> 01 <

COMPRESSION AND INFORMATION

```
                    &@%(*/%@#*
            .#.,(,@,,,,,,,,,,/&*(.
          (,&,*,,@,,,,,*&,,,,,,,,*%&(
         //%,,,,,*@*,,,*%(*,,,,,,,,,*@/
      ..(@@/,,,,,,,,,,//,,,,,,,,,,,,,,*/&
&#, #*&#,,,,,,,,,,,,,,,,,,,,,(&&%,,,,,,@
&@**,,,,,,,,*,,,,,,,,,,,,,,,%/,,,,,,,*#,
 %&#*,,,,,((,**,%*,,,,,,,,,,#(*,,,,,,@,.
   &&&,,,,@,,,,,,,,,,,,,,,,,,/@@@&,#
      /&&&*@*,,,,,,,@#*,,,,,,/&/
           *,(&@@#*# #,*##( **
```

The original Akatsuki cloud represented justice and fairness. Later however it began to represent the rain of blood that fell in Amegakure during its wars.

This game is a game of information.

How can we represent information as compact as possible.

What is the smallest amount of symbols we need in order to still think of Akatsiku when we see the image?

> 02 <

GAME RULES

This game is more of a puzzle, you can play it alone, or with friends, but try to start with the easy and obvious cards.

Remember that it is actually very difficult to do what you are about to do, so do not get discouraged.

> 1. WATCH NARUTO IF YOU HAVENT

> 2. PICK AN IMAGE CARD

> 3. FIND THE ENCODING CARD
> MATCHING THE OUTPUT IMAGE

> 4. IF YOU ARE HAVING FUN GOTO 2

> 5. WATCH:

Hunter x Hunter
One Piece
Bleach
One Punch Man
Haikyuu
Hajime no Ippo
My Hero Academia
Sword Art Online

> 03 <

WHAT IS AN IMAGE

Images are just an array of pixels, a pixel is just a dot of color, your monitor probably has more than 2,073,600 pixels.

Each pixel has 3 one byte values:

red: from 0 to 255
green: from 0 to 255
blue: from 0 to 255

So the image is list of pixels, and each pixel has the 3 colors.For example, we could have one red and one magenta pixel next to each other:

[...,(255,0,0),(255,0,255),...]

In this game we will use text symbols to represent pixels, but the idea is the same. Each card has 40 columns and 31 rows, so it has 1240 pixels. This tree has 42 pixels:

```
.......
...*...
..***..
.*****.
...|...
.......
```

> 04 <

ENCODE AND DECODE

First we will transform our text image into something we can use, like a list of numbers. We will encode each symbol from the image with a number, and we will create a symbol table so we can use it to decode the image later.
For example, having this 6x7 image:

```
.......
...*...
..***..
.*****.
...|...
.......
```

The first symbol we see is ., we will give it the number 0, next is *, so this will be number 1, and then last we see |, which will be number 2

Our symbol table will look like this:

{'.': 0, '*': 1, '|': 2}

and our encoded image would look like:

```
0000000
0001000
0011100
0111110
0002000
0000000
```

> 05 <

ENCODE AND DECODE

```
def encode(x):
  sym = {}
  r = []

  for v in x:
    if v not in sym:
      # first time we see a symbol
      # we put it in the dictionary
      sym[v] = len(sym)

    # append the number of the symbol
    r.append(sym[v])

  # return both the list and the table
  return [r, sym]

def decode(x, sym):
  # invert keys and values from
  #   { ".": 1, "@": 2}
  # to
  #   {1: ".", 2: "@"}
  rs = {sym[k] : k for k in sym}
  r = ''

  for v in x:
    # lookup symbol from number
    r += rs[v]

  return r
```

> 06 <

RUNLENGTH ENCODING

Lets flatten this image to how it actually looks as a list of numbers:

```
0000000
0001000
0011100
0111110
0002000
0000000
```

becomes:

```
[
  0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,1,1,1,
  0,0,0,1,1,1,1,1,0,0,0,0,2,0,0,0,0,0,0,
  0,0,0,0
]
```

There is a lot of repetition in this list, so we can just write how many times each number appears in the sequence:

```
[
  10,0,1,1,5,0,3,1,3,0,5,1,4,0,1,2,10,0
]
```

meaning, 10 zeroes, 1 time one, and again 5 zeroes, 3 ones etc..

This is called RunLength Encoding. To decode simply do the reverse operation.

> 07 <

RUNLENGTH ENCODING

```
# run length encode a list of numbers
# from:
#   [1,1,1,1,1,1,1,2]
# to:
#   [7,1,1,2]

def rle(x):
  r = []
  for v in x:
    if len(r) == 0 or r[-1] != v:
      r.append(0)
      r.append(v)
    if v == r[-1]:
      r[-2] += 1
  return r

# run length decode
# from:
#   [7,1,1,2]
# to:
#   [1,1,1,1,1,1,1,2]
def rld(x):
  r = []

  for i in range(0, len(x), 2):
    for k in range(x[i]):
      r.append(x[i+1])

  return r
```

> 08 <

DELTA ENCODING

Lets look at a list of numbers:

[1, 3, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]

If we runlength encode this list it will blow up, most elements appears only once

```
[
  1,1,3,2,4,1,5,1,6,1,7,1,8,1,9,1,
  10,1,11,1,12,1
]
```

However we can encode the difference between each element, so

```
[
  1, # first element is 1
  2, # 3 - 1
  0, # 3 - 3
  1, # 4 - 3
  ...
]
```

This is called delta encoding, we encode the the difference (the delta).

So the list becomes very repetitive:
[1, 2, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1]

Now we can runlength encode it:
[1, 1, 1, 2, 1, 0, 9, 1]
To go back to the original list first we need to run length decode and then to delta decode.

DELTA ENCODING

```
# delta encode a list of numbers
# from:
#   [1,2,3,4,5]
# to:
#   [1,1,1,1,1]
def delta(x):
  r = [x[0]]

  for i in range(1, len(x)):
    current = x[i]
    prev = x[i-1]
    d = current - prev
    r.append(d)

  return r


# delta decode a list of numbers
# from:
#   [1,1,1,1,1]
# to:
#   [1,2,3,4,5]
def undelta(x):
  r = [x[0]]

  for i in range(1, len(x)):
    r.append(x[i] + r[-1])

  return r
```


REDUCE INFORMATION

Going back to our tree:

```
                .......
                ...*...
                ..***..
                .*****.
                ...|...
                .......
```

How would it look if we try to squeeze it in fewer pixels? For example take every 2 pixels and average them together and then explode it back:

```
 original     squeeze     exploded
0000000    0 0 0 0    0000000
0001000    0 0 0 0    0000000
0011100 -> 0 1 0 0 -> 0011000
0111110    0 1 1 0    0011110
0002000    0 1 0 0    0011000
0000000    0 0 0 0    0000000
```

Then if we draw it again:

```
                .......
                .......
                ..**...
                ..****.
                ..**...
                .......
```

Its ugly, but it .. kind of looks like

REDUCE INFORMATION

```
# average every n elements
# from:
#  [1,2,4,4,9,5] with n=2
# to:
#  [1,4,7]
def squeeze(x, n):
  r = []

  for i in range(0, len(x), n):
    avg = sum(x[i:i+n])/n
    r.append(int(avg))

  return r


# explode the elements
# from:
#   [1,4,7] with n=2
# to:
#   [1,1,4,4,7,7]
def unsqueeze(x, n):
  r = []

  for v in x:
    for i in range(n):
      r.append(v)

  return r
```


FILTERS

You can do all kinds of manpulations of the image data.

An example is Black And White filter:

```
  for every pixel
     if the pixel is not zero
       set the pixel to WHITE
     else
       set the pixel to BLACK
```

Simple Blur filter:

```
  for every block of 8x8 pixels
    replace them with their
    average
```

Invert filter:

```
  for each pixel:
    set it to the opposite
    e.g. ORANGE <-> BLUE
         RED <-> GREEN
         WHITE <-> BLACK
```

In our example we use simplified versions of those filters, but the fundamental idea is the same.

Take the pixels and manipulate them.

FILTERS

```
def blur(x):
  # fake blur, averaging every 3 element
  s = squeeze(x, 3)
  return unsqueeze(s,3)

def invert(x):
  # invert the values
  # [1,1,3,0,0] -> [2,2,0,3,3]
  r = []

  m = max(x)
  for v in x:
    r.append(m - v)

  return r

def bw(x):
  # make everything "black and white"
  # [2,7,0,0] -> [1,1,0,0]
  r = []

  for v in x:
    if v == 0:
      r.append(0)
    else:
      r.append(1)

  return r
```



```
........................................
........................................
........................................
........................................
........................................
.............@@++++++++++@@.............
..........@++++++++++++++++@............
.........@+++++++++++++++++++@..........
........(+++++++++++*++++++++@..........
.....@@++++++++++++@++++++++@++&@.......
...@/+++++++++++++++++@@@@%+++++++@.....
...@+++++++++++++++++++++++++++++++@....
...@+++++++++++@+++++++++++++++++++@....
...@+++++++++++@++++++++++++++++++@.....
.....@++++++++@@++++++++++++++++@.......
.......@@@@@@%++++++++++++++++++++@.....
.......@++++++++++++++++++@&+++++++@....
........@+++++++++++++++@++++++++++@....
..........@@++++++++++++@+++++++++@.....
................%+++++++++@/++@@@.......
.................@+++++++++++++(........
...................@++++++++++@.........
.....................@+++++++@..........
.....................@++++@@............
....................@@/.................
........................................
........................................
........................................
........................................
........................................
........................................
```



```
0000000000000000000000000000000000000000
0000000000000000000000000000000000000000
0000000000000000000000000000000000000000
0000000000000000000000000000000000000000
0000000000000000000000000000000000000000
0000000000000112222222222110000000000000
0000000000122222222222222221000000000000
0000000001222222222222222222210000000000
0000000032222222222242222222210000000000
0000011222222222222122222222122510000000
0001622222222222222222111172222222100000
0001222222222222222222222222222222210000
0001222222222221222222222222222222210000
0001222222222221222222222222222222100000
0000012222222211222222222222222210000000
0000000111111722222222222222222222100000
0000000122222222222222222215222222210000
0000000012222222222222221222222222210000
0000000000112222222222221222222222100000
0000000000000000722222222216221110000000
0000000000000000012222222222222300000000
0000000000000000000122222222221000000000
0000000000000000000001222222210000000000
0000000000000000000001222211000000000000
0000000000000000000011600000000000000000
0000000000000000000000000000000000000000
0000000000000000000000000000000000000000
0000000000000000000000000000000000000000
0000000000000000000000000000000000000000
0000000000000000000000000000000000000000
0000000000000000000000000000000000000000
```


SYMBOL TABLE

```
{
    ".": 0,
    "@": 1,
    "+": 2,
    "(": 3,
    "*": 4,
    "&": 5,
    "/": 6,
    "%": 7
}
```

> 17 <

rle(encoded)

size: 216

213	0	2	1	9	2	2	1	24	0
1	1	16	2	1	1	21	0	1	1
19	2	1	1	18	0	1	3	11	2
1	4	8	2	1	1	15	0	2	1
12	2	1	1	8	2	1	1	2	2
1	5	1	1	10	0	1	1	1	6
17	2	4	1	1	7	7	2	1	1
8	0	1	1	31	2	1	1	7	0
1	1	11	2	1	1	19	2	1	1
7	0	1	1	11	2	1	1	18	2
1	1	10	0	1	1	8	2	2	1
16	2	1	1	14	0	6	1	1	7
20	2	1	1	12	0	1	1	18	2
1	1	1	5	7	2	1	1	12	0
1	1	15	2	1	1	10	2	1	1
14	0	2	1	12	2	1	1	9	2
1	1	21	0	1	7	9	2	1	1
1	6	2	2	3	1	24	0	1	1
13	2	1	3	27	0	1	1	10	2
1	1	30	0	1	1	7	2	1	1
31	0	1	1	4	2	2	1	32	0
2	1	1	6	257	0				

> 18 <

```
0000000000000000000000000000000000000000
0000000000000000000000000000000000000000
0000000000000000000000000000000000000000
0000000000000000000000000000000000000000
0000000000000000000000000000000000000000
0000000000000111222222111000000000000000
0000000001112222222222222220000000000000
0000000011122222222222222222200000000000
0000000111222222222222222222111000000000
0000001112222222221112222221112220000000
0022222222222222222211111133322211100000
0000222222222222222222222222222222111000
0001112222222221112222222222222221110000
0011122222222211122222222222222211100000
0000111222222111222222222222222111000000
0000000001113332222222222222222221110000
0000000022222222222222222222222222200000
0000000111222222222222111222222222111000
0000000000002222222222221112222221110000
0000000000000022222222222233311100000000
0000000000000000111222222222222111000000
0000000000000000001112222222220000000000
0000000000000000000011122222200000000000
0000000000000000000000222111000000000000
0000000000000000000002220000000000000000
0000000000000000000000000000000000000000
0000000000000000000000000000000000000000
0000000000000000000000000000000000000000
0000000000000000000000000000000000000000
0000000000000000000000000000000000000000
0000000000000000000000000000000000000000
```

> 19 <

```
........................................
........................................
........................................
........................................
........................................
.............@@@++++++@@@...............
.........@@@+++++++++++++++.............
........@@@++++++++++++++++++...........
.......@@@++++++++++++++++++@@@.........
......@@@+++++++++@@@++++++@@@+++.......
..++++++++++++++++++@@@@@@(((+++@@@.....
....++++++++++++++++++++++++++++++@@@...
...@@@+++++++++@@@+++++++++++++++@@@....
..@@@+++++++++@@@+++++++++++++++@@@.....
....@@@++++++@@@+++++++++++++++@@@......
.........@@@(((++++++++++++++++++@@@....
........+++++++++++++++++++++++++++.....
.......@@@++++++++++++@@@+++++++++@@@...
............++++++++++++@@@++++++@@@....
..............++++++++++++(((@@@........
................@@@++++++++++++@@@......
..................@@@+++++++++..........
....................@@@++++++...........
......................+++@@@............
.....................+++................
........................................
........................................
........................................
........................................
........................................
........................................
```

> 20 <

rle(blur(encoded))

size: 172

213	0	3	1	6	2	3	1	24	0
3	1	15	2	21	0	3	1	18	2
18	0	3	1	18	2	3	1	15	0
3	1	9	2	3	1	6	2	3	1
3	2	9	0	18	2	6	1	3	3
3	2	3	1	9	0	30	2	3	1
6	0	3	1	9	2	3	1	15	2
3	1	6	0	3	1	9	2	3	1
15	2	3	1	9	0	3	1	6	2
3	1	15	2	3	1	15	0	3	1
3	3	18	2	3	1	12	0	27	2
12	0	3	1	12	2	3	1	9	2
3	1	15	0	12	2	3	1	6	2
3	1	18	0	12	2	3	3	3	1
24	0	3	1	12	2	3	1	24	0
3	1	9	2	30	0	3	1	6	2
33	0	3	2	3	1	33	0	3	2
258	0								

> 21 <

squeeze(encoded,5)

size: 248

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	2	1	0	0	0	0	0
1	2	2	1	0	0	0	0	2	2
2	1	0	0	0	1	2	2	2	1
0	0	0	1	2	1	2	1	1	0
1	2	2	2	1	2	1	0	0	2
2	2	2	2	2	0	0	2	2	1
2	2	2	0	0	2	2	1	2	2
1	0	0	1	1	1	2	2	1	0
0	0	2	2	2	2	1	0	0	1
2	2	2	2	2	0	0	0	2	2
1	2	2	0	0	0	1	2	1	2
1	0	0	0	0	2	2	2	0	0
0	0	0	1	2	2	1	0	0	0
0	0	2	2	0	0	0	0	0	0
1	1	0	0	0	0	0	0	1	0
0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0		

> 22 <

rle(squeeze(encoded,5))

size: 144

43	0	1	2	1	1	5	0	1	1
2	2	1	1	4	0	3	2	1	1
3	0	1	1	3	2	1	1	3	0
1	1	1	2	1	1	1	2	2	1
1	0	1	1	3	2	1	1	1	2
1	1	2	0	6	2	2	0	2	2
1	1	3	2	2	0	2	2	1	1
2	2	1	1	2	0	3	1	2	2
1	1	3	0	4	2	1	1	2	0
1	1	5	2	3	0	2	2	1	1
2	2	3	0	1	1	1	2	1	1
1	2	1	1	4	0	3	2	5	0
1	1	2	2	1	1	5	0	2	2
6	0	2	1	6	0	1	1	7	0
1	1	51	0						

> 23 <

rle(bw(squeeze(encoded,5)))

size: 82

43	0	2	1	5	0	4	1	4	0
4	1	3	0	5	1	3	0	6	1
1	0	7	1	2	0	6	1	2	0
6	1	2	0	6	1	2	0	6	1
3	0	5	1	2	0	6	1	3	0
5	1	3	0	5	1	4	0	3	1
5	0	4	1	5	0	2	1	6	0
2	1	6	0	1	1	7	0	1	1
51	0								

> 24 <

```
........................................
........................................
........................................
........................................
........................................
...............+++++@@@@@...............
..........@@@@@++++++++++@@@@@..........
..........+++++++++++++++@@@@@..........
.....@@@@@+++++++++++++++@@@@@..........
.....@@@@@+++++@@@@@+++++@@@@@@@@@@.....
@@@@@+++++++++++++++@@@@@+++++@@@@@.....
.....++++++++++++++++++++++++++++++.....
.....++++++++++@@@@@+++++++++++++++.....
.....++++++++++@@@@@++++++++++@@@@@.....
.....@@@@@@@@@@@@@@@++++++++++@@@@@.....
..........++++++++++++++++++++@@@@@.....
.....@@@@@+++++++++++++++++++++++++.....
..........++++++++++@@@@@++++++++++.....
..........@@@@@+++++@@@@@+++++@@@@@.....
...............+++++++++++++++..........
...............@@@@@++++++++++@@@@@.....
....................++++++++++..........
....................@@@@@@@@@@..........
....................@@@@@...............
....................@@@@@...............
........................................
........................................
........................................
........................................
........................................
........................................
```

-> 25 <-

squeeze(encoded,20)

size: 62

```
0  0  0  0  0  0  0  0  0  0
0  0  0  0  1  0  1  1  1  1
1  1  1  1  1  1  1  1  1  1
1  1  1  1  1  1  0  1  0  1
0  1  0  1  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0
0  0
```

-> 26 <-

rle(squeeze(encoded,20))

size: 26

```
14  0  1  1  1  0 20  1  1  0
 1  1  1  0  1  1  1  0  1  1
 1  0  1  1 18  0
```

-> 27 <-

rle(bw(squeeze(encoded,20)))

size: 26

```
14  0  1  1  1  0 20  1  1  0
 1  1  1  0  1  1  1  0  1  1
 1  0  1  1 18  0
```

-> 28 <-

```
........................................
........................................
........................................
........................................
........................................
........................................
........................................
@@@@@@@@@@@@@@@@@@@@....................
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
....................@@@@@@@@@@@@@@@@@@@@
....................@@@@@@@@@@@@@@@@@@@@
....................@@@@@@@@@@@@@@@@@@@@
....................@@@@@@@@@@@@@@@@@@@@
........................................
........................................
........................................
........................................
........................................
........................................
........................................
........................................
........................................
```

-> 29 <-

squeeze(encoded,40)

size: 31

```
0  0  0  0  0  0  0  1  1  1
1  1  1  1  1  1  1  1  1  0
0  0  0  0  0  0  0  0  0  0
0
```

-> 30 <-

rle(squeeze(encoded,40))

size: 6

```
 7  0 12  1 12  0
```

-> 31 <-

rle(bw(squeeze(encoded,40)))

size: 6

```
 7  0 12  1 12  0
```

-> 32 <-

```
........................................
........................................
........................................
........................................
........................................
........................................
........................................
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
........................................
........................................
........................................
........................................
........................................
........................................
........................................
........................................
........................................
........................................
........................................
........................................
```

> 33 <

squeeze(blur(encoded),5)

size: 249

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	1	1	0	0	0	0	0
1	2	2	0	0	0	0	0	1	2
2	1	0	0	0	0	2	2	2	1
0	0	0	1	2	1	1	1	1	0
1	2	2	2	1	2	1	0	0	2
2	2	2	2	1	0	0	1	2	1
2	2	1	0	0	2	1	1	2	2
1	0	0	1	1	1	2	2	1	0
0	0	2	2	2	2	1	0	0	0
2	2	2	2	2	0	0	0	2	2
1	2	1	0	0	0	1	2	1	1
1	0	0	0	0	2	2	2	0	0
0	0	0	1	2	2	1	0	0	0
0	0	1	2	0	0	0	0	0	0
1	1	0	0	0	0	0	0	1	0
0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	

> 34 <

rle(squeeze(blur(encoded),5))

size: 140

43	0	2	1	5	0	1	1	2	2
5	0	1	1	2	2	1	1	4	0
3	2	1	1	3	0	1	1	1	2
4	1	1	0	1	1	3	2	1	1
1	2	1	1	2	0	5	2	1	1
2	0	1	1	1	2	1	1	2	2
1	1	2	0	1	2	2	1	2	2
1	1	2	0	3	1	2	2	1	1
3	0	4	2	1	1	3	0	5	2
3	0	2	2	1	1	1	2	1	1
3	0	1	1	1	2	3	1	4	0
3	2	5	0	1	1	2	2	1	1
5	0	1	1	1	2	6	0	2	1
6	0	1	1	7	0	1	1	52	0

> 35 <

rle(delta(squeeze(blur(encoded)),5))

size: 180

43	0	1	1	1	0	1	-1	4	0
2	1	1	0	1	-2	4	0	2	1
1	0	2	-1	3	0	1	2	2	0
2	-1	2	0	2	1	1	-1	3	0
1	-1	2	1	2	0	1	-1	1	1
2	-1	1	0	1	2	4	0	2	-1
1	0	2	1	1	-1	1	1	1	0
2	-1	1	0	1	2	1	-1	1	0
1	1	1	0	2	-1	1	0	1	1
2	0	1	1	1	0	2	-1	2	0
1	2	3	0	2	-1	2	0	1	2
4	0	1	-2	2	0	1	2	1	0
1	-1	1	1	2	-1	2	0	2	1
1	-1	2	0	1	-1	3	0	1	2
2	0	1	-2	4	0	2	1	1	0
2	-1	4	0	2	1	1	-2	5	0
1	1	1	0	1	-1	5	0	1	1
1	-1	6	0	1	1	1	-1	51	0

> 36 <

```
........................................
........................................
........................................
........................................
........................................
...............@@@@@@@@@@...............
..........@@@@@++++++++++...............
..........@@@@@++++++++++@@@@@..........
..........+++++++++++++++@@@@@..........
.....@@@@@+++++@@@@@@@@@@@@@@@@@@@@.....
@@@@@+++++++++++++++@@@@@+++++@@@@@.....
.....+++++++++++++++++++++++++@@@@@.....
.....@@@@@+++++@@@@@++++++++++@@@@@.....
.....+++++@@@@@@@@@@++++++++++@@@@@.....
.....@@@@@@@@@@@@@@@++++++++++@@@@@.....
..........++++++++++++++++++++@@@@@.....
..........+++++++++++++++++++++++++.....
..........++++++++++@@@@@+++++@@@@@.....
..........@@@@@+++++@@@@@@@@@@@@@@@.....
...............+++++++++++++++..........
...............@@@@@++++++++++@@@@@.....
....................@@@@@+++++..........
....................@@@@@@@@@@..........
....................@@@@@...............
....................@@@@@...............
........................................
........................................
........................................
........................................
........................................
........................................
```

> 37 <

squeeze(blur(encoded),20)

size: 63

0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	1	0	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	0	1	0	1
0	1	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0							

> 38 <

rle(squeeze(blur(encoded),20))

size: 26

14	0	1	1	1	0	1	1	1	0
18	1	1	0	1	1	1	0	1	1
1	0	1	1	21	0				

> 39 <

rle(delta(squeeze(blur(encoded)),20))

size: 30

14	0	1	1	1	-1	1	1	1	-1
1	1	17	0	1	-1	1	1	1	-1
1	1	1	-1	1	1	1	-1	20	0

> 40 <

```
........................................
........................................
........................................
........................................
........................................
........................................
........................................
@@@@@@@@@@@@@@@@@@@@....................
@@@@@@@@@@@@@@@@@@@@....................
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
....................@@@@@@@@@@@@@@@@@@@@
....................@@@@@@@@@@@@@@@@@@@@
....................@@@@@@@@@@@@@@@@@@@@
........................................
........................................
........................................
........................................
........................................
........................................
........................................
........................................
........................................
........................................
```

> 41 <

squeeze(blur(encoded),40)

size: 32

```
0   0   0   0   0   0   0   0   1   1
1   1   1   1   1   1   1   1   1   0
0   0   0   0   0   0   0   0   0   0
0   0
```

> 42 <

rle(squeeze(blur(encoded),40))

size: 6

```
8   0  11   1  13   0
```

> 43 <

rle(delta(squeeze(blur(encoded)),40))

size: 10

```
8   0   1   1  10   0   1  -1  12   0
```

> 44 <

```
........................................
........................................
........................................
........................................
........................................
........................................
........................................
........................................
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
........................................
........................................
........................................
........................................
........................................
........................................
........................................
........................................
........................................
........................................
........................................
........................................
```

> 45 <

```
0000000000000000000000000000000000000000
0000000000000000000000000000000000000000
0000000000000000000000000000000000000000
0000000000000000000000000000000000000000
0000000000000000000000000000000000000000
0000000000000111111111111100000000000000
0000000000111111111111111111000000000000
0000000001111111111111111111110000000000
0000000011111111111111111111110000000000
0000011111111111111111111111111110000000
0001111111111111111111111111111111100000
0001111111111111111111111111111111110000
0001111111111111111111111111111111110000
0001111111111111111111111111111111100000
0000011111111111111111111111111110000000
0000000111111111111111111111111111100000
0000000111111111111111111111111111110000
0000000011111111111111111111111111110000
0000000000111111111111111111111111100000
0000000000000000111111111111111110000000
0000000000000000011111111111111100000000
0000000000000000000111111111111000000000
0000000000000000000001111111110000000000
0000000000000000000001111111000000000000
0000000000000000000011100000000000000000
0000000000000000000000000000000000000000
0000000000000000000000000000000000000000
0000000000000000000000000000000000000000
0000000000000000000000000000000000000000
0000000000000000000000000000000000000000
0000000000000000000000000000000000000000
```

> 46 <

```
........................................
........................................
........................................
........................................
........................................
.............@@@@@@@@@@@@@..............
..........@@@@@@@@@@@@@@@@@@............
.........@@@@@@@@@@@@@@@@@@@@@..........
........@@@@@@@@@@@@@@@@@@@@@@..........
.....@@@@@@@@@@@@@@@@@@@@@@@@@@@@.......
...@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@.....
...@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@....
...@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@....
...@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@.....
.....@@@@@@@@@@@@@@@@@@@@@@@@@@@@.......
.......@@@@@@@@@@@@@@@@@@@@@@@@@@@@.....
.......@@@@@@@@@@@@@@@@@@@@@@@@@@@@@....
........@@@@@@@@@@@@@@@@@@@@@@@@@@@@....
..........@@@@@@@@@@@@@@@@@@@@@@@@@.....
................@@@@@@@@@@@@@@@@@.......
.................@@@@@@@@@@@@@@@........
...................@@@@@@@@@@@@.........
.....................@@@@@@@@@..........
.....................@@@@@@@............
....................@@@.................
........................................
........................................
........................................
........................................
........................................
........................................
```

> 47 <

rle(bw(encoded))

size: 82

```
213   0  13   1  24   0  18   1  21   0
 21   1  18   0  22   1  15   0  28   1
 10   0  32   1   8   0  33   1   7   0
 33   1   7   0  32   1  10   0  28   1
 14   0  28   1  12   0  29   1  12   0
 28   1  14   0  25   1  21   0  17   1
 24   0  15   1  27   0  12   1  30   0
  9   1  31   0   7   1  32   0   3   1
257   0
```

> 48 <

```
1111111111111111111111111111111111111111
1111111111111111111111111111111111111111
1111111111111111111111111111111111111111
1111111111111111111111111111111111111111
1111111111111111111111111111111111111111
1111111111111000000000000011111111111111
1111111111000000000000000000111111111111
1111111110000000000000000000001111111111
1111111100000000000000000000001111111111
1111100000000000000000000000000001111111
1110000000000000000000000000000000011111
1110000000000000000000000000000000001111
1110000000000000000000000000000000001111
1110000000000000000000000000000000011111
1111100000000000000000000000000001111111
1111111000000000000000000000000000011111
1111111000000000000000000000000000001111
1111111100000000000000000000000000001111
1111111111000000000000000000000000011111
1111111111111111000000000000000001111111
1111111111111111100000000000000011111111
1111111111111111111000000000000111111111
1111111111111111111110000000001111111111
1111111111111111111110000000111111111111
1111111111111111111100011111111111111111
1111111111111111111111111111111111111111
1111111111111111111111111111111111111111
1111111111111111111111111111111111111111
1111111111111111111111111111111111111111
1111111111111111111111111111111111111111
1111111111111111111111111111111111111111
```

```
> 49 <
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@.............@@@@@@@@@@@@@@
@@@@@@@@@@..................@@@@@@@@@@@@
@@@@@@@@@.....................@@@@@@@@@@
@@@@@@@@......................@@@@@@@@@@
@@@@@............................@@@@@@@
@@@................................@@@@@
@@@.................................@@@@
@@@.................................@@@@
@@@................................@@@@@
@@@@@............................@@@@@@@
@@@@@@@............................@@@@@
@@@@@@@.............................@@@@
@@@@@@@@............................@@@@
@@@@@@@@@@.........................@@@@@
@@@@@@@@@@@@@@@@.................@@@@@@@
@@@@@@@@@@@@@@@@@...............@@@@@@@@
@@@@@@@@@@@@@@@@@@@............@@@@@@@@@
@@@@@@@@@@@@@@@@@@@@@.........@@@@@@@@@@
@@@@@@@@@@@@@@@@@@@@@.......@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@@@@...@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
```

```
> 50 <
         rle(invert(bw(encoded)))

                 size: 82

213   1  13   0  24   1  18   0  21   1
 21   0  18   1  22   0  15   1  28   0
 10   1  32   0   8   1  33   0   7   1
 33   0   7   1  32   0  10   1  28   0
 14   1  28   0  12   1  29   0  12   1
 28   0  14   1  25   0  21   1  17   0
 24   1  15   0  27   1  12   0  30   1
  9   0  31   1   7   0  32   1   3   0
257   1
```

```
> 51 <
0000000000000000000000000000000000000000
0000000000000000000000000000000000000000
0000000000000000000000000000000000000000
0000000000000000000000000000000000000000
0000000000000000000000000000000000000000
0000000000000111111111111100000000000000
0000000000111111111111111111000000000000
0000000001111111111111111111110000000000
0000000011111111111111111111110000000000
0000011111111111111111111111111110000000
0001111111111111111111111111111111100000
0001111111111111111111111111111111110000
0001111111111111111111111111111111110000
0001111111111111111111111111111111100000
0000011111111111111111111111111110000000
0000000111111111111111111111111111100000
0000000111111111111111111111111111110000
0000000011111111111111111111111111110000
0000000000111111111111111111111111100000
0000000000000000111111111111111110000000
0000000000000000011111111111111100000000
0000000000000000000111111111111000000000
0000000000000000000001111111110000000000
0000000000000000000001111111000000000000
0000000000000000000011100000000000000000
0000000000000000000000000000000000000000
0000000000000000000000000000000000000000
0000000000000000000000000000000000000000
0000000000000000000000000000000000000000
0000000000000000000000000000000000000000
0000000000000000000000000000000000000000
```

```
> 52 <
........................................
........................................
........................................
........................................
........................................
.............@@@@@@@@@@@@...............
.........@@@@@@@@@@@@@@@@@@.............
........@@@@@@@@@@@@@@@@@@@@@...........
.......@@@@@@@@@@@@@@@@@@@@@@@@.........
......@@@@@@@@@@@@@@@@@@@@@@@@@@@.......
..@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@.....
....@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@...
...@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@....
..@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@.....
....@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@......
.........@@@@@@@@@@@@@@@@@@@@@@@@@@@....
........@@@@@@@@@@@@@@@@@@@@@@@@@@@.....
.......@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@...
............@@@@@@@@@@@@@@@@@@@@@@@@....
..............@@@@@@@@@@@@@@@@@@........
................@@@@@@@@@@@@@@@@@@......
..................@@@@@@@@@@@@..........
....................@@@@@@@@@...........
......................@@@@@@............
.....................@@@................
........................................
........................................
........................................
........................................
........................................
........................................
```

```
> 53 <
         rle(bw(blur(encoded)))

                 size: 82

213   0  12   1  24   0  18   1  21   0
 21   1  18   0  24   1  15   0  27   1
  9   0  33   1   9   0  33   1   6   0
 33   1   6   0  33   1   9   0  30   1
 15   0  27   1  12   0  27   1  12   0
 30   1  15   0  24   1  18   0  18   1
 24   0  18   1  24   0  12   1  30   0
  9   1  33   0   6   1  33   0   3   1
258   0
```

```
> 54 <
7777777777777777777777777777777777777777
7777777777777777777777777777777777777777
7777777777777777777777777777777777777777
7777777777777777777777777777777777777777
7777777777777777777777777777777777777777
7777777777777665555555556677777777777777
7777777777655555555555555556777777777777
7777777776555555555555555555567777777777
7777777745555555555553555555567777777777
7777766555555555555655555555655267777777
7776155555555555555555566605555555677777
7776555555555555555555555555555555567777
7776555555555556555555555555555555567777
7776555555555556555555555555555555677777
7777765555555566555555555555555567777777
7777777666666055555555555555555555677777
7777777655555555555555555562555555567777
7777777765555555555555556555555555567777
7777777777665555555555556555555555677777
7777777777777777055555555561556667777777
7777777777777777765555555555555477777777
7777777777777777777655555555556777777777
7777777777777777777776555555567777777777
7777777777777777777776555566777777777777
7777777777777777777766177777777777777777
7777777777777777777777777777777777777777
7777777777777777777777777777777777777777
7777777777777777777777777777777777777777
7777777777777777777777777777777777777777
7777777777777777777777777777777777777777
7777777777777777777777777777777777777777
```

```
> 55 <
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%//&&&&&&&&&//%%%%%%%%%%%%%%
%%%%%%%%%%/&&&&&&&&&&&&&&&&/%%%%%%%%%%%%
%%%%%%%%%/&&&&&&&&&&&&&&&&&&&/%%%%%%%%%%
%%%%%%%%*&&&&&&&&&&&(&&&&&&&&/%%%%%%%%%%
%%%%%//&&&&&&&&&&&&/&&&&&&&&/&&+/%%%%%%%
%%%/@&&&&&&&&&&&&&&&&&////.&&&&&&&/%%%%%
%%%/&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&/%%%%
%%%/&&&&&&&&&&&/&&&&&&&&&&&&&&&&&&&/%%%%
%%%/&&&&&&&&&&&/&&&&&&&&&&&&&&&&&&/%%%%%
%%%%%/&&&&&&&&//&&&&&&&&&&&&&&&&/%%%%%%%
%%%%%%%//////.&&&&&&&&&&&&&&&&&&&&/%%%%%
%%%%%%%/&&&&&&&&&&&&&&&&&&/+&&&&&&&/%%%%
%%%%%%%%/&&&&&&&&&&&&&&&/&&&&&&&&&&/%%%%
%%%%%%%%%%//&&&&&&&&&&&&/&&&&&&&&&/%%%%%
%%%%%%%%%%%%%%%%.&&&&&&&&&/@&&///%%%%%%%
%%%%%%%%%%%%%%%%%/&&&&&&&&&&&&&*%%%%%%%%
%%%%%%%%%%%%%%%%%%%/&&&&&&&&&&/%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%/&&&&&&&/%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%/&&&&//%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%//@%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```