Вход: файл input.json

Формат:

1) $char(F) != 2,3$

$module$ совпадает с характеристикой поля,

a, b – параметры

Для умножения необходимо указать $factor, x, y$

$factor$ – число BigInt

Для сложения – $x1, y1, x2, y2$

Значения можно задавать в виде десятичных чисел:

```
{
  "module": "97",
  "a": "2",
  "b": "3",
  "factor": "432121321321321312321321312324342342342341321312313213213213213131",
  "x": "3",
  "y": "6"
}
```

Либо в виде шестнадцатеричных (с префиксом **0x**):

```
{
  "module": "0x61",
  "a": "0x2",
  "b": "0x3",
  "x1": "0x3",
  "y1": "0x6",
  "x2": "0x3",
  "y2": "0x6"
}
```

Так же возможно указание бесконечно удаленной точки (x == null, y == null):

```
{
  "module": "0x97",
  "a": "0x2",
  "b": "0x3",
  "x1": "0x3",
  "y1": "0x6",
  "x2": null,
  "y2": null
}
```

2) $\boldsymbol{char(F) == 2}$

$\boldsymbol{module, a, b, x, y, x1, y1, x2, y2}$ указываются в виде массива показателей степеней ненулевых одночленов

Например: $x^4 + x + 1$ будет выглядеть как [4, 1, 0]

$\boldsymbol{module}$ – неприводимый многочлен над F_2

$\boldsymbol{a, b, x, y, x1, y1, x2, y2} \in \boldsymbol{GF(2^n)}$.

$\boldsymbol{factor}$ – число BigInt

Пример сложения:

```
{
  "module": [ 4, 1, 0 ],
  "a": [ 0 ],
  "b": [ 2, 1, 0 ],
  "x1": [ 1, 0 ],
  "y1": [ 1 ],
  "x2": [ 1 ],
  "y2": [ 2, 0 ]
}
```

Пример умножения:

```
{
  "module": [ 4, 1, 0 ],
  "a": [ 0 ],
  "b": [ 2, 1, 0 ],
  "factor": "432121321321321312321321312324342342342341321312313213213213213131",
  "x": [ 1, 0 ],
  "y": [ 2, 0 ]
}
```

Выход: файл output.txt

Для первого случая выводятся 10 и 16-ичные записи, для второго – пара точек из $GF(2^n)$

Запуск:

После изменения файла input.json, запустить run.cmd

Результат в файле output.txt

Для вышеописанных случаев в текущей директории есть примеры input-файлов (файлы вида input_*.json)