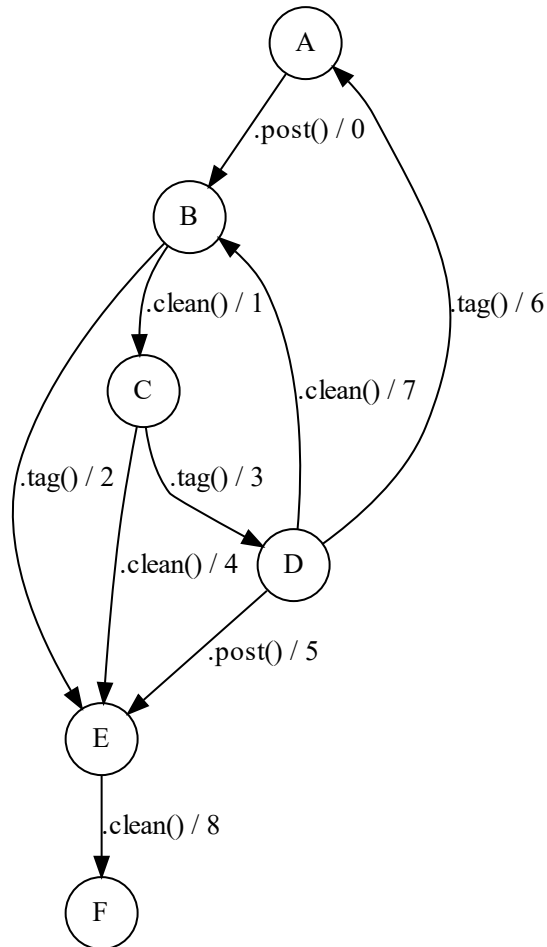
## 2. Двоичные данные:

```
(b'JVVRP\xb7zlvpx00~\x00\x8c\x00p\xc8\xdck\xe5\x83\x8d\x0b\x1b\x03\x00\x9a\x00)'
b'\xe09\x80\r\xd5_\xab\xd1X\xfd4TXR\x9d\xd4\xd1\xbf\x14&\xa6\xa9hk'
b'\xe3\xbf\xd26\x10\xc2\xcf\xe0\xbf\x00\xd3\x9f(\xda\x88\xa6\xbfv\x15'
b'M\x18\xd4\x14\xec\xbf@v\x17\x07\x14\x13\xc6?L\xbb\x16\x80\xac\xfa\xea?Td'
b'\x95m\xb3E\xdf?\x02\x00\x00\x00\xa0\x00\x1b\xdc\xde8h\xdetY\xb5\xdf/\xe1'
b'._\xf4\xe9\xa6\t\x88\xa5@1\xf6\x9a\xe8!L \x011\xd6\xf0\xfa\xaaAT\xcdq\\b'
b'\x82\xab\x05G`\xb6\x1dA\xe5\x11Y\xaf\xb8\x94\xd3>')
```

## Результат разбора:

```
{'A1': 'zlv',
 'A2': [{'B1': 16226433952422092392, 'B2': -369860818, 'B3': 2470},
        {'B1': 2443373180027446664, 'B2': 822157388, 'B3': -3882},
        {'B1': 7087665040270142202, 'B2': 1191553922, 'B3': -18848}],
 'A3': {'C1': 832466527513790576, 'C2': 27, 'C3': [16669, 4581, -20647]},
 'A4': -8151,
 'A5': 3574431801,
 'A6': 1490135903,
 'A7': {'D1': 253,
      'D2': [-0.2786019615317088,
              -0.6068614304978746,
              -0.5142592230128804,
              -0.04401284929182303,
              -0.8775425409918813,
              0.172457221472472,
              0.8430998327001844,
              0.4886292047645544],
      'D3': [38072, 16083]},
 'A8': 954129435}
```

**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.post()           0
o.clean()          1
o.tag()            3
o.clean()          7
o.clean()          1
o.tag()            3
o.tag()            6
o.post()           0
o.tag()            2
o.clean()          8

```

## 2. Пример использования класса C32:

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

## Вариант №30

**Задача 3.1.** Реализовать разбор двоичного формата данных (в духе формата WAD игры Doom или графического формата PNG). Данные начинаются с сигнатуры 0x52 0x52 0x55 0x54, за которой следует структура А. Порядок байт: от старшего к младшему. Адреса указаны в виде смещений от начала данных. В решении разрешено использовать модуль struct.

Структура А:	1	int64
	2	Структура В

Структура В:	1	Размер (uint16) и адрес (uint32) массива char
	2	int64
	3	Размер (uint16) и адрес (uint32) массива структур С
	4	double
	5	int8
	6	Адрес (uint32) структуры D
	7	double

Структура С:	1	uint16
	2	Размер (uint16) и адрес (uint16) массива int64
	3	uint64
	4	Массив int8, размер 5
	5	Размер (uint32) и адрес (uint16) массива uint8

Структура D:	1	float
	2	int32
	3	int8
	4	int8

Примеры разбора двоичного формата с помощью функции f31:

1. Двоичные данные:

```
(b'RRUT\xc9):\x80\x1d{\x10\x07\x00\x02\x00\x00\x005\x1a&\x03\xfc\x9cr'  
b'| \x83\x00\x02\x00\x00\x00]\xbf\xe85\xdcXKM\xf8M\x00\x00\x00\x8f?\xe5\xb3'  
b'\x8dtW\x14\xcewq\xc9p\xb9\xdd\xd3+\x9a\x9ag\xbc\xd9\xcc\lw\xdd\x19V'  
b'\xef\xf5f\xc8\xa0U\xea\xdb\x8f\t"[\x9f\xb7\xae+]\_x83\xbd\xfa\x82\xd1\x00'  
b'\x02\x007\x03h\xed\xb1\xcd\x05\x97\xcb\x85T\x8f\x7f\xec\x00\x00\x00\x03'  
b'\x00Gq\x10\x00\x02\x00J\xc8\x1b\xde\xaf\xe0Y\x04J\x84<\xcf\xbbv\x00\x00\x00'  
b'\x03\x00Z\xbfI\xd0\t\xde\x8b4\xf5\xb1E')
```

Результат разбора:

```
{'A1': -3951562876116070393,  
'A2': {'B1': 'wq',  
       'B2': 1884197877592521859,
```

```

'B3': [{'C1': 33489,
        'C2': [-3931438112312681830, 7475088953276095769],
        'C3': 245707527596382155,
        'C4': [-123, 84, -113, 127, -20],
        'C5': [86, 239, 245]},
        {'C1': 28944,
        'C2': [7406345878083702537, 2475748031420128607],
        'C3': 14419363478921348170,
        'C4': [-124, 60, -49, -69, 118],
        'C5': [131, 189, 250]}],
'B4': -0.7602369045729906,
'B5': 77,
'B6': {'D1': -0.788330614566803, 'D2': -561302283, 'D3': -79, 'D4': 69},
'B7': 0.6781680366772436}

```

## 2. Двоичные данные:

```

(b'RRUT\xeeJ\x8d\xe1gw/o\x00\x03\x00\x00\x005\xa3\x90\x05\xaa\x8d\x90'
 b'\x89@\x00\x03\x00\x00\x00p\xbf\xc5\xc7\xb2\x96\x15\xf0\x00\xbf\x00\x00\x00'
 b'\xbb\xbf\xab,\x05\xfa]\xf7`j fj\x04\x05\xdb3\xb9\x0e\xef0\xf7)f\xca!\xaf\xd2'
 b'\x13\xa8\xadc\xb1\xb35\x8eb\xdc\x97\x92~f\x99\xd21"\x10\xa2\xe5\x16\xb1"'
 b'\x8d-#\xbfxbe%\xef\xe3F \x86\xd5t\xfbu\xa61\x00\x02\x008H\x00\x15\xccY\x1b'
 b'\x80\x820\xe8\xa8\xd0E\x00\x00\x00\x03\x00H\xd7\x98\x00\x02\x00K"'
 b'\x96\x08\x8d\xc6\x0c&1-\xea\x7f\xa5|\x00\x00\x00\x02\x00[\x9a@\x00\x02\x00]'
 b'.\x86R\xb4\nm7\xa0\xe5s%\x03\xa5\x00\x00\x00\x03\x00m?\x13?\xa2\x17'
 b'\x8f\x184\xf2<')

```

## Результат разбора:

```

{'A1': -1276051545163092113,
 'A2': {'B1': 'j fj',
        'B2': -6660817618803324608,
        'B3': [{'C1': 42545,
                  'C2': [-797044933113868561, 3528334354522681298],
                  'C3': 5188170738143297666,
                  'C4': [79, -24, -88, -48, 69],
                  'C5': [19, 168, 173]},
                {'C1': 55192,
                  'C2': [7183719923236134039, -7890756686260198896],
                  'C3': 2492188848802244145,
                  'C4': [45, -22, 127, -91, 124],
                  'C5': [162, 229]},
                {'C1': 39488,
                  'C2': [1635126129455213247, -4745122872951798059],
                  'C3': 3352457905846433696,
                  'C4': [-27, 115, 37, 3, -91],

```

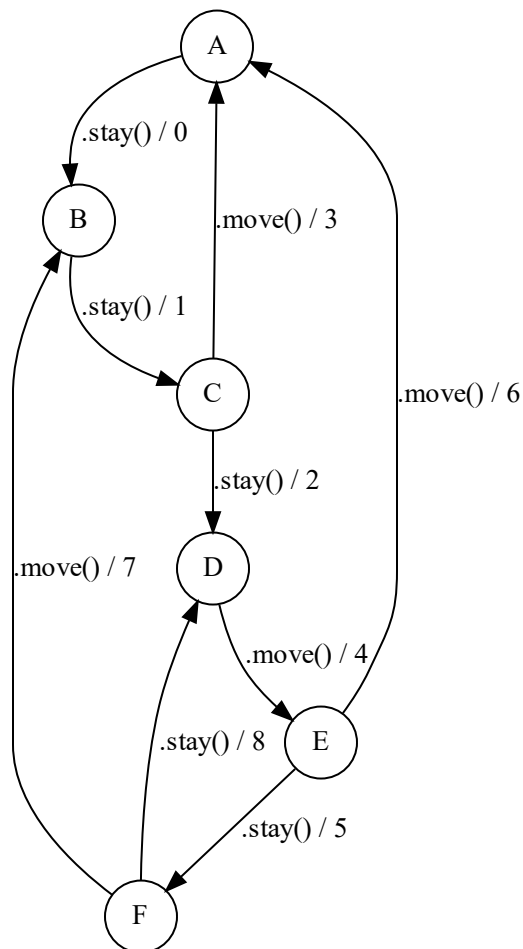


```

        'C5': [116, 251, 117]}],
    'B4': -0.17015678718951222,
    'B5': -65,
    'B6': {'D1': 0.5751897096633911, 'D2': 395253812, 'D3': -14, 'D4': 60},
    'B7': -0.053070246517548325}}

```

**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.stay()	0
o.stay()	1
o.stay()	2
o.move()	4
o.stay()	5
o.stay()	8
o.move()	4
o.move()	6
o.stay()	0
o.stay()	1
o.move()	3
o.stay()	0

## 2. Пример использования класса C32:

o = C32()	
o.stay()	0
o.stay()	1
o.stay()	2
o.stay()	RuntimeError
o.move()	4
o.stay()	5
o.stay()	8
o.stay()	RuntimeError
o.move()	4
o.move()	6
o.stay()	0
o.stay()	1
o.move()	3

## Вариант №31

**Задача 3.1.** Реализовать разбор двоичного формата данных (в духе формата WAD игры Doom или графического формата PNG). Данные начинаются с сигнатуры 0x4e 0x4f 0x4d 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	Адрес (uint16) структуры B
	3	Структура F
	4	int64

Структура B:	1	int16
	2	int16
	3	Структура C

Структура C:	1	Размер (uint32) и адрес (uint32) массива адресов (uint16) структур D
	2	Массив uint16, размер 3
	3	uint8
	4	int64
	5	Структура E
	6	int32

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

Структура E:	1	uint16
	2	uint64
	3	uint64

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

Примеры разбора двоичного формата с помощью функции f31:

1. Двоичные данные:

```
(b'NOMMB\x80\xde\x8fk\xed\xebT\x00\\\xd3f\x00\x00\x00\x04\x00\x00\x00\x8d'
b'\xb6g\xc9o\xa0\x0b\x88\xed\xab?\n\x87Q\x00\x00\x00\x02\x00 \xff\xc1G\xa2'
b'??fJ\x00\x00\x00\x04\x00,\x01\xad\xbd\xf5r\x18\x00\x00\x00\x02\x00:\x80#'
b'\xd6\x01=\xea\xa1\x1c\x00\x00\x00\x04\x00F\x00"\x000\x00<\x00J+\x9ea0'
b'\x00\x00\x00\x04\x00\x00\x00T#\xa2(oS\xd4\xef\xda\xa9)\xd5\x7f:\xf5,='
b'\xfd\x05\xe3Z\xb8\x910\x00\xdb\xf7\xb2\x07\xe7MD\xb23\xfb\xfd\xfa'
b'\r\x00\xf6\xdf\x0b')
```

Результат разбора:

```
{'A1': 4792074711094651732,
'A2': {'B1': 11166,
'B2': 24911,
'B3': {'C1': [{'D1': 0.5411272644996643, 'D2': [107, -85]},
{'D1': 0.7476545572280884, 'D2': [-1, -63, 71, -94]},
{'D1': -0.11984652280807495, 'D2': [1, -83]},
{'D1': 0.11456510424613953, 'D2': [-128, 35, -42, 1]}],
'C2': [9122, 10351, 21460],
'C3': 239,
'C4': -2690573305437817556,
'C5': {'E1': 15869,
'E2': 424282538661314779,
'E3': 17848336963237884467},
'C6': -67241459}},
'A3': {'F1': 54118, 'F2': [0, -10, -33, 11]},
'A4': -5303048554941806355}
```

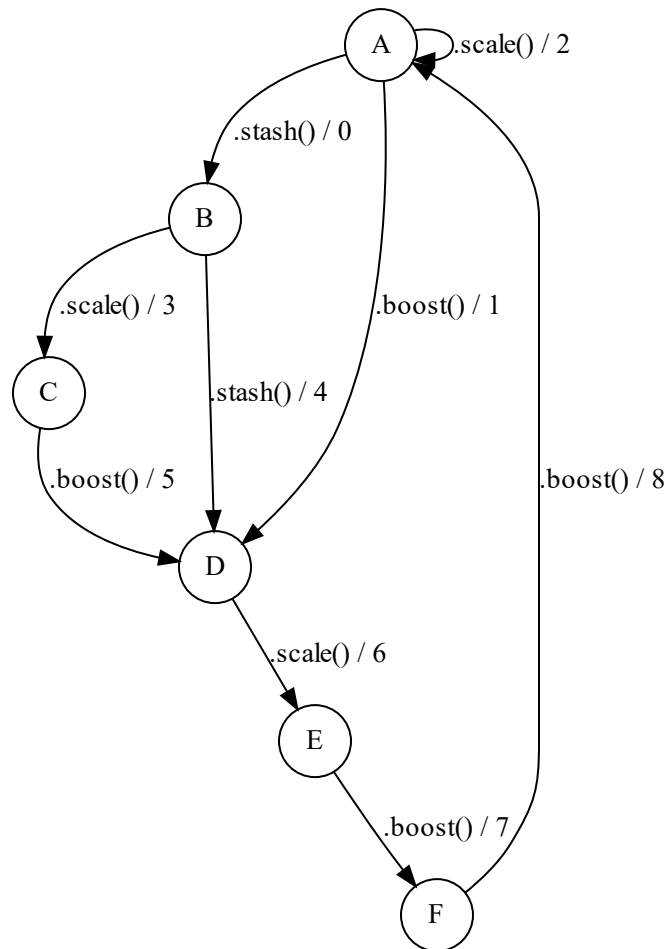
## 2. Двоичные данные:

```
(b'NOMM\xcb\xca\xf6R\xd8\x19.\xe1\x00m\x13\xd6\x00\x00\x00\x03\x00\x00\x00\x9e'
b'\xec\xba\xd6\x8f\x1f\xdbT\x89$\x89\x02A>\xdf`\x1f\x00\x00\x00\x04\x00 \xbaw'
b'?T~:\x00\x00\x00\x02\x00.s\xb7\xa3\xbe\x13\x85x\x00\x00\x00\x03\x00:\xaa'
b'\xd2\xf1o\xbf\b\x91\x1b\x00\x00\x00\x04\x00G\xb5\xaf;A?.\x96\xf0\x00\x00\x00'
b'\x04\x00U\x00$\x000\x00=\x00K\x00Y\x99\xe1Y\xbf\x00\x00\x00\x05\x00\x00\x00'
b'c\xc2(\xa1\x1aS\xaaY=\x9cAv\xbe\x9fR\xc0\xa0V\xea\xf3\xa6E\xf7\xefz\xfc^'
b'\x94\n\xb74\xfd&V#\xa0\xab\xa4\xafu')
```

## Результат разбора:

```
{'A1': -3761923703041347871,
'A2': {'B1': -26143,
'B2': 22975,
'B3': {'C1': [{'D1': 0.436280220746994, 'D2': [36, -119, 2, 65]},
{'D1': 0.830051064491272, 'D2': [-70, 119]},
{'D1': -0.1440638303756714, 'D2': [115, -73, -93]},
{'D1': -0.8850266337394714, 'D2': [-86, -46, -15, 111]},
{'D1': 0.6819906234741211, 'D2': [-75, -81, 59, 65]}],
'C2': [49704, 41242, 21418],
'C3': 89,
'C4': 4439495310940525248,
'C5': {'E1': 12448,
'E2': 6263086127276093306,
'E3': 18185135119112928550},
'C6': 1445175467}},
'A3': {'F1': 5078, 'F2': [-92, -81, 117]},
'A4': -1388561624887241591}
```

**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.stash()	0
o.scale()	3
o.boost()	5
o.scale()	6
o.boost()	7
o.boost()	8
o.scale()	2
o.scale()	2
o.boost()	1
o.scale()	6
o.boost()	7
o.stash()	RuntimeError
o.boost()	8

## 2. Пример использования класса C32:

o = C32()	
o.stash()	0
o.scale()	3
o.boost()	5
o.scale()	6
o.boost()	7
o.stash()	RuntimeError
o.boost()	8
o.scale()	2
o.stash()	0
o.stash()	4
o.scale()	6
o.boost()	7

## Вариант №32

**Задача 3.1.** Реализовать разбор двоичного формата данных (в духе формата WAD игры Doom или графического формата PNG). Данные начинаются с сигнатуры 0x54 0x4d 0x45 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	Адрес (uint16) структуры B
	2	Массив uint32, размер 2
	3	Адрес (uint16) структуры D

Структура B:	1	uint32
	2	Массив uint32, размер 3
	3	uint64
	4	int32
	5	Размер (uint32) и адрес (uint16) массива адресов (uint32) структур C
	6	Размер (uint16) и адрес (uint32) массива int32
	7	uint8

Структура C:	1	uint32
	2	uint64

Структура D:	1	float
	2	uint16

Примеры разбора двоичного формата с помощью функции f31:

1. Двоичные данные:

```
(b'TMEF\x00TJ\x9e-\xb1\xee$\xbe\xd9\x00}\xd8\xdf\xb8R\xf81\xe\x8e\xb19T\x88'
b'\xf6\xdf8\xe5\xf7\x86\x04\xe2t4\x91\x1c\xc0\xd7\xd2\xad\\\x90Y*\x99\xe9\x0fv'
b"\x00\x00\x00\x10\x00\x00\x00\x1c\x00\x00\x00(\x84'\x9b\xb5\xbf\xa94\x9c"
b'f\xaaMr5)\xc6\xff\x96\xf8\x1d\xf9\x7fD\x1b\xd7`T3\xbe\xac\xd7\xe8\xe0'
b'\x1b\x98\xd8\xf7\xde\xa2\x10d&Z\xf6\xac\x06Z\xdd\x14\x00\x00\x00\x03'
b'\x004\x00\x05\x00\x00\x00@=\xbeH\xbbLz\xf1')
```

Результат разбора:

```
{'A1': {'B1': 2135170007,
        'B2': [1616131006, 2899831008, 463001847],
        'B3': 16042402844973397676,
        'B4': 106618132,
        'B5': [{'C1': 3638540370, 'C2': 17884109193472464008},
                {'C1': 4141824229, 'C2': 17835948744906150172},
                {'C1': 3235369645, 'C2': 6669929087641390966}],
        'B6': [-2077779019, -1079429988, 1722436978, 891930367, -1762124295],
        'B7': 61},
```

```
'A2': [1251880369, 3995385561],  
'A3': {'D1': -0.1960269808769226, 'D2': 31473}}
```

## 2. Двоичные данные:

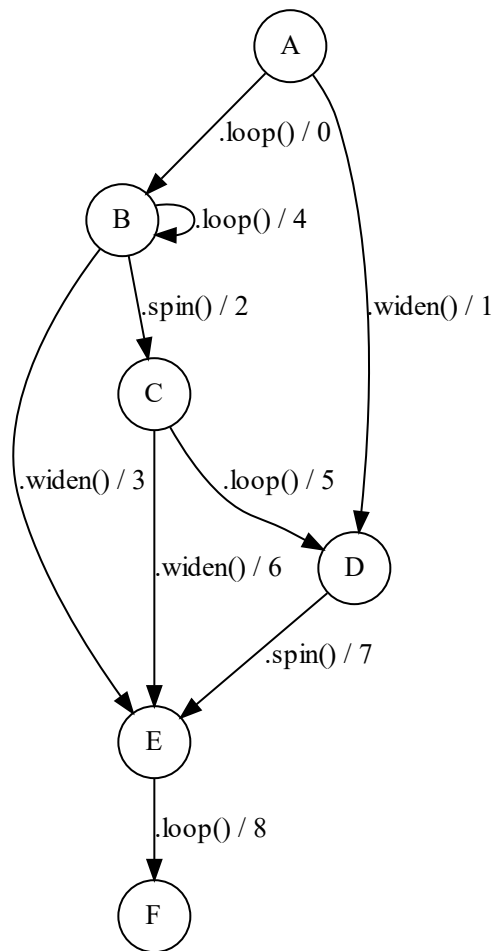
```
(b'TMEF\x00D\x93\x15b\x1b\xa9\xa7\xa3a\x00m\x81\x7f\xf2p$V&\xfc\xa9\r\x1c'  
b'~\xd3#\x05h\xd4\x0e\xe4\x07\x7f\xac\x00\x00\x00\x10\x00\x00\x00\x1c'  
b'|\x98\xd7\xfc\x1ezD\x93\xf8\xbc\xaf\nm\xa9\xa9\x81\xd9\x1d!\xff-*\xab\x88'  
b'\xf8\x9c\xa8\xc0\x9870*\xd9\x93\xae\x7fC\x1b1\x04\xe0Pr\x06=\x83J\xaf '  
b'\x00\x00\x00\x02\x00(\x00\x05\x00\x00\x00o>\xac\x07\x80\x11\x95')
```

## Результат разбора:

```
{ 'A1': { 'B1': 757771144,  
          'B2': [4171016384, 2553753642, 3650334335],  
          'B3': 4835512521925358086,  
          'B4': 1032014511,  
          'B5': [{ 'C1': 2172646000, 'C2': 2618323099972344957},  
                  { 'C1': 2127766315, 'C2': 389794528187875244}],  
          'B6': [2090391548, 511329427, -121852150, 1839835521, -652402177],  
          'B7': 111},  
  'A2': [2467652123, 2846335841],  
  'A3': { 'D1': 0.3359947204589844, 'D2': 4501}}
```

**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.spin()</code>	<code>RuntimeError</code>
<code>o.loop()</code>	0
<code>o.loop()</code>	4
<code>o.spin()</code>	2
<code>o.loop()</code>	5
<code>o.spin()</code>	7
<code>o.spin()</code>	<code>RuntimeError</code>
<code>o.loop()</code>	8

2. Пример использования класса C32:

<code>o = C32()</code>	
<code>o.loop()</code>	0
<code>o.loop()</code>	4
<code>o.spin()</code>	2
<code>o.loop()</code>	5
<code>o.loop()</code>	RuntimeError
<code>o.spin()</code>	7
<code>o.spin()</code>	RuntimeError
<code>o.loop()</code>	8

## Вариант №33

**Задача 3.1.** Реализовать разбор двоичного формата данных (в духе формата WAD игры Doom или графического формата PNG). Данные начинаются с сигнатуры 0x42 0x59 0x50 0x5a 0x93, за которой следует структура А. Порядок байт: от старшего к младшему. Адреса указаны в виде смещений от начала данных. В решении разрешено использовать модуль struct.

Структура А:	1	int64
	2	float
	3	Адрес (uint32) структуры В
	4	Структура D
	5	float
	6	int16

Структура В:	1	Размер (uint32) и адрес (uint32) массива char
	2	int16
	3	Массив структур С, размер 2
	4	Размер (uint16) и адрес (uint16) массива int64
	5	double
	6	Размер (uint16) и адрес (uint32) массива double

Структура С:	1	int32
	2	int8
	3	float
	4	float

Структура D:	1	uint64
	2	Массив uint16, размер 3
	3	float

Примеры разбора двоичного формата с помощью функции f31:

1. Двоичные данные:

```
(b'BYRZ\x93\xaf\x8d^\x88\xf6?+\x0c\xa6\x00\x00\x00R{\xe7\xec\xe3\x83\xd5\xf9'
b'\x17R\x9b^\x02\xba\xdc\xbfI\x90\x92>D\xd7\xff\x86!ttvrvRJ\xad\xf4|('
b'\x8b\xf3\xb2r\x11\xa0\xf2\xc5!W?\xe0I\xc0;\xe7\xdc\xf0\xbf\xe5\xe7\xc9c\xe2'
b'\xa96\x00\x00\x00\x05\x00\x00\x00-F\x8fR\xae\x17\xf3^\xbfHJ1\xbe\x91\xd9'
b'\x8a\x93\xdf\x03$\xfa>\xf4{.\xbf\x04\xd5A\x00\x02\x002\xbf\xee\x1c\xf0\n '
b'\xea\xbe\x00\x02\x00\x00\x00B')
```

Результат разбора:

```
{'A1': -5796873601019574026,
'A2': 0.6681617498397827,
'A3': {'B1': 'ttvrv',
```

```

'B2': 18063,
'B3': [{ 'C1': 1387141107,
        'C2': 94,
        'C3': -0.782382071018219,
        'C4': -0.28486281633377075},
      { 'C1': -1814101212,
        'C2': -6,
        'C3': 0.4775022864341736,
        'C4': -0.5188789963722229}],
'B4': [5929743124953336819, -5588384804647919273],
'B5': -0.9410324285339227,
'B6': [0.5090027971299644, -0.6845442725073443]},
'A4': { 'D1': 8928365248198867223,
        'D2': [21147, 24066, 47836],
        'D3': -0.7873622179031372},
'A5': 0.1922302097082138,
'A6': -31199}

```

## 2. Двоичные данные:

```

(b'BYPZ\x93%\x99\x8d\x08q\x13W\xbfBf\xb9\x00\x00\x00cM\xd5%\xf9\xdd\x8bv'
b'\xec#\n\x1a\xe7\xab\xbfj5\x00?\x07\xd7d\xe2\x93aopcwz\xaf\x12\x08a\xdb'
b'\xdc\x0b\xdb\xc9\xe6\x13s+\xcf\xfd\x91\x80\x96\xd2<\xf46\x81\x18'
b'1\x85\xd6\xf9 6D?\xec\xd7\x8b\x0cA5\x82\xbf\xe8\x15\xbb\xa5\xd9\xd2f\x00'
b'\x00\x00\x06\x00\x00\x00-.qM\x17\x8f87>\xcee\xbe\xbe\xd3ga\xceX'
b'\x89\xa2\x97\xbf4K\xb3\xbe\xf9\xd9\xb4\x00\x04\x003?\xed\x87\xf6\xb8'
b'\xa0G`\x00\x02\x00\x00\x005')

```

## Результат разбора:

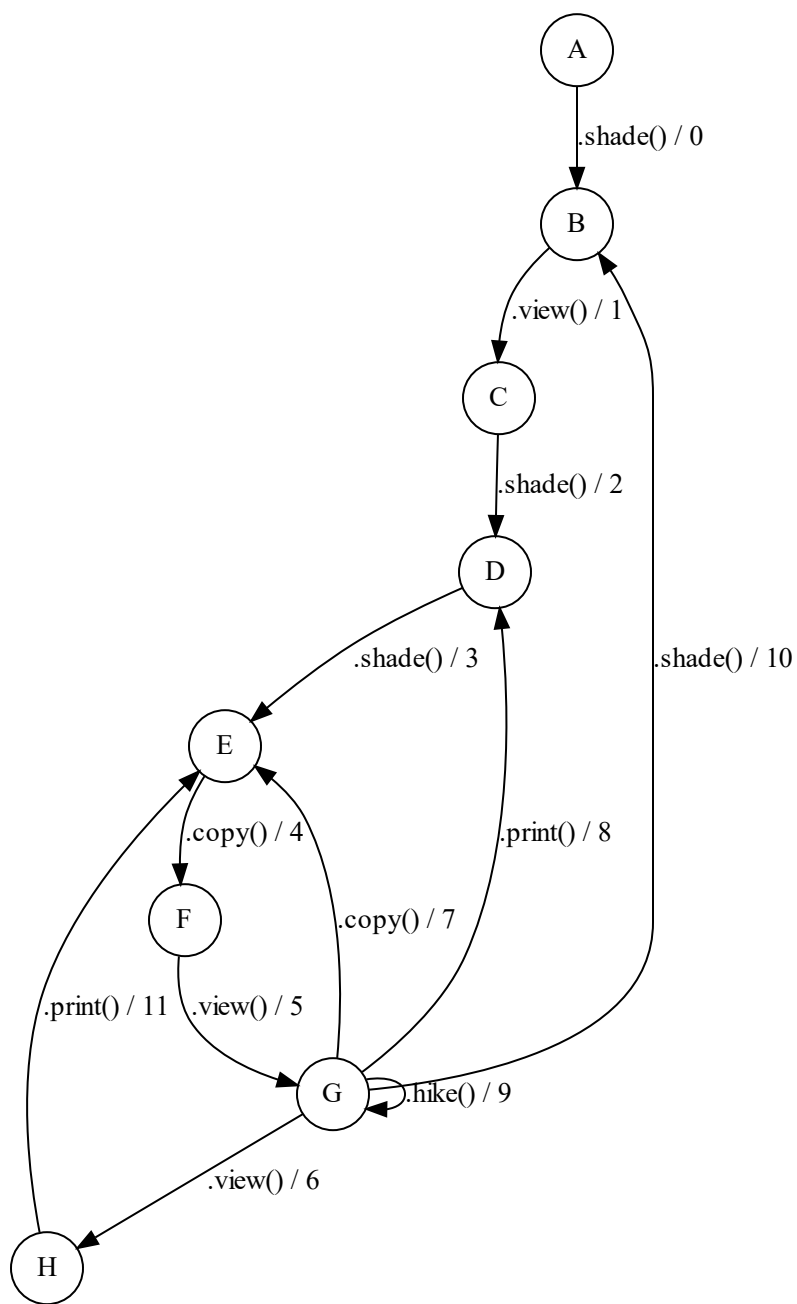
```

{ 'A1': 2689662233481384791,
  'A2': -0.7593799233436584,
  'A3': { 'B1': 'aopcwz',
          'B2': 11889,
          'B3': [{ 'C1': 1293389624,
                    'C2': 55,
                    'C3': 0.4031199812889099,
                    'C4': -0.41289809346199036},
                { 'C1': -833058398,
                    'C2': -105,
                    'C3': -0.7042800784111023,
                    'C4': -0.48798906803131104}],
          'B4': [-5831589351097889829,
                  -3898407042065367663,
                  -9180919634684791167,
                  1743321689118225988],

```

```
'B5': 0.9228471380213072,  
'B6': [0.90131142037869, -0.7526529540323963]},  
'A4': {'D1': 5608430666053809900,  
       'D2': [9052, 2586, 59307],  
       'D3': -0.9148712158203125},  
'A5': 0.530630350112915,  
'A6': -7533}
```

**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.shade()	0
o.view()	1
o.print()	RuntimeError
o.shade()	2
o.copy()	RuntimeError
o.shade()	3
o.copy()	4
o.view()	5
o.copy()	7
o.copy()	4
o.view()	5
o.hike()	9
o.view()	6
o.print()	11
o.copy()	4
o.view()	5
o.shade()	10

## 2. Пример использования класса C32:

o = C32()	
o.shade()	0
o.view()	1
o.shade()	2
o.shade()	3
o.copy()	4
o.view()	5
o.print()	8
o.shade()	3
o.copy()	4
o.view()	5
o.view()	6
o.print()	11
o.copy()	4
o.view()	5
o.hike()	9
o.shade()	10

## Вариант №34

**Задача 3.1.** Реализовать разбор двоичного формата данных (в духе формата WAD игры Doom или графического формата PNG). Данные начинаются с сигнатуры 0x55 0x54 0x56 0x53, 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	int16
	3	Размер (uint32) и адрес (uint32) массива структур Е
	4	Структура F

Структура В:	1	uint32
	2	Массив char, размер 7
	3	Адрес (uint32) структуры С
	4	uint32

Структура С:	1	double
	2	int8
	3	int16
	4	Адрес (uint16) структуры D

Структура D:	1	int16
	2	uint8

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

Структура F:	1	double
	2	float
	3	uint64

Примеры разбора двоичного формата с помощью функции f31:

1. Двоичные данные:

```
(b'UTVS\xa9\xc2\xd6#ecnhcsu\x00\x00\x008\xf1\xb1\xfa\xd7\t\x00\x00\x00'
 b'\x06\x00\x00\x00Q?\xe8\xb7\xde0\xd4v\xd0\xbe@F\xef\x1d\x80=N\xd7\x8f\xff'
 b'\x9f+\xa8a\x0f?\xc2\x93\x1b#\xeadx\x96}\x1b\x005\xdf\xe2`\xad\xebi\x7f'
 b'\xce\x11?\xa7g\x00\x00\x00\x02\x00E\xee\x1dM\x9c\x00\x00\x00\x02\x00'
 b'G\xbb\x8d\x9e\xfd\x00\x00\x00\x02\x00I\xb1q\xf3B\x00\x00\x00\x02\x00'
 b'K\xa3\xf5\xaf\xbe\x00\x00\x00\x02\x00ME\xb4\x92P\x00\x00\x00\x02\x00'
 b'O\x0f\x8f5\xf6')
```

Результат разбора:



```
{'A1': {'B1': 2848118307,
        'B2': 'ecnhcsu',
        'B3': {'C1': 0.14511431935969532,
                'C2': -106,
                'C3': 32027,
                'C4': {'D1': 11146, 'D2': 15}},
        'B4': 4054939898},
'A2': -10487,
'A3': [{'E1': [-33, -30], 'E2': -18, 'E3': [29, 77, 156]},
        {'E1': [96, -83], 'E2': -69, 'E3': [141, 158, 253]},
        {'E1': [-21, 105], 'E2': -79, 'E3': [113, 243, 66]},
        {'E1': [127, -50], 'E2': -93, 'E3': [245, 175, 190]},
        {'E1': [17, 63], 'E2': 69, 'E3': [180, 146, 80]},
        {'E1': [-89, 103], 'E2': 15, 'E3': [143, 53, 246]}],
'A4': {'F1': 0.7724448737314678,
        'F2': -0.18777059018611908,
        'F3': 2125766432952156063}}
```

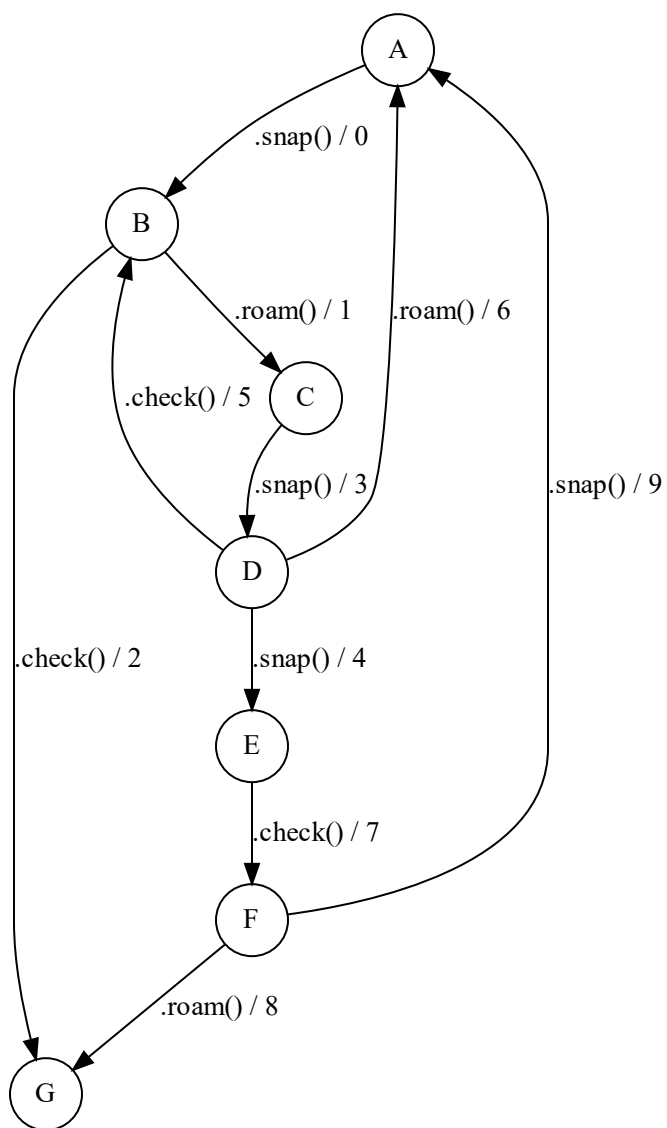
## 2. Двоичные данные:

```
(b"UTVSE\xfb'\x1ezikdqos\x00\x00\x008\xe3R\x8d9\xc4]\x00\x00\x00"
b'\x03\x00\x00\x00K?\xa6/0\xb5\xf4\xcd \xbeF\xf3:\xb9P"\xd0\xd44\xd7'
b'\xbd\xc4\xb1\x9b?\xe1\xe7\x07\x96\xf6\x88\xfa\x11\xec\x89\x005}q\xae'
b"\x92\xea'\x00\x00\x00\x02\x00EK\t\x88\x07\x00\x00\x00\x02\x00G1]uE\x00"
b'\x00\x00\x02\x00I\x11\xd1Pj')
```

## Результат разбора:

```
{'A1': {'B1': 1174087454,
        'B2': 'zikdqos',
        'B3': {'C1': 0.5594518612381585,
                'C2': 17,
                'C3': -4983,
                'C4': {'D1': -15183, 'D2': 155}},
        'B4': 3813838137},
'A2': -15267,
'A3': [{'E1': [125, 113], 'E2': 75, 'E3': [9, 136, 7]},
        {'E1': [-82, -110], 'E2': 49, 'E3': [93, 117, 69]},
        {'E1': [-22, 39], 'E2': 17, 'E3': [209, 80, 106]}],
'A4': {'F1': 0.04332970710893336,
        'F2': -0.19428721070289612,
        'F3': 13353211175462295485}}
```

**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.snap()          0
o.roam()          1
o.snap()          3
o.check()         5
o.roam()          1
o.snap()          3
o.roam()          6
o.snap()          0
o.snap()          RuntimeError
o.roam()          1
o.snap()          3
o.snap()          4
o.check()         7
o.check()         RuntimeError
o.roam()          8

```

## 2. Пример использования класса C32:

```

o = C32()
o.snap()          0
o.roam()          1
o.snap()          3
o.roam()          6
o.snap()          0
o.roam()          1
o.snap()          3
o.snap()          4
o.check()         7
o.snap()          9
o.snap()          0
o.check()         2

```

## Вариант №35

**Задача 3.1.** Реализовать разбор двоичного формата данных (в духе формата WAD игры Doom или графического формата PNG). Данные начинаются с сигнатуры 0x57 0x55 0x50 0x71, 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	Размер (uint32) и адрес (uint16) массива структур В
	3	double
	4	Структура С
	5	Адрес (uint32) структуры Е

Структура В:	1	int8
	2	uint8
	3	float

Структура С:	1	uint16
	2	Структура D

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

Структура Е:	1	Массив uint16, размер 2
	2	Массив int32, размер 2
	3	int8
	4	int8
	5	uint64

Примеры разбора двоичного формата с помощью функции f31:

### 1. Двоичные данные:

```
(b'WUPq\xd0r\xa5\xc1\x00\x00\x00\x02\x002\xbf\xd8k\xf8\x002\xa8@\xcb\xcd'
b'.\x96\xeb\x85\x08_\x99r\x00\x00\x00\x03\x00>\xe6D\xebQ\xdf\xde'
b'\x02\x1a\x00\x00\x00J\xc6\xc8?\x10\xb8\xc8\xccv=Y\xd9!\xcf\xee~\xc9\x16y'
b'\xef\xb3\xd3mfN E\x13\x8c\x0f\xf1P\xf0\xe0\x82\xaa"]\x9aq\xfc\x1b@'
b'\x8ba\xdd\xa7')
```

Результат разбора:

```
{'A1': 3497174465,
'A2': [{'B1': -58, 'B2': 200, 'B3': 0.5653195381164551},
{'B1': -52, 'B2': 118, 'B3': 0.053185585886240005}],
'A3': -0.3815898897106571,
```

```
'A4': {'C1': 52173,
      'C2': {'D1': 3357129528854944114,
            'D2': [3488513737, 377089971, 3547162190],
            'D3': 16592645664020496922}},
'A5': {'E1': [8261, 5004],
      'E2': [267473136, -528307678],
      'E3': 93,
      'E4': -102,
      'E5': 8213469784447245735}}
```

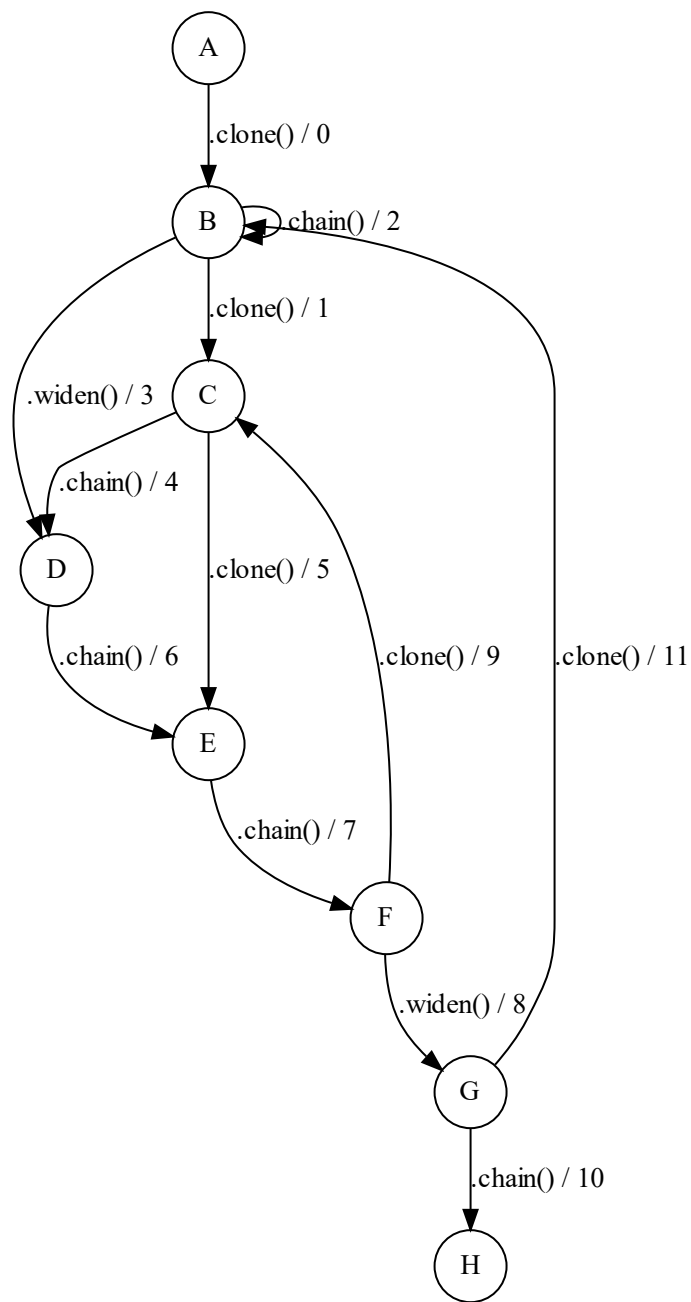
## 2. Двоичные данные:

```
(b'WUPq\xea\x86q\xc4\x00\x00\x00\x03\x002?\xd5<\x87\x01\xa8^\x14\xfb\xc6'
b'<8\xb2\xc5\xe0\xf8(*\x00\x00\x00\x03\x00D\xc9/A\xed0\xf7\x06+\x00\x00'
b'\x00P\xa4q?\tk*\xf9\xfd=\x137\x84\xea\xca>\xb5M\r"\xb9\xb9\xfb\xe1\x9a1\x8c'
b'\xd5\xc2\xa8#\xf9\x8fJ\xd7\xc8\x8c\xac\xbe\xaf\xd6\x1e~(\xeb\x86\x87'
b'\xb0\xa7\xea\x89F\xd7')
```

## Результат разбора:

```
{'A1': 3934679492,
 'A2': [{'B1': -92, 'B2': 113, 'B3': 0.5363031625747681},
        {'B1': -97, 'B2': 253, 'B3': 0.03594161570072174},
        {'B1': -22, 'B2': 202, 'B3': 0.3541034758090973}],
 'A3': 0.33181929742701866,
 'A4': {'C1': 64454,
      'C2': {'D1': 4339414803924133930,
            'D2': [582597115, 3784978828, 3586304035],
            'D3': 14496878212513465899}},
 'A5': {'E1': [63887, 19159],
      'E2': [-930304834, -1344921986],
      'E3': 40,
      'E4': -21,
      'E5': 9693910958179174103}}
```

**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.clone()	0
o.clone()	1
o.widen()	RuntimeError
o.clone()	5
o.chain()	7
o.chain()	RuntimeError
o.widen()	8
o.clone()	11
o.chain()	2
o.widen()	3
o.chain()	6
o.chain()	7
o.clone()	9
o.chain()	4
o.chain()	6
o.chain()	7
o.widen()	8
o.widen()	RuntimeError
o.chain()	10

## 2. Пример использования класса C32:

o = C32()	
o.chain()	RuntimeError
o.clone()	0
o.clone()	1
o.clone()	5
o.widen()	RuntimeError
o.chain()	7
o.widen()	8
o.clone()	11
o.chain()	2
o.widen()	3
o.chain()	6
o.chain()	7
o.clone()	9
o.clone()	5
o.chain()	7
o.widen()	8
o.chain()	10

## Вариант №36

**Задача 3.1.** Реализовать разбор двоичного формата данных (в духе формата WAD игры Doom или графического формата PNG). Данные начинаются с сигнатуры 0xd4 0x41 0x55 0x51, 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	int64
	2	uint32
	3	Адрес (uint32) структуры В
	4	Адрес (uint32) структуры F
	5	uint32
	6	uint16

Структура В:	1	Массив адресов (uint16) структур С, размер 4
	2	Структура D

Структура С:	1	uint64
	2	int8

Структура D:	1	Массив uint16, размер 8
	2	Размер (uint32) и адрес (uint16) массива float
	3	uint32
	4	Адрес (uint32) структуры E
	5	Размер (uint16) и адрес (uint16) массива uint8
	6	uint64
	7	int8

Структура E:	1	uint32
	2	uint16
	3	int32

Структура F:	1	uint32
	2	double
	3	int8

Примеры разбора двоичного формата с помощью функции f31:

### 1. Двоичные данные:

```
(b'\xd4AUQ\x17d\xb3\xd8J\x10\xe8\x02_\x96L\xb3X\x00\x00\x00\x8b\x00\x00\x00'  
b'$\x83x4G\xbc\xa0K\xdf\xdd\xa1\x89}\xcd\r\xff-6\x04\xfe\x85\xa2x\x15'  
b'\x8b\xc7@\xfa\xfdS\xe0\x1f\xd040\x85>M\xb9\x0c*\xc3!Y\x99\xbe\xb1\xa4'  
b"\xc4\xbe\x9b~\xbb5\xeeXs\x00\x84\xa'\xac\xa1}\x1e\x00'\x000\x009\x00"  
b'\xe0\x1em\xfc\xc1AT<!C\xad\t<B\x17i\x02\x00\x00\x00B\x00\xaa\xf8'  
b'\xad\x18J\x00\x00\x00\x04\x00T\x00\xc7\x8b\xd7\x92VAJC\xd0\xd7\xd4L\x01\xcc'  
b'\xf3\x05\x90\x05\xe2\xe7\xbf\x0e')
```



Результат разбора:

```
{'A1': 209435296321987607,  
'A2': 3008140895,  
'A3': {'B1': [{'C1': 14807142478169197472, 'C2': 114},  
              {'C1': 8692657556817325567, 'C2': 21},  
              {'C1': 2296928160249333643, 'C2': -48},  
              {'C1': 3030000390726045492, 'C2': -61}],  
'B2': {'D1': [7904, 64621, 16833, 15444, 17185, 2477, 16956, 26903],  
       'D2': [-0.29950812458992004, -0.3840689957141876],  
       'D3': 414054570,  
       'D4': {'E1': 901480091, 'E2': 22766, 'E3': -1971060621},  
       'D5': [39, 172, 161, 125],  
       'D6': 4848759788904352711,  
       'D7': -48}},  
'A4': {'F1': 21812439, 'F2': -0.7463405430750014, 'F3': 14},  
'A5': 880313124,  
'A6': 48199}
```

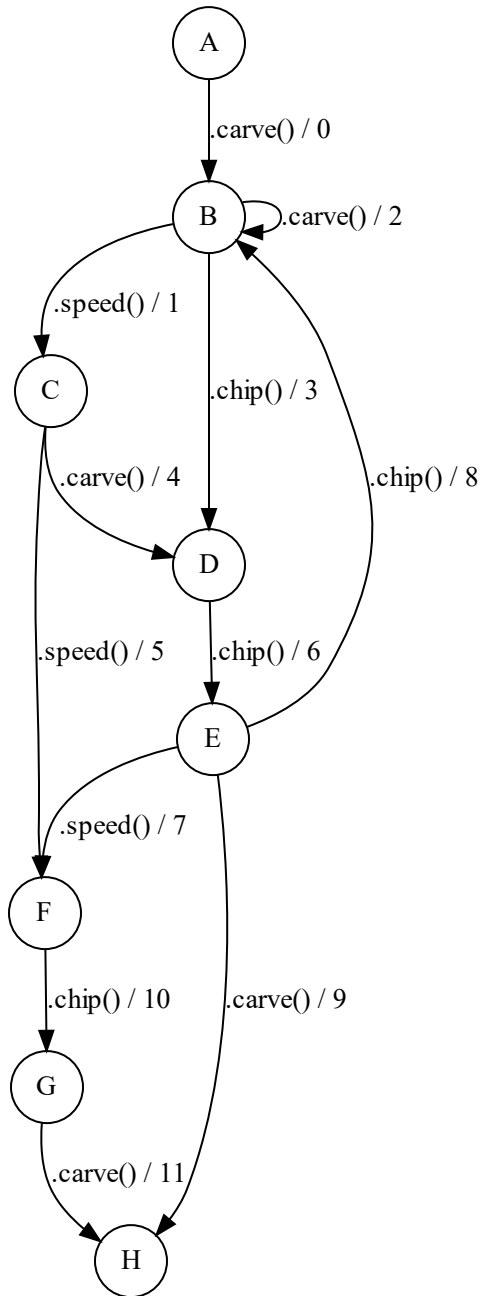
## 2. Двоичные данные:

```
(b'\xd4AUQ\xcb\x04q\xea\x81\x8bR|\xb5I\x07\xebV\x00\x00\x00\x89\x00\x00\x00'  
b'\x899\xfav\xb2\xbc\xc2\xc6\xe8_\xc1\xcd\x0b_\x8f\xa5\xe3a\x08'  
b"\x84\x03\xc0\x1c\x14\x81\x0c!Zb\xac/X su\xde\x93Smg\x1b\xdf'V?\xb8\x90"  
b"\x9c\xbd\xe6\x12\xcf\xbb6e\x9f\xe3U\xc8I\x04\x1e\x00'\x000\x009\x00\xca"  
b"\x10\xca\xecob\x03,\xf0'\xac\x10&\xbc\x02\x00\x00\x00B\x00\xdf\xfa9"  
b'J\x00\x00\x00\x02\x00T\x00\x86\xd0\xc8-\x12\xfd\x04\x82\x0c\xce\x03yC8\xecb'  
b'R\xcc\x18\xe8?\xfa')
```

Результат разбора:

```
{'A1': 8958375998893655243,  
'A2': 3943123381,  
'A3': {'B1': [{'C1': 850548522314285756, 'C2': 95},  
              {'C1': 13836047651749209487, 'C2': 28},  
              {'C1': 3435228755017892116, 'C2': 88},  
              {'C1': 7452704852925575968, 'C2': 27}],  
'B2': {'D1': [31178, 4192, 60618, 25199, 11267, 10224, 4268, 48166],  
       'D2': [0.8365458846092224, -0.07644790410995483],  
       'D3': 963245279,  
       'D4': {'E1': 3150910182, 'E2': 25910, 'E3': -933895265},  
       'D5': [73, 4],  
       'D6': 9368891379356848262,  
       'D7': 12}},  
'A4': {'F1': 1132004302, 'F2': 0.7530271157369333, 'F3': -6},  
'A5': 4198074720,  
'A6': 45686}
```

**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.carve()</code>	0
<code>o.carve()</code>	2
<code>o.chip()</code>	3
<code>o.chip()</code>	6
<code>o.chip()</code>	8
<code>o.speed()</code>	1
<code>o.carve()</code>	4
<code>o.chip()</code>	6
<code>o.speed()</code>	7
<code>o.chip()</code>	10
<code>o.carve()</code>	11

## 2. Пример использования класса C32:

<code>o = C32()</code>	
<code>o.carve()</code>	0
<code>o.carve()</code>	2
<code>o.chip()</code>	3
<code>o.chip()</code>	6
<code>o.chip()</code>	8
<code>o.speed()</code>	1
<code>o.carve()</code>	4
<code>o.chip()</code>	6
<code>o.speed()</code>	7
<code>o.chip()</code>	10
<code>o.chip()</code>	RuntimeError
<code>o.carve()</code>	11

## Вариант №37

**Задача 3.1.** Реализовать разбор двоичного формата данных (в духе формата WAD игры Doom или графического формата PNG). Данные начинаются с сигнатуры 0x50 0x42 0x5a 0x44 0x52, за которой следует структура А. Порядок байт: от младшего к старшему. Адреса указаны в виде смещений от начала данных. В решении разрешено использовать модуль struct.

Структура А:	1	Адрес (uint16) структуры В
	2	double
	3	uint8
	4	Размер (uint16) и адрес (uint32) массива uint16
	5	float
	6	uint8
	7	uint16

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

Структура С:	1	int32
	2	uint16
	3	int32

Структура D:	1	int32
	2	Массив uint64, размер 8

Примеры разбора двоичного формата с помощью функции f31:

### 1. Двоичные данные:

```
(b'PBZDR;\x00B\xcf\xcau\xac\xb5\xeb?\xee\x02\x00\x97\x00\x00\x00\xf8\x17'
b'\x05?\x0f\x01\xe6\xfd\xae\xd9"\x8d[\xab\xfa,\xa0\xaa\xfa\xc8\xe8\xa0'
b'\xc1[\xd8q\xb3\x17\xd1\xd8\x0b\xc1\x15=\r\xe1\xaf\x88\xdf\x1e-\xcf\x12N\xdcq'
b'wrvxcb\x03\x00\x00\x00\x1d\x00$RX\x10\x07D:\x96\xef\x10\xa2\xfe'
b'\xff\x89\x9f\x06n\xc5n\x1e\xadb\xf7nwR\x13\x18\xddX\xad\xce\xa8a\xd5}'
b'\x0fa\xd3\xf4\xdb\xc9\xb6C1\\xea\xbe\x0c\xfb\xbb\xe7\x93\xaa-Jy\x05\r\xb0'
b'|\xfa\xb1\xc9T#\x82\xfe=\xe5\xa5<\x1f(')
```

Результат разбора:

```
{'A1': {'B1': 756998024,
```

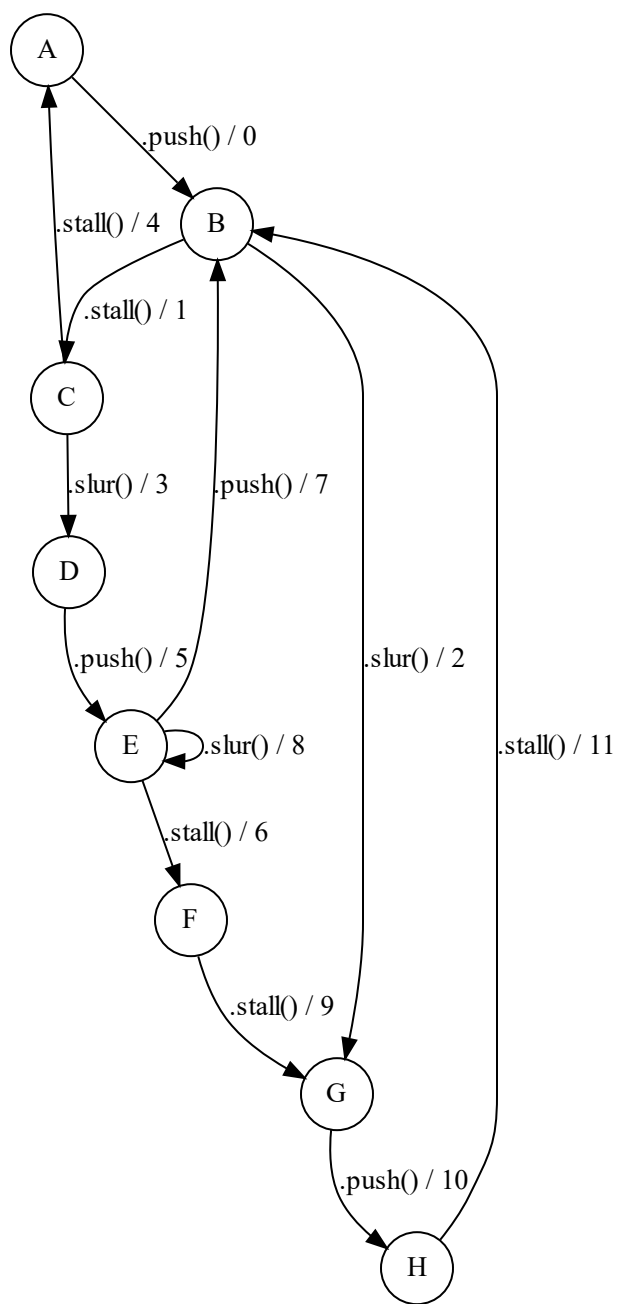


```

'D2': [17524086371039117006,
       12905841302362153797,
       2952687879515128896,
       5124457079676812327,
       13010879286448639483,
       3078165165672606314,
       17305935784653806203,
       14634254856987511001]}},
'A2': 0.11546519561310142,
'A3': 155,
'A4': [44544, 36102],
'A5': -0.899134635925293,
'A6': 125,
'A7': 22484}

```

**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.push()          0
o.stall()         1
o.stall()         4
o.push()          0
o.slur()          2
o.push()          10
o.slur()          RuntimeError
o.stall()         11
o.stall()         1
o.slur()          3
o.push()          5
o.slur()          8
o.stall()         6
o.stall()         9
o.push()          10

```

## 2. Пример использования класса C32:

```

o = C32()
o.push()          0
o.stall()         1
o.slur()          3
o.stall()         RuntimeError
o.push()          5
o.slur()          8
o.stall()         6
o.stall()         9
o.push()          10
o.stall()         11
o.slur()          2
o.push()          10
o.stall()         11
o.stall()         1
o.push()          RuntimeError
o.stall()         4

```



## Вариант №38

**Задача 3.1.** Реализовать разбор двоичного формата данных (в духе формата WAD игры Doom или графического формата PNG). Данные начинаются с сигнатуры 0x5a 0x42 0x53 0x51, за которой следует структура А. Порядок байт: от старшего к младшему. Адреса указаны в виде смещений от начала данных. В решении разрешено использовать модуль struct.

Структура А:	1	Размер (uint32) и адрес (uint16) массива адресов (uint16) структур В
	2	Структура С

Структура В:	1	int32
	2	uint16

Структура С:	1	uint16
	2	Размер (uint32) и адрес (uint32) массива char
	3	Адрес (uint16) структуры D
	4	uint32
	5	Размер (uint32) и адрес (uint16) массива int16
	6	uint16
	7	int8

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

Примеры разбора двоичного формата с помощью функции f31:

### 1. Двоичные данные:

```
(b'ZBSQ\x00\x00\x00\x03\x005M\xb9\x00\x00\x00\x06\x00\x00\x00;\x00Gm\xf2'
b'\xd23\x00\x00\x00\x07\x00}\xbb\xf87\xebQ\x13\xf2\xb1\n\xefQb)\xf1\r\xe5'
b"\xdb\xa6\xb3\x19\xf3\x00#\x00)\x00/vitxrwL%b\xac'6?\xcc\xe7[.\x9a\xd4\xf0V"
b'\xac\xf3\xdc\xdb\x87\xba\x06\xdac\xf1\x85\x05\xe6`\x16\xdc\xb7\xaf\xbf\xd0'
b'n\xb7\x19&)\xda(D\xcf/\x10\x8b\xd0<cRu\x81\xec\x00\x03\x00\x00\x00A#\x95\xc2'
b'k\xa9u\xc0\x8ak\x1f\xc1f\xc\x9c')
```

Результат разбора:

```
{'A1': [{'B1': -347008014, 'B2': 45322},
        {'B1': -279879127, 'B2': 61709},
        {'B1': -438589773, 'B2': 6643}],
'A2': {'C1': 19897,
       'C2': 'vitxrw',
```

```

'C3': {'D1': 0.22581043029428072,
      'D2': [1454175196,
              3683105286,
              3663982981,
              98983958,
              3703025599,
              3496916761,
              640277032],
      'D3': 17615,
      'D4': 12048,
      'D5': -8372125309998890516,
      'D6': [19493, 25260, 10038]},
'C4': 1844630067,
'C5': [9109, -15765, -22155, -16246, 27423, -16026, -868],
'C6': 48120,
'C7': 55}}

```

## 2. Двоичные данные:

```

(b'ZBSQ\x00\x00\x00\x02\x00/\x19\x05\x00\x00\x00\x02\x00\x00\x003\x00;\x8b\x7f'
 b'I\xc8\x00\x00\x00\x07\x00qw\xc0<K\xddK\xa9\x0b6[$\x874R}\x00#\x00)m'
 b'h\xaa\xde\xc6\xd0b\x10?\xd5\xb0\xdcT\xd9t<\x8b\x1b\x86\x9f\x7f:\xa5\x9c\xab'
 b'\x93H\x17\x1e\xfd\xa4\x8fD\x80\xaba\xdf\x86B\xb7j\xdd\xdb\x17\xaf'
 b'\x93\xce\xd7A\x14o\xac\xf8\xda;\x8e\x00\x03\x00\x00\x005\xe4\xd3\x19'
 b'\xce=\xdd\xb7A\x1a\xad\xf4\xf7&\x16')

```

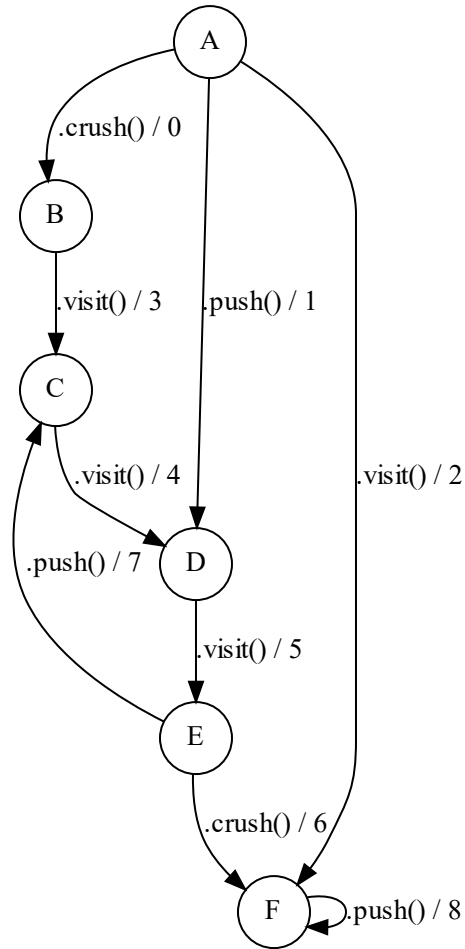
## Результат разбора:

```

{'A1': [{'B1': 1272794025, 'B2': 2870}, {'B1': 1529120564, 'B2': 21117}],
'A2': {'C1': 6405,
      'C2': 'mh',
      'C3': {'D1': 0.3389197186095918,
            'D2': [2333836959,
                    2134549916,
                    2878556183,
                    519939215,
                    1149283169,
                    3750118071,
                    1792924439],
            'D3': 44947,
            'D4': 52951,
            'D5': 4689495900699638670,
            'D6': [43742, 50896, 25104]},
      'C4': 2340374984,
      'C5': [-6957, 6606, 15837, -18623, 6829, -2825, 9750],
      'C6': 30656,
      'C7': 60}}

```

**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.crush()</code>	0
<code>o.visit()</code>	3
<code>o.visit()</code>	4
<code>o.visit()</code>	5
<code>o.push()</code>	7
<code>o.crush()</code>	RuntimeError
<code>o.visit()</code>	4
<code>o.visit()</code>	5
<code>o.crush()</code>	6
<code>o.push()</code>	8
<code>o.push()</code>	8

## 2. Пример использования класса C32:

<code>o = C32()</code>	
<code>o.crush()</code>	0
<code>o.visit()</code>	3
<code>o.visit()</code>	4
<code>o.visit()</code>	5
<code>o.push()</code>	7
<code>o.visit()</code>	4
<code>o.visit()</code>	5
<code>o.crush()</code>	6
<code>o.crush()</code>	RuntimeError
<code>o.push()</code>	8
<code>o.push()</code>	8
<code>o.visit()</code>	RuntimeError
<code>o.push()</code>	8
<code>o.push()</code>	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 0x51 0x4d 0x4c 0x46, 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	Структура В

Структура В:	1	uint64
	2	uint8
	3	uint16
	4	double
	5	Структура С
	6	double

Структура С:	1	uint64
	2	uint16
	3	Массив структур D, размер 2
	4	Адрес (uint16) структуры Е
	5	uint64
	6	int8
	7	int64

Структура D:	1	uint8
	2	int16
	3	Массив int32, размер 5

Структура Е:	1	double
	2	int16
	3	Размер (uint16) и адрес (uint16) массива uint8
	4	Массив uint8, размер 7
	5	Размер (uint16) и адрес (uint16) массива int16
	6	uint16
	7	uint64

Примеры разбора двоичного формата с помощью функции f31:

1. Двоичные данные:

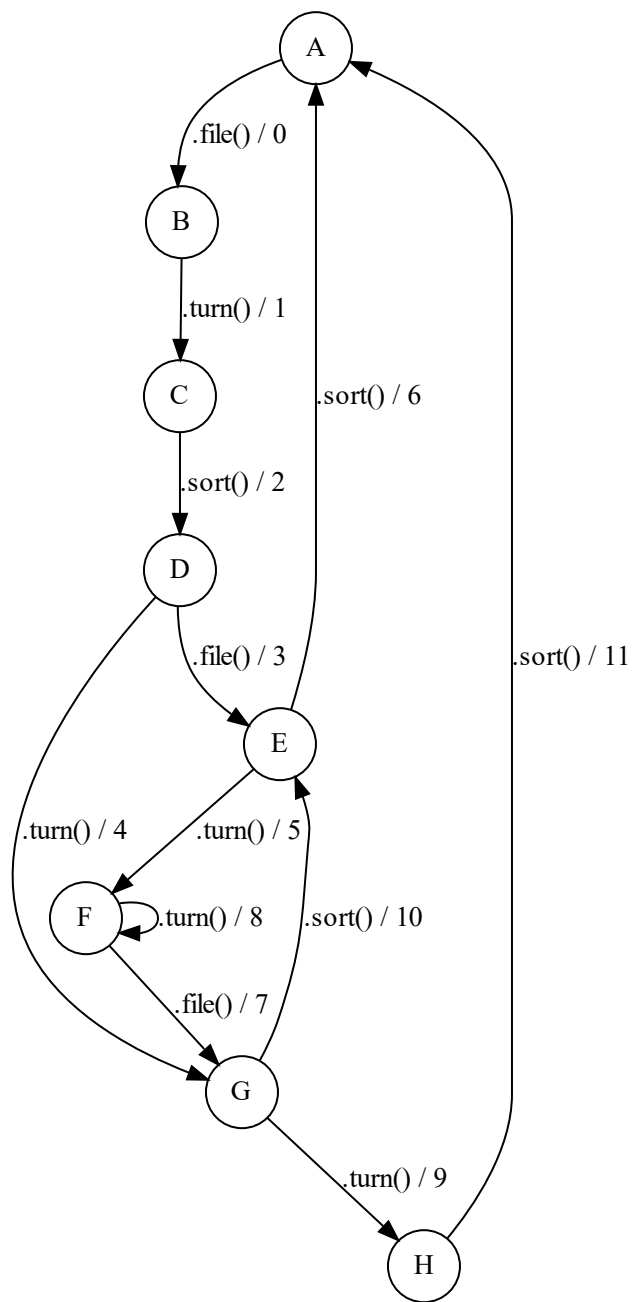
```
(b'QMLF\r\xef\xf7\r\x1c\x11%7\xea\x8e\xa8Pej/?\xe4z\xfa9 \xbf\x80R'  
b'\xbb\x83\xd\xa1\xf3\x90\xd9e\xf1\xec&\xc5\x12\xcf\xd5\xb1z\xbd\x87\xc2'  
b'\xeb\xf3o\x9b\xf6BB\x18\xaf\xb5\xdf\x19\xd6\x9cbq\\\x12Y\x87*B'\xeb\xfc'  
b'!j\x98\x10\xa6l\xb9\xca\xd0\x19z\x00tq\xa9\xec&\xd7\xea\x81\x1f\xbaZb'  
b'\xc5\x1e\x1304\xae\xbf\xe4\xd4\x10\x03^\x03\x94\x12fm\xe5\xdd\xc5'
```



Результат разбора:

```
{'A1': -360417098,
 'A2': {'B1': 9365485272659245766,
        'B2': 223,
        'B3': 61994,
        'B4': 0.8069680080018398,
        'B5': {'C1': 7190135655074348603,
                 'C2': 8835,
                 'C3': [{'D1': 165,
                         'D2': 28222,
                         'D3': [1381414778,
                                -1125134391,
                                1913759794,
                                -1258360192,
                                -617749925]}],
                 {'D1': 47,
                  'D2': -30539,
                  'D3': [-1984695452,
                         1200544613,
                         901131489,
                         1069163838,
                         1711392380]}]},
        'C4': {'E1': -0.44838613143333395,
                 'E2': -20556,
                 'E3': [148, 139, 236, 159],
                 'E4': [145, 186, 95, 189, 163, 130, 6],
                 'E5': [30702, 25628],
                 'E6': 56834,
                 'E7': 10869641856025978570},
        'C5': 331600367101513715,
        'C6': 105,
        'C7': -1663094169491744761},
 'B6': 0.08514687020278533}}
```

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



1. Пример использования класса C32:



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

## 2. Пример использования класса C32:

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

## Вариант №40

**Задача 3.1.** Реализовать разбор двоичного формата данных (в духе формата WAD игры Doom или графического формата PNG). Данные начинаются с сигнатуры 0x76 0x47 0x50 0x59 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	uint64
	2	Адрес (uint16) структуры B
	3	int8

Структура B:	1	Адрес (uint32) структуры C
	2	Размер (uint16) и адрес (uint16) массива int8
	3	int32
	4	uint32
	5	double
	6	float
	7	uint8

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

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

Примеры разбора двоичного формата с помощью функции f31:

### 1. Двоичные данные:

```
(b'vGPYW\xce\xc6\\\xa1\xcb\xee\xc8Xs\x00\xfd\xa7\x95\xb4K+\x9fL\x81'  
b'[\x96\xb9\x03\x00\x10\x00\xe0\xcd\xb2R\x17\x97;\xdf\x10|\x04\xb3\xbd'  
b'n\xd8\x90\x7f\x97\x03\x00&\x00\xfa9\xc2:\xe8\x82\x04\x88\xa5\xf8Sh\xc4\xa86'  
b'_\x92\x02\x00<\x00a\\\x0f\xc2\x14\xcf\xa2{\x1dg\xb3\x1e\xa1r\xaa\x9c\x19\xbe'  
b'\x03\x00Q\x00\xa7g\xbc\x9f\xd8\x7f\x13\x00)\x00>\x00T\x00\xc2-\x1a\x0fg'  
b'\x00\x00\x00\x02\x00q\x00\xcdf.8\xf75\x14\x85\xf8\x80\x11\xa6\x88J\xcf?@AG>N')
```

Результат разбора:

```
{'A1': 6397625829032773326,  
'A2': {'B1': {'C1': [{'D1': 13372976754417675083,  
                 'D2': [167, 149, 180],  
                 'D3': [-32, -51, -78, 82, 23, -105],  
                 'D4': 59},  
         'D1': 10916603281014633220,
```

```

'D2': [223, 16, 124],
'D3': [-6, 57, -62, 58, -24, -126],
'D4': 4},
{'D1': 10547208950802568184,
'D2': [136, 165],
'D3': [97, 92, 15, -62, 20, -49],
'D4': -94},
{'D1': 13698151997505609395,
'D2': [123, 29, 103],
'D3': [-89, 103, -68, 105, -97, -40],
'D4': 127}},
'C2': 11714},
'B2': [26, 15],
'B3': 942565069,
'B4': 2232701943,
'B5': 0.2444620905762418,
'B6': 0.19458484649658203,
'B7': 78},
'A3': -3}

```

## 2. Двоичные данные:

```

(b'vGPYW\x0cn\xcfs\x8f\t\xf6t\x00f\xeb]\x93\xcf\xceB\xd7X\xca\xc2i\x03'
b'\x00\x10\x00\x1d\xf4\x89\rG\xe1\x8a\xca\x01\x07\x14tW;\xda:\x92p\x03\x00&'
b'\x00\xfc\xd5\x9c"|\xbe\x1c\xa9\x9a\xf4\xcc\xb4\x05\x04\xf9\x99_E\x03'
b'\x00<\x00Mc\xc72\xf3\xdb\x1e\xdb\xe4\xb7\xfb%\x8c\xb2\xe4\xf5\x05'
b'\xc8\x03\x00R\x00W-W\xd9\xb1|?\x13\x00)\x00?\x00U\x00\n\x8d\x89\x1c'
b'h\x00\x00\x00\x02\x00r\x00\x7f\xc5\xde\xd5dY\x06wp\x0f\xe3\x0e9\xf1\xb5\xbf'
b'\xe9V\xd4>%' )

```

## Результат разбора:

```

{'A1': 17728859417615494668,
'A2': {'B1': {'C1': [{'D1': 7620876002381713103,
'D2': [235, 93, 147],
'D3': [29, -12, -119, 13, 71, -31],
'D4': -118},
{'D1': 8111610587820553236,
'D2': [202, 1, 7],
'D3': [-4, -43, -100, 34, 124, -66],
'D4': 28},
{'D1': 4998883406197929164,
'D2': [169, 154, 244],
'D3': [77, 99, -57, 50, -13, -37],
'D4': 30},
{'D1': 14413196545066018299,

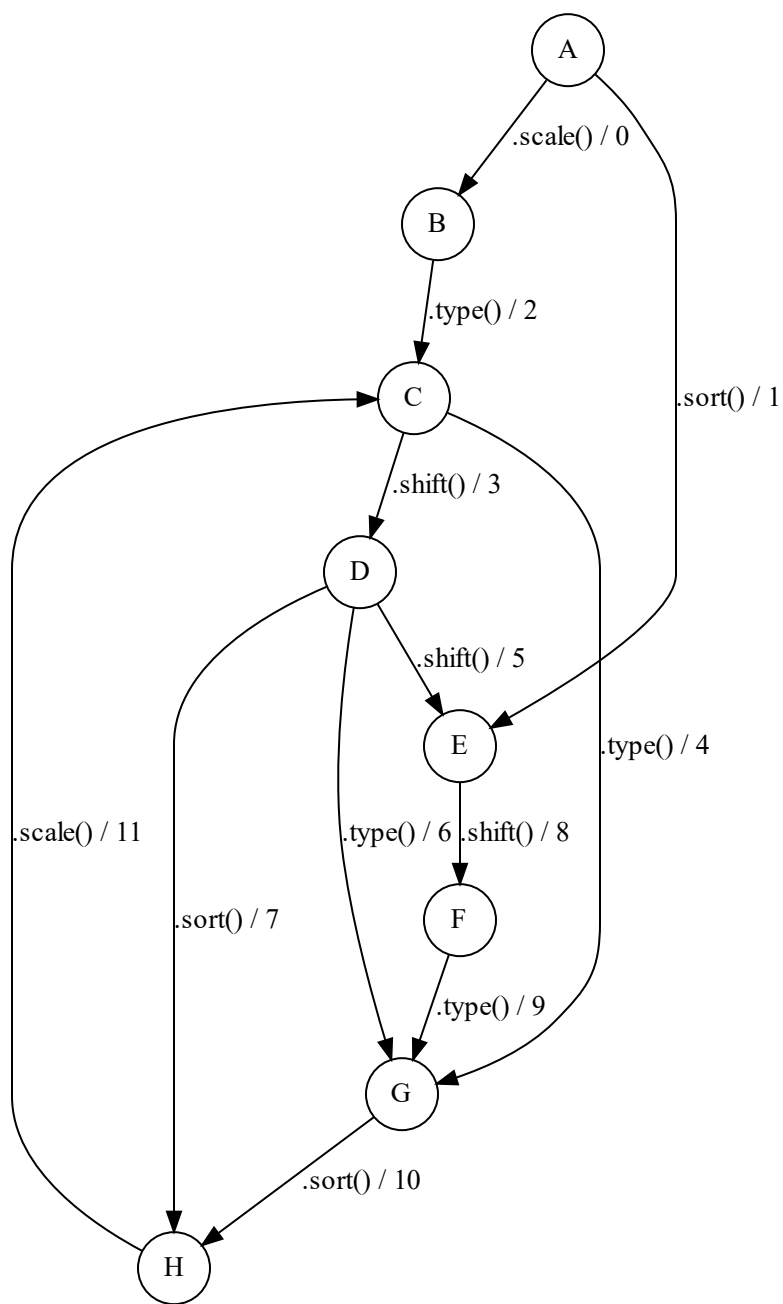
```

```

        'D2': [219, 228, 183],
        'D3': [87, 45, 87, -39, -79, 124],
        'D4': 63}],
    'C2': 36106},
    'B2': [-119, 28],
    'B3': -706820737,
    'B4': 1996904804,
    'B5': -0.08571201909495385,
    'B6': 0.41472557187080383,
    'B7': 37},
    'A3': 102}

```

**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.scale()	0
o.type()	2
o.shift()	3
o.shift()	5
o.sort()	RuntimeError
o.shift()	8
o.type()	9
o.sort()	10
o.scale()	11
o.shift()	3
o.sort()	7
o.scale()	11
o.type()	4
o.sort()	10
o.scale()	11
o.shift()	3
o.type()	6

## 2. Пример использования класса C32:

o = C32()	
o.scale()	0
o.type()	2
o.shift()	3
o.type()	6
o.sort()	10
o.scale()	11
o.shift()	3
o.sort()	7
o.scale()	11
o.type()	4
o.sort()	10
o.scale()	11
o.shift()	3
o.scale()	RuntimeError
o.shift()	5
o.scale()	RuntimeError
o.shift()	8
o.type()	9