This is CS50

```c
#include <stdio.h>

int main(void)
{
    printf("hello, world\n");
}
```

```
01111111 01000101 01001100 01000110 00000010 00000001 00000001 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000010 00000000 00111110 00000000 00000001 00000000 00000000 00000000
10110000 00000101 01000000 00000000 00000000 00000000 00000000 00000000
01000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
11010000 00010011 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 01000000 00000000 00111000 00000000
00001001 00000000 01000000 00000000 00100100 00000000 00100001 00000000
00000110 00000000 00000000 00000000 00000101 00000000 00000000 00000000
01000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
01000000 00000000 01000000 00000000 00000000 00000000 00000000 00000000
01000000 00000000 01000000 00000000 00000000 00000000 00000000 00000000
11111000 00000001 00000000 00000000 00000000 00000000 00000000 00000000
11111000 00000001 00000000 00000000 00000000 00000000 00000000 00000000
00001000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000011 00000000 00000000 00000000 00000100 00000000 00000000 00000000
00111000 00000010 00000000 00000000 00000000 00000000 00000000 00000000
...
```

```
make hello

./hello
```

```
clang hello.c
./a.out
```

default name

assembly output.

output

customise name

```
clang -o hello hello.c
./hello
```

rm → remove file

```c
#include <stdio.h>


int main(void)
{
    printf("hello, world\n");
}
```

```c
#include <cs50.h>
#include <stdio.h>


int main(void)
{
    string name = get_string("What's your name? ");
    printf("hello, %s\n", name);
}
```

placeholder

implement

Standard IO not need to add.

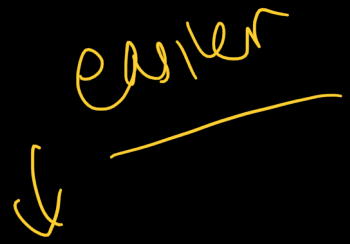
in order to use ← get_string

```
clang -o hello hello.c -lcs50

./hello
```

↑ link in cs50 to complile

* -lm ↑ Link math




```
make hello

./hello
```

compiling

#include

- preprocessing
- compiling
- assembling
- linking

preprocessing

compiling

assembling

linking

```
#include <cs50.h>
#include <stdio.h>

int main(void)
{
    string name = get_string("What's your name? ");
    printf("hello, %s\n", name);
}
```

```c
#include <cs50.h>
#include <stdio.h>


int main(void)
{
    string name = get_string("What's your name? ");
    printf("hello, %s\n", name);
}
```

```
string get_string(string prompt);
#include <stdio.h>


int main(void)
{
    string name = get_string("What's your name? ");
    printf("hello, %s\n", name);
}
```

```
string get_string(string prompt);
#include <stdio.h>


int main(void)
{
    string name = get_string("What's your name? ");
    printf("hello, %s\n", name);
}
```

```
string get_string(string prompt);
int printf(string format, ...);


int main(void)
{
    string name = get_string("What's your name? ");
    printf("hello, %s\n", name);
}
```

```
...
string get_string(string prompt);
int printf(string format, ...);
...

int main(void)
{
    string name = get_string("What's your name? ");
    printf("hello, %s\n", name);
}
```

preprocessing

compiling

assembling

linking

```
...
string get_string(string prompt);
int printf(string format, ...);
...

int main(void)
{
    string name = get_string("What's your name? ");
    printf("hello, %s\n", name);
}
```

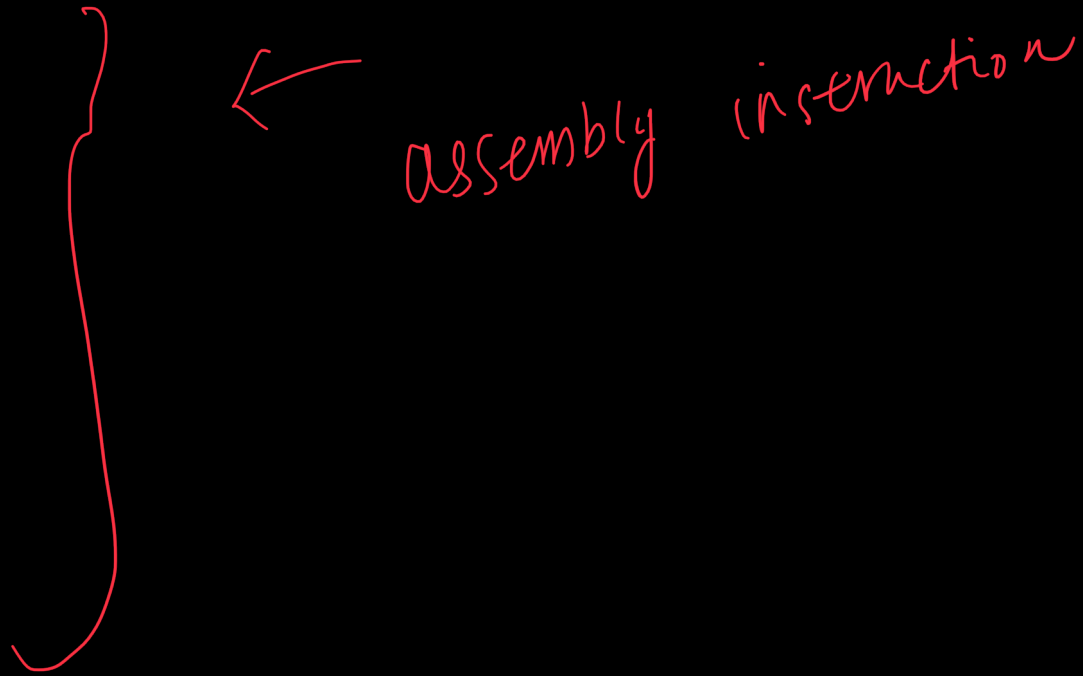
```
...
main:                                   # @main
    .cfi_startproc
# BB#0:
    pushq    %rbp
.Ltmp0:
    .cfi_def_cfa_offset 16
.Ltmp1:
    .cfi_offset %rbp, -16
    movq    %rsp, %rbp
.Ltmp2:
    .cfi_def_cfa_register %rbp
    subq    $16, %rsp
    xorl    %eax, %eax
    movl    %eax, %edi
    movabsq    $.L.str, %rsi
    movb    $0, %al
    callq    get_string
    movabsq    $.L.str.1, %rdi
    movq    %rax, -8(%rbp)
    movq    -8(%rbp), %rsi
    movb    $0, %al
    callq    printf
    ...
```

assembly language

```
...
main:                                   # @main
    .cfi_startproc
# BB#0:
    pushq    %rbp
.Ltmp0:
    .cfi_def_cfa_offset 16
.Ltmp1:
    .cfi_offset %rbp, -16
    movq    %rsp, %rbp
.Ltmp2:
    .cfi_def_cfa_register %rbp
    subq    $16, %rsp
    xorl    %eax, %eax
    movl    %eax, %edi
    movabsq    $.L.str, %rsi
    movb    $0, %al
    callq    get_string
    movabsq    $.L.str.1, %rdi
    movq    %rax, -8(%rbp)
    movq    -8(%rbp), %rsi
    movb    $0, %al
    callq    printf
    ...
```

Computer instruction

```
...
main:                                   # @main
    .cfi_startproc
# BB#0:
    pushq    %rbp
.Ltmp0:
    .cfi_def_cfa_offset 16
.Ltmp1:
    .cfi_offset %rbp, -16
    movq    %rsp, %rbp
.Ltmp2:
    .cfi_def_cfa_register %rbp
    subq    $16, %rsp
    xorl    %eax, %eax
    movl    %eax, %edi
    movabsq    $.L.str, %rsi
    movb    $0, %al
    callq    get_string
    movabsq    $.L.str.1, %rdi
    movq    %rax, -8(%rbp)
    movq    -8(%rbp), %rsi
    movb    $0, %al
    callq    printf
    ...
```



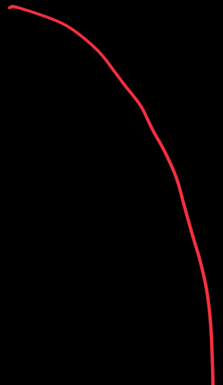
preprocessing

compiling

assembling ← turn into 0 and 1

linking

```
...
main:                                   # @main
    .cfi_startproc
# BB#0:
    pushq    %rbp
.Ltmp0:
    .cfi_def_cfa_offset 16
.Ltmp1:
    .cfi_offset %rbp, -16
    movq    %rsp, %rbp
.Ltmp2:
    .cfi_def_cfa_register %rbp
    subq    $16, %rsp
    xorl    %eax, %eax
    movl    %eax, %edi
    movabsq    $.L.str, %rsi
    movb    $0, %al
    callq    get_string
    movabsq    $.L.str.1, %rdi
    movq    %rax, -8(%rbp)
    movq    -8(%rbp), %rsi
    movb    $0, %al
    callq    printf
    ...
```



```
01111111010001010100110001000110
00000010000000010000000100000000
00000000000000000000000000000000
00000000000000000000000000000000
00000001000000000011111000000000
00000001000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
10100000000000100000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
01000000000000000000000000000000
00000000000000000100000000000000
00001010000000000000000100000000
01010101010010001000100111100101
01001000100000111110110000010000
00110001110000001000100111000111
01001000101111100000000000000000
00000000000000000000000000000000
00000000000000001011000000000000
11101000000000000000000000000000
00000000010010001011111100000000
00000000000000000000000000000000
00000000000000000000001001000
...
```

preprocessing

compiling

assembling

linking

Link all o/l to a single file.

```c
#include <cs50.h>
#include <stdio.h>


int main(void)
{
    string name = get_string("What's your name? ");
    printf("hello, %s\n", name);
}
```

```c
#include <cs50.h>
#include <stdio.h>


int main(void)
{
    string name = get_string("What's your name? ");
    printf("hello, %s\n", name);
}
```

```c
#include <cs50.h>
#include <stdio.h>


int main(void)
{
    string name = get_string("What's your name? ");
    printf("hello, %s\n", name);
}
```

```c
#include <cs50.h>
#include <stdio.h>


int main(void)
{
    string name = get_string("What's your name? ");
    printf("hello, %s\n", name);
}
```

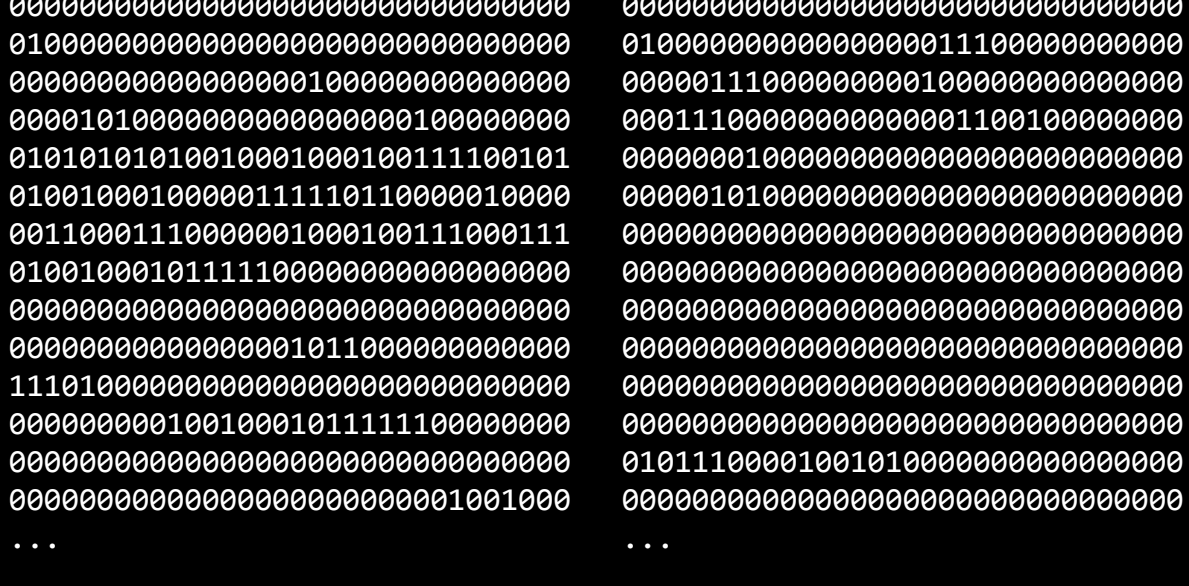
hello.c

hello.c

cs50.c

hello.c cs50.c stdio.c


cs50.c
stdio.c



stdio.c

0111111101000101010011000100011000000010000000010000000100000000000000000000000000000000000000000
0000000000000000000000000000000010000000000111110000000000000000001000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000010100000000000010000000000000000000000000000000000000000000000000000000000000000000
0000000001000000000000000000000000000000000000000000000000100000000000000000010100000000000000000001000000
000101010101001000100010011110010101001000100000111110110000010000001100011100000010001001110001110100
1000101111100000000000000000000000000000000000000000000000000000101100000000000001110100000
0000000000000000000000000000001001000101111110000000000000000000000000000000000000000000000000000
0000000001001000...01111111010001010100110001000110000000100000000100000001000000000000000000000000000
00000000000000000000000000000000000000000110000000000111110000000000000000001000000000000000000
0000000110000000000011110000000000000000000000000000000000000000100000000000000000000000000000
00000000000000000000000000000000101000001100100000000000000000000000000000000000000000000000000
0000000000000000000000000000100000000000000000111000000000000000001110000000001000000000000000000111000000
00000000110010000000000000000100000000000000000000000000000010100000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000101110000100101000000000000000
000000000000000000000000000000000...00101111011011000110100101100010011000110010111001110011011011111
0010111000110110001000000010111101110101011100110111001000101111011011000110100101100010001011110111110
0000111000001101100101111100110110001101000010110101101100011010010110111001110101011110000010110101110
0111011011100111010100101111011011000110100101100010011000110101111101101110011011110110111001110011001101
1010000110000101110010011001010110010000101110011000010010000000100000001000000101010011010111110100111 0
0100010101000101010001000100010101000100001000000001010000001000000001011110110110001101001011000100010 11
1101111100000111000001101100101111100110110001101000010110101101100011010010110111001110101011110000010
1101011001110110111001110101001011110110110001100100001011010110110001101001011011100111010101111000000
1011010111100000111000001101100010110100110110001101 00...

preprocessing

compiling

assembling

linking

compiling

debugging

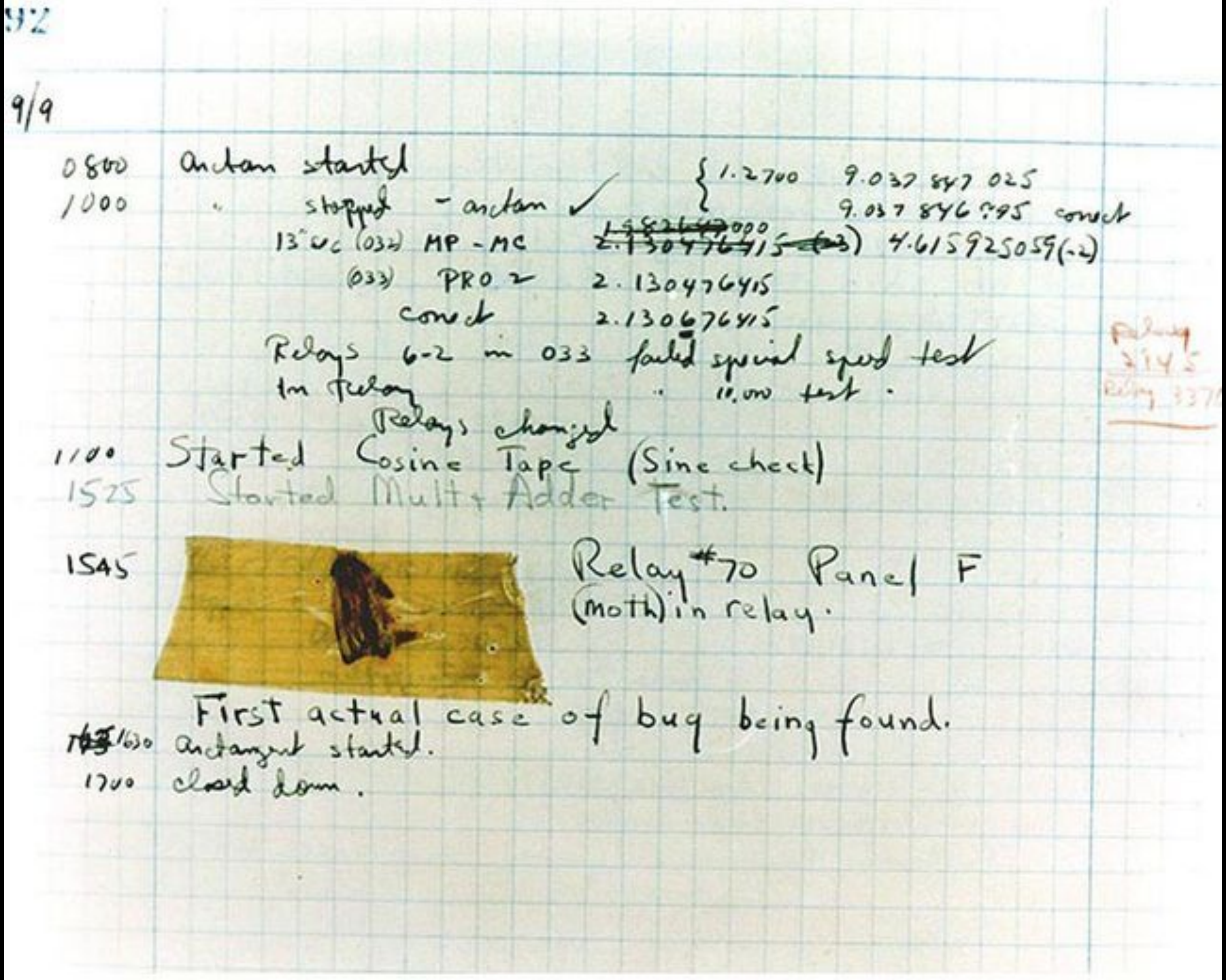


Photo # NH 96566-KN (Color) First Computer "Bug", 1947

9/9

0800 Arctan started
1000 " stopped - arctan ✓ {1.2700 9.037 847 025
9.037 846 995 correct
13" uc (032) MP - MC ~~1.982147000~~ ~~2.130476415~~ (-3) 4.615925059(-2)
(033) PRO 2 2.130476415
correct 2.130676415
Relays 6-2 in 033 failed special speed test
in relay · 11.000 test ·
Relays changed
1100 Started Cosine Tape (Sine check)
1525 Started Mult + Adder Test.
1545 Relay #70 Panel F
(moth) in relay.
First actual case of bug being found.
~~1630~~ 1630 Antangent started.
1700 closed down.

In relay . 11.00 test.

Relays changed

1100 Started Cosine Tape (Sine check)

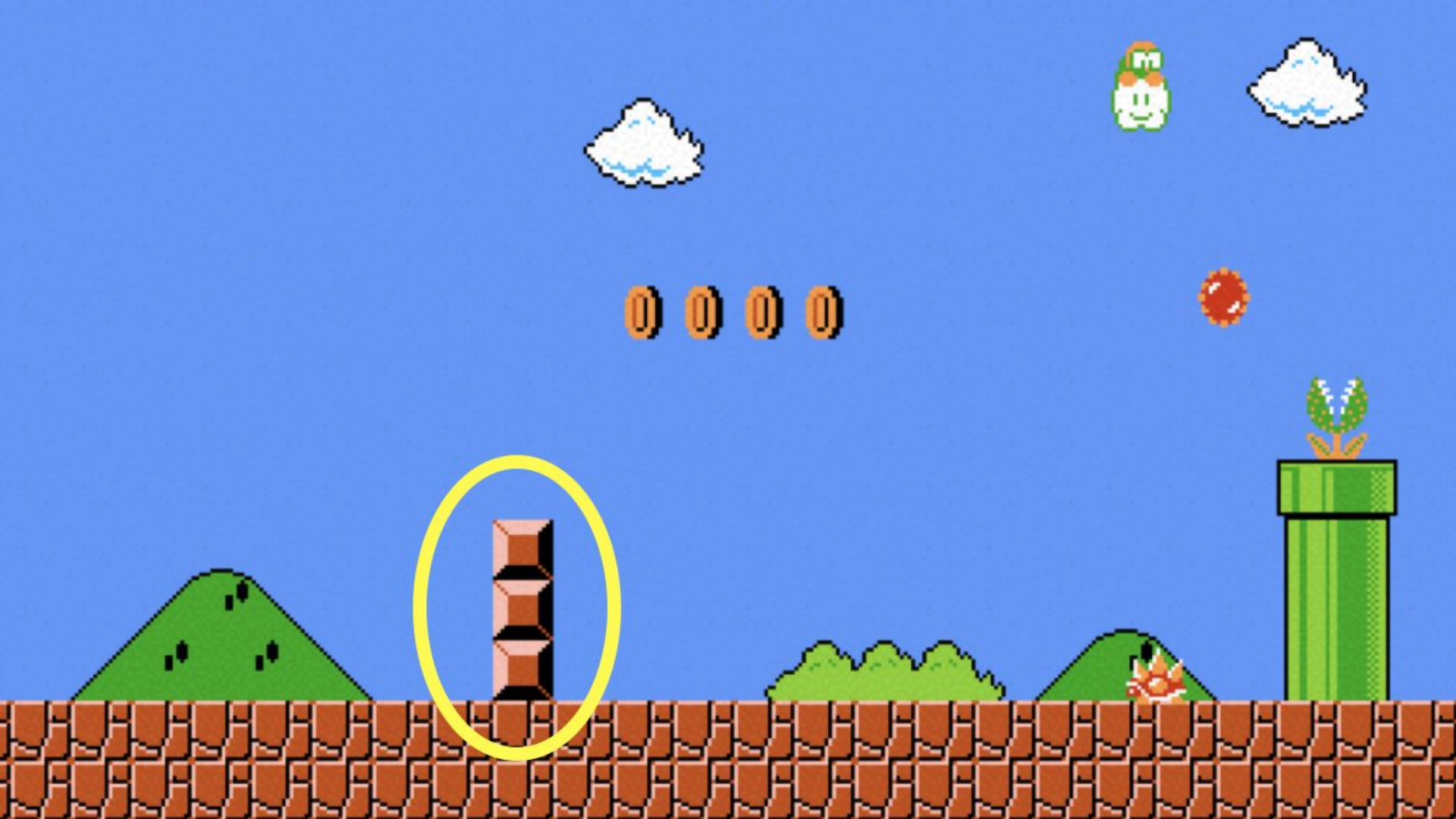
1525 Started Multy Adder Test.

1545  Relay #70 Panel F
(moth) in relay.

First actual case of bug being found.

~~1655~~ 1630 Arctangent started.

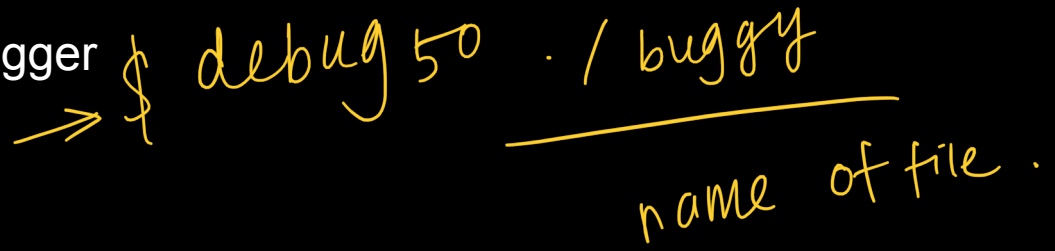
1700 closed down.



printf

printf

debugger



`printf`

debugger

rubber duck → talk to your own code one by one

types

```
bool

char

double

float

int

long

string

...
```

1 byte or 8 bits

bool	1 byte
char	1 byte
double	8 bytes
float	4 bytes
int	4 bytes
long	8 bytes
string	? bytes
...	

0/1 false or true

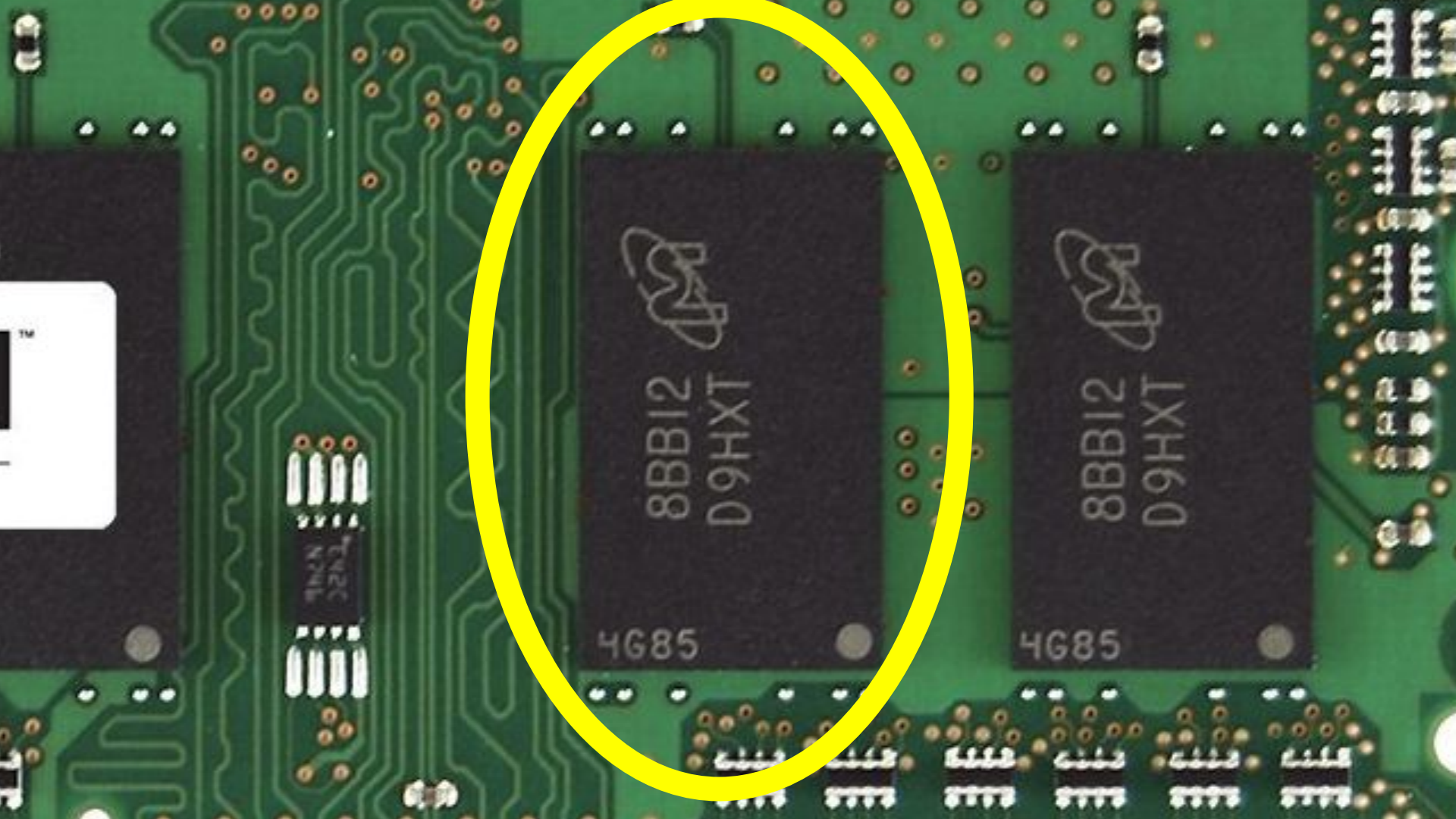
256

twice

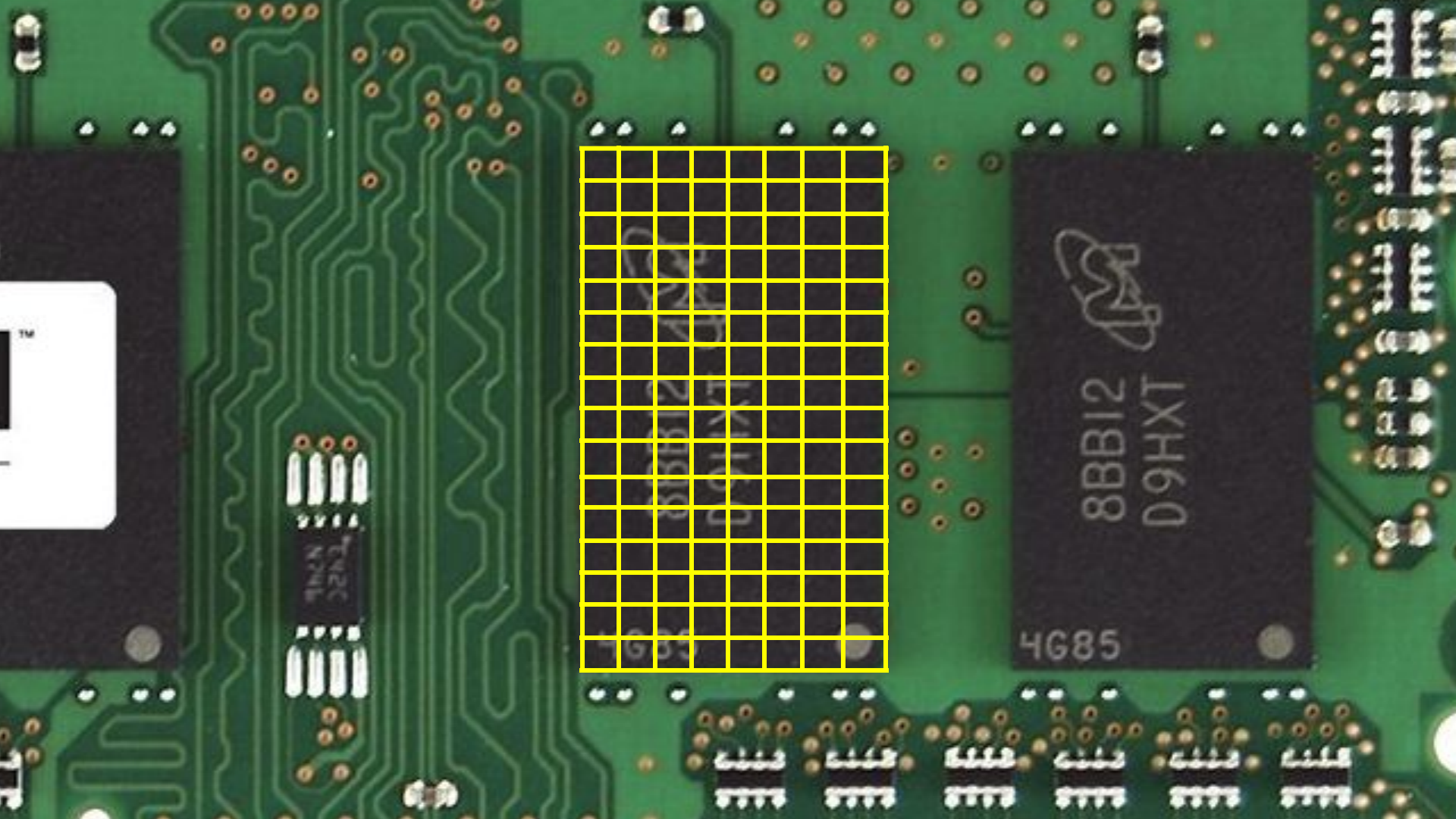

crucial
by Micron
8BBI2
D9HXT
8BBI2
D9HXT
4G85
4G85
4G85
47J6

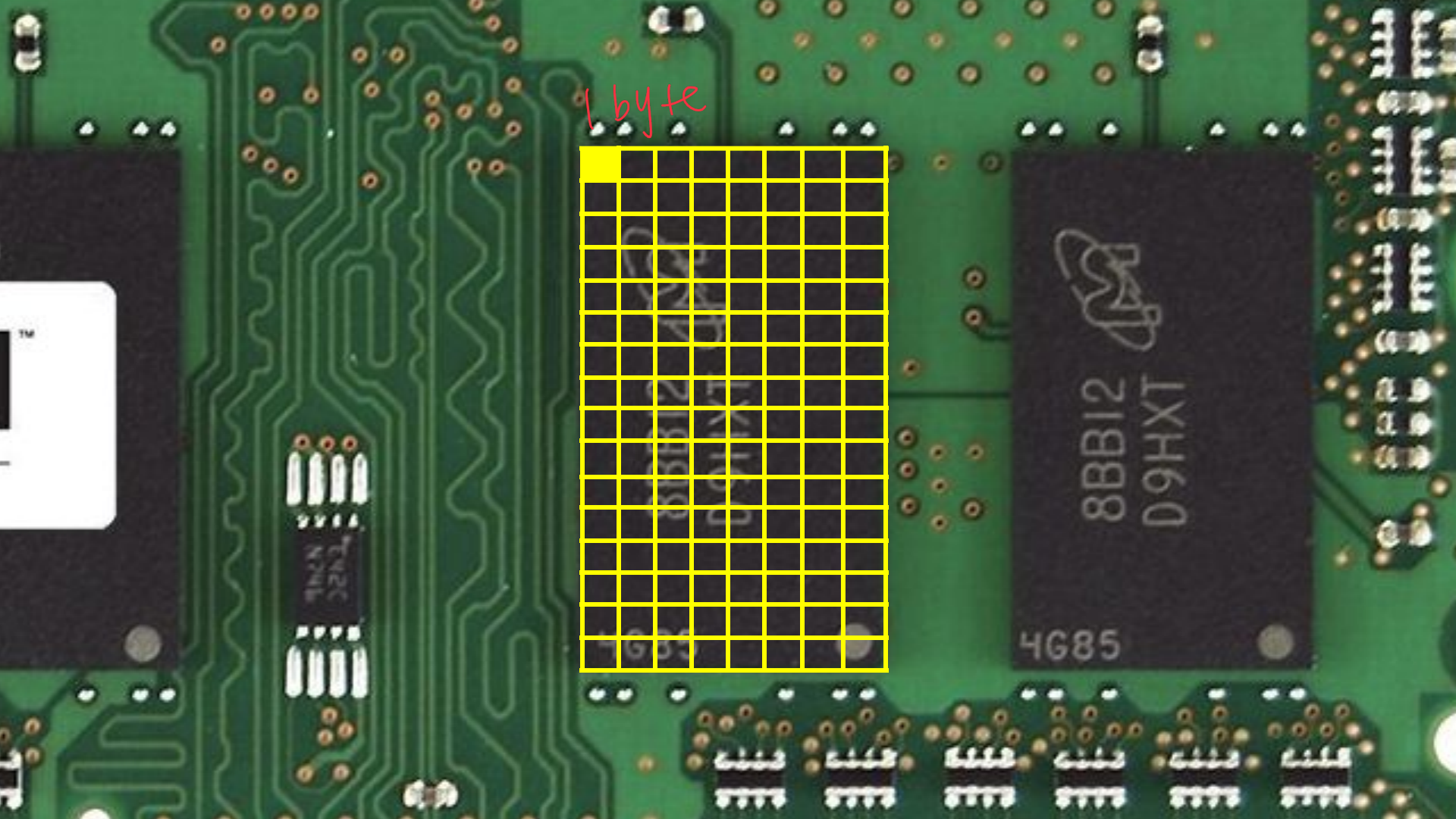
1 gb

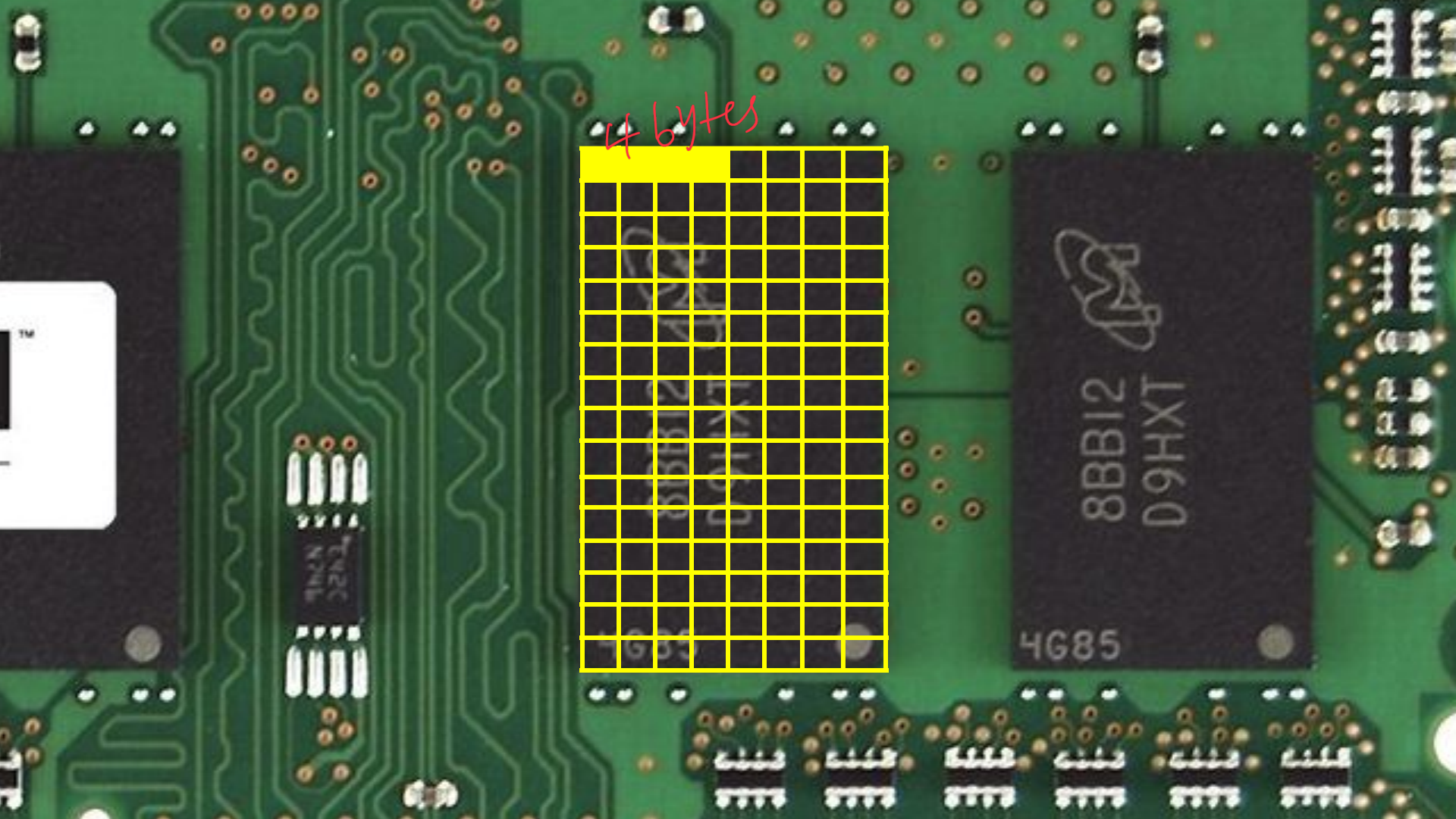

crucial
by Micron
8BB12
D9HXT
8BB12
D9HXT
4G85
4G85
4G85
47J6

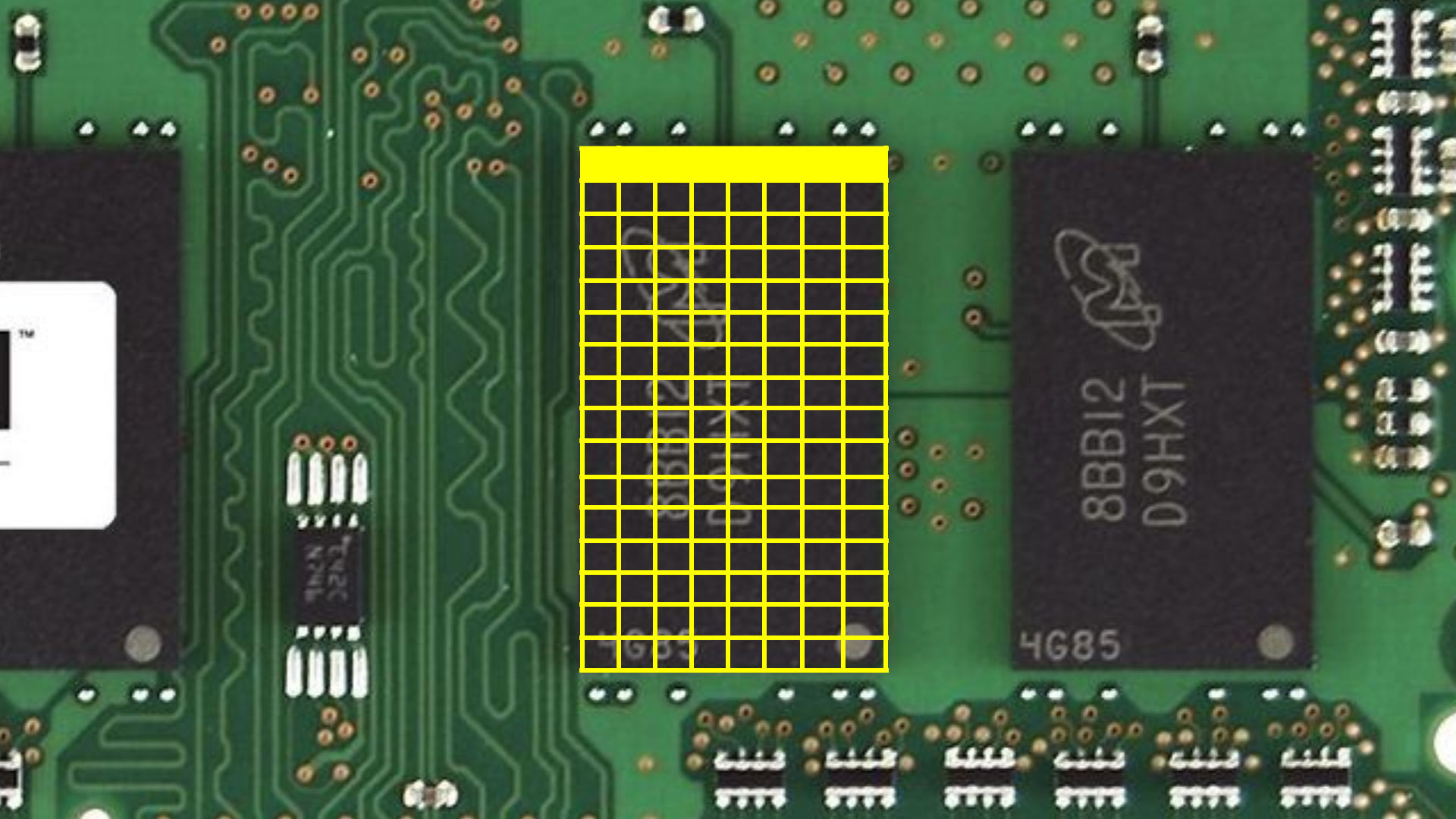

8BBI2
D9HXT
4G85
8BBI2
D9HXT
4G85

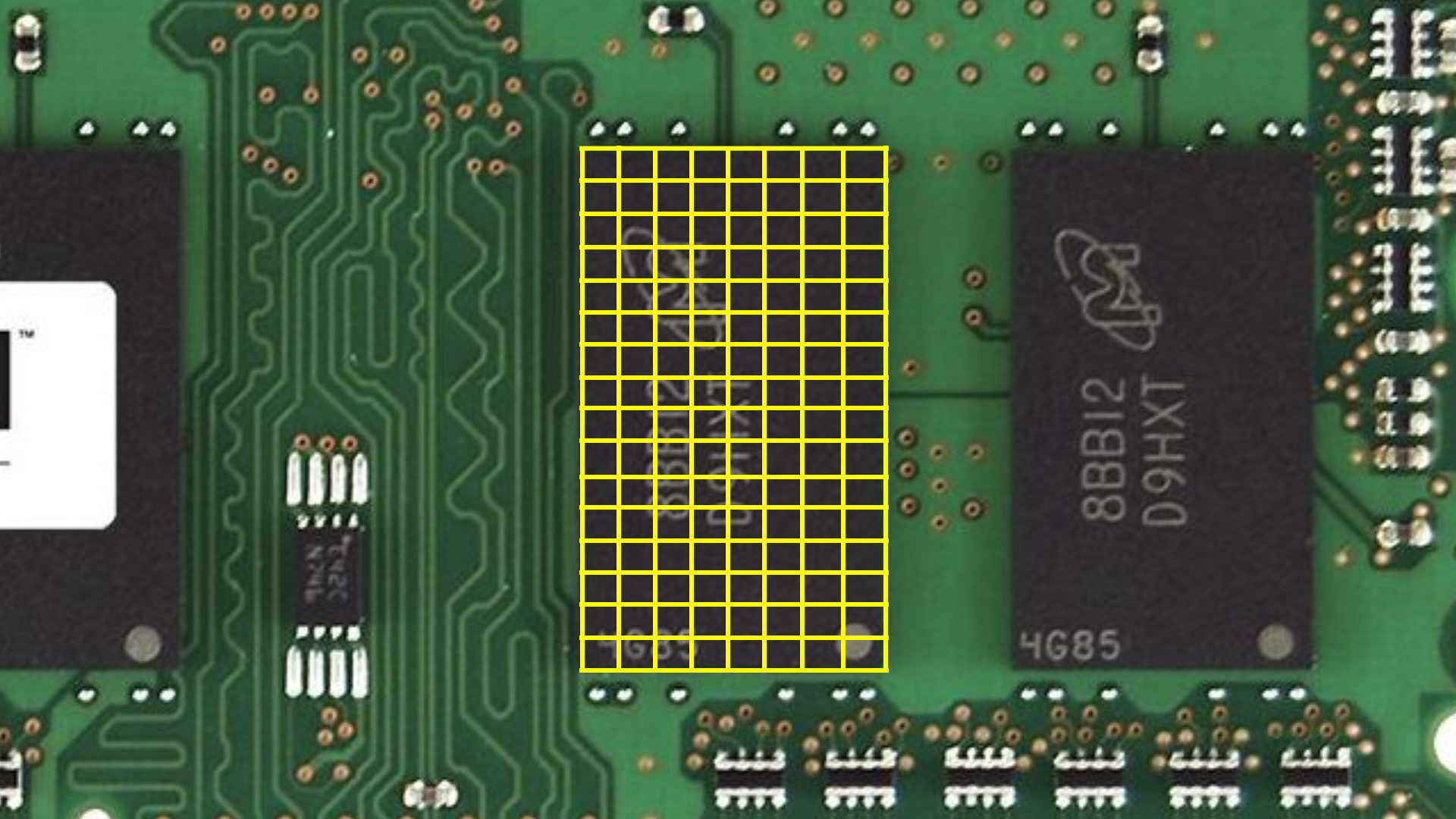

8BBI2
D9HXT
4G85
8BBI2
D9HXT
4G85

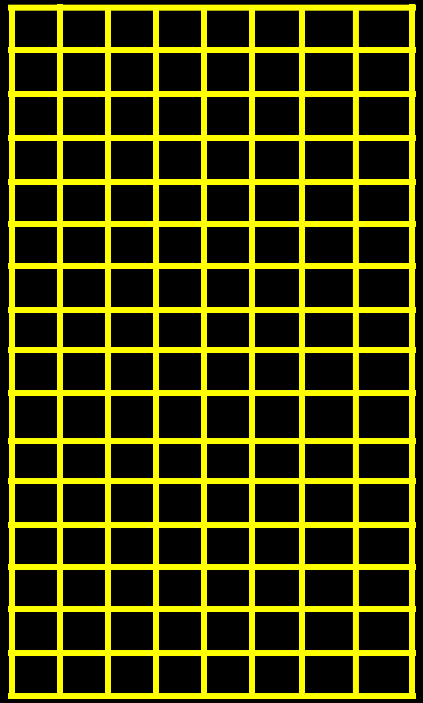

8BBI2
D9HXT
4G85


1 byte
8BBI2
D9HXT
4G85


4 bytes
8BBI2
D9HXT
4G85


8BBI2
D9HXT
4G85

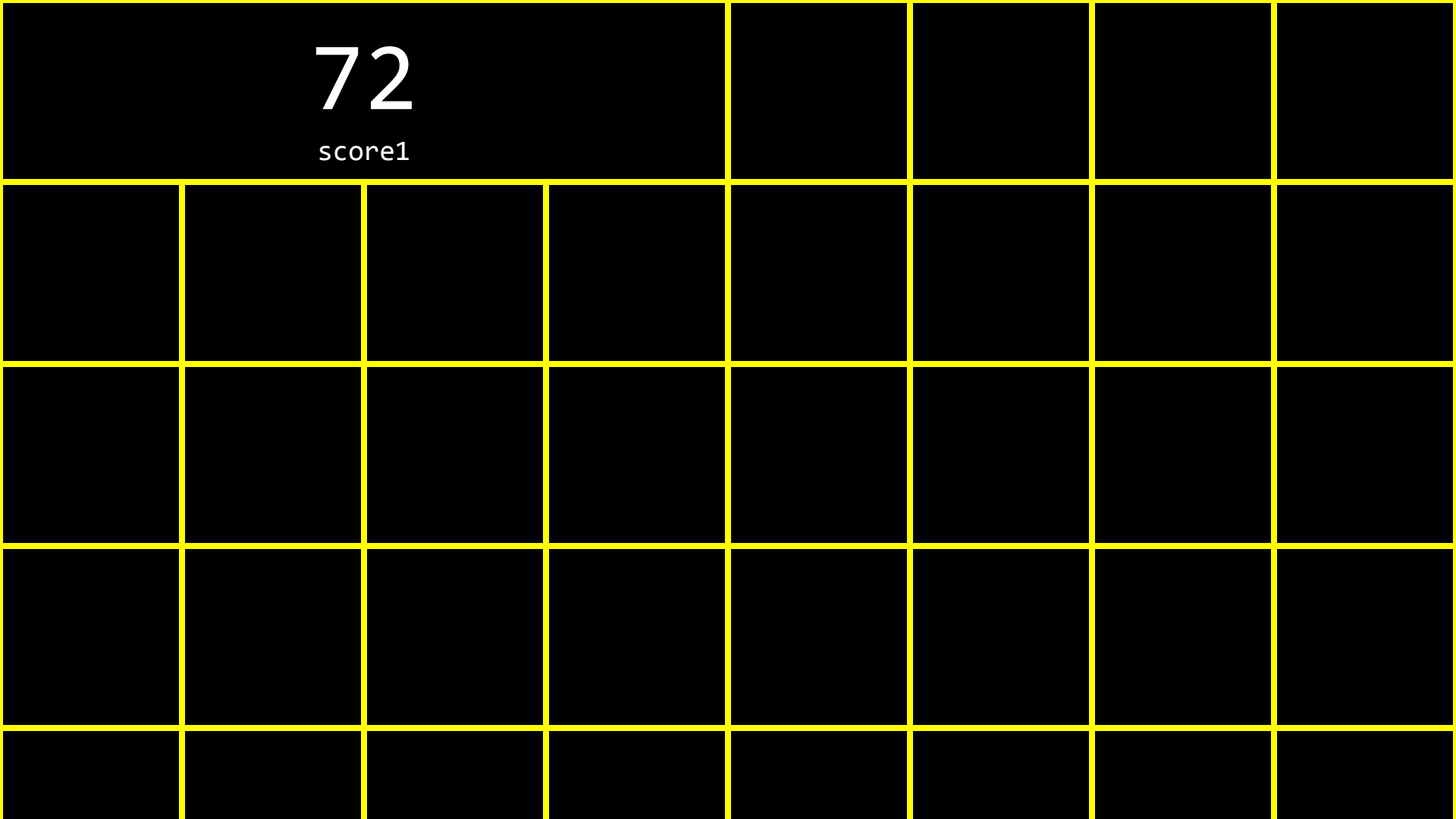

8BBI2
D9HXT
4G85



```
int score1 = 72;

int score2 = 73;

int score3 = 33;
```


72
score1

72

score1

73

score2

72

score1

73

score2

33

score3

00000000000000000000001001000

score1

00000000000000000000001001001

score2

00000000000000000000100001

score3

```
int score1 = 72;

int score2 = 73;

int score3 = 33;
```

arrays

store multiple values of the same type back to back contiguously


```
int scores[3];
```

```
int scores[3];

scores[0] = 72;

scores[1] = 73;

scores[2] = 33;
```

72

scores[0]

73

scores[1]

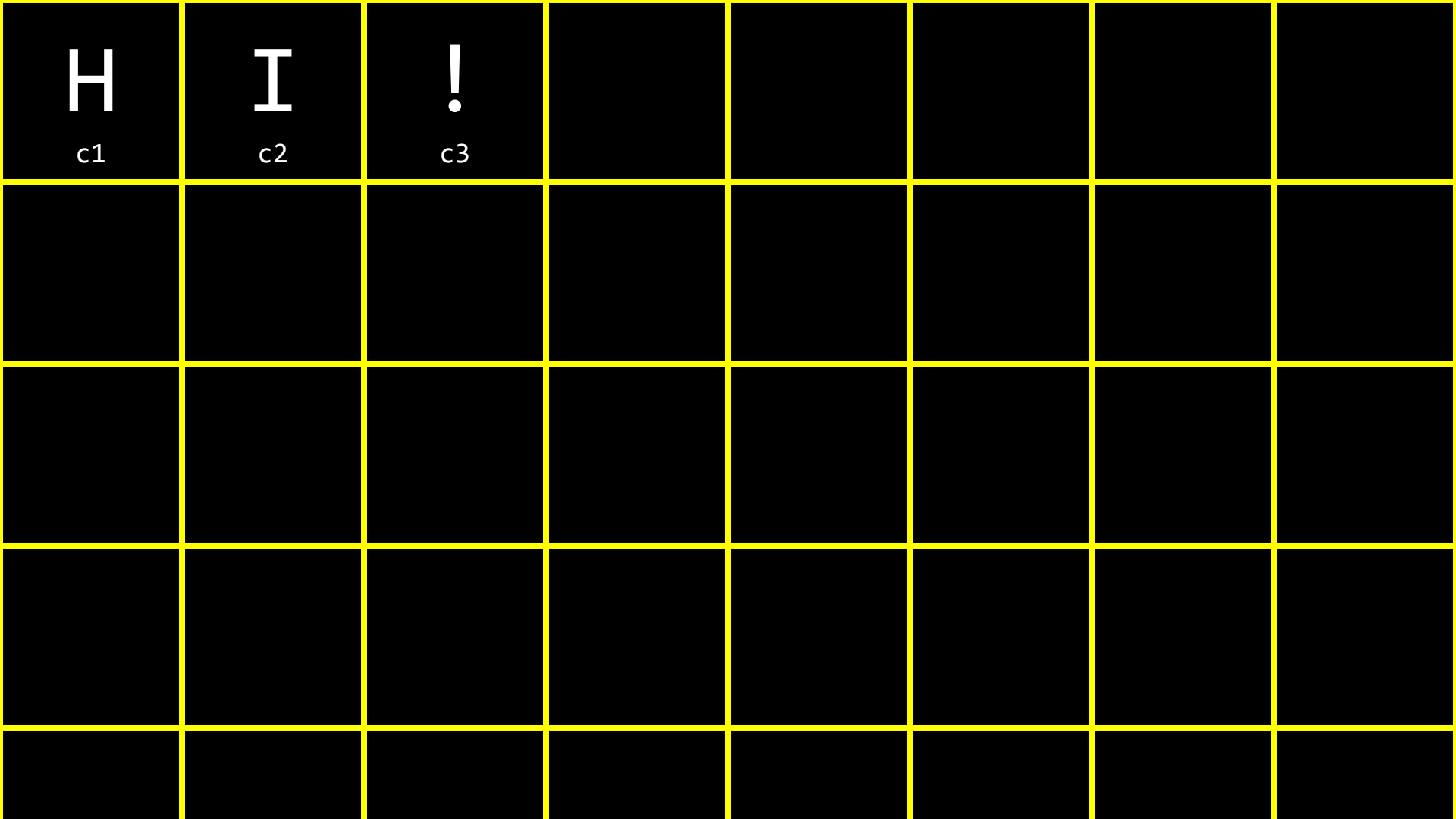
33

scores[2]

```
char c1 = 'H';

char c2 = 'I';

char c3 = '!';
```


H
I
!
c1
c2
c3

72 c1	73 c2	33 c3					

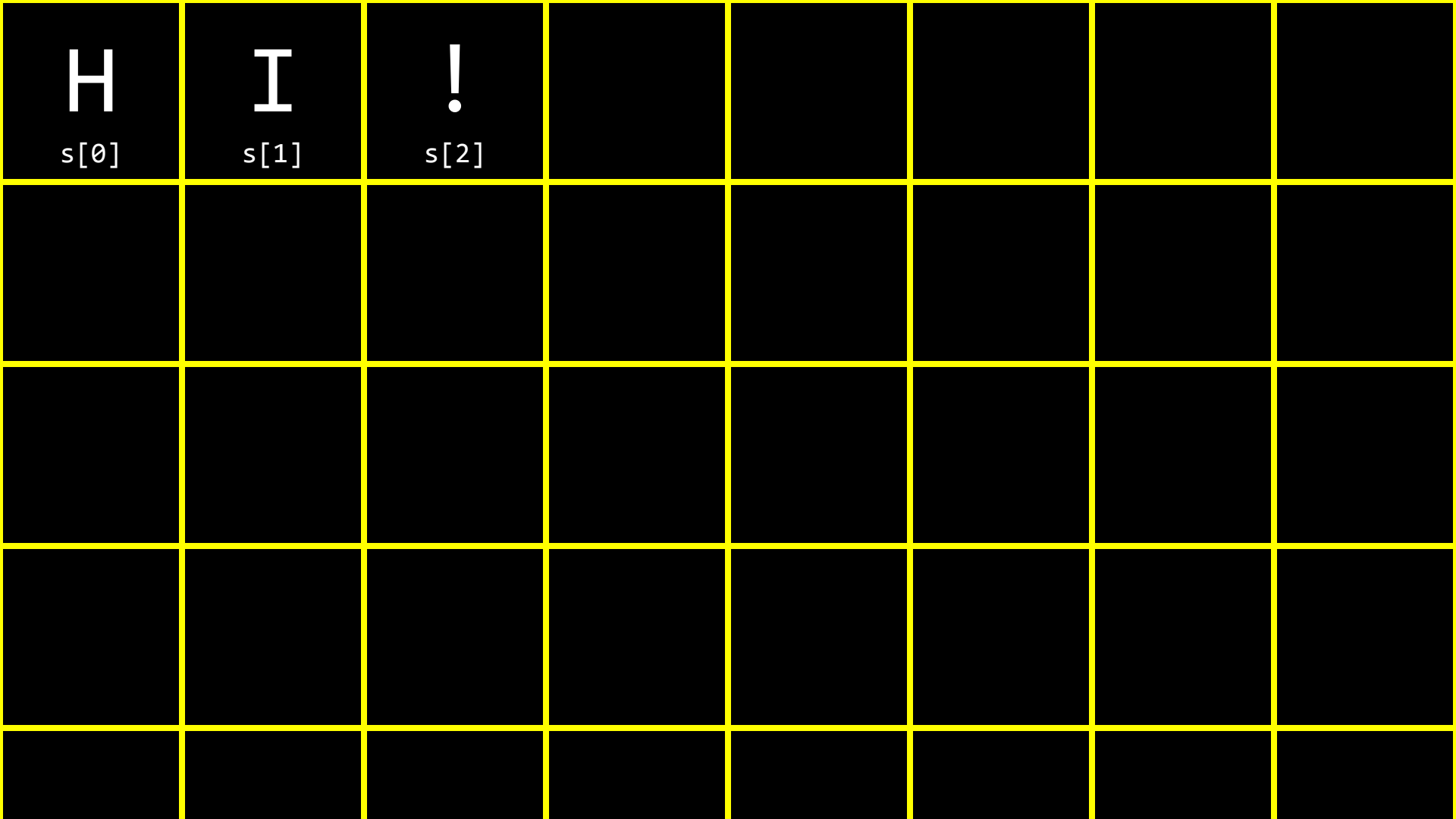
01001000 c1	01001001 c2	00100001 c3					

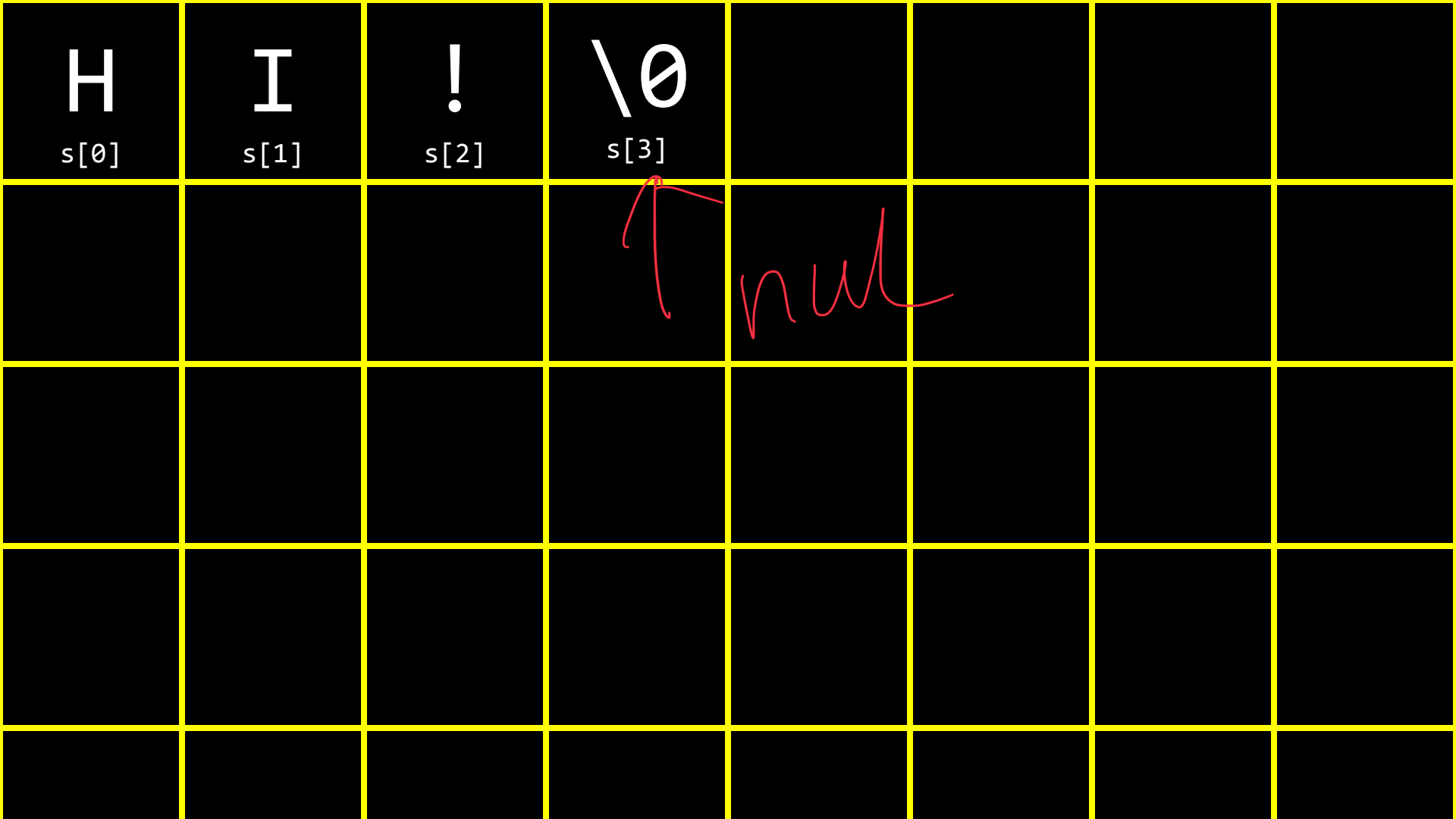
```
string s = "HI!";
```

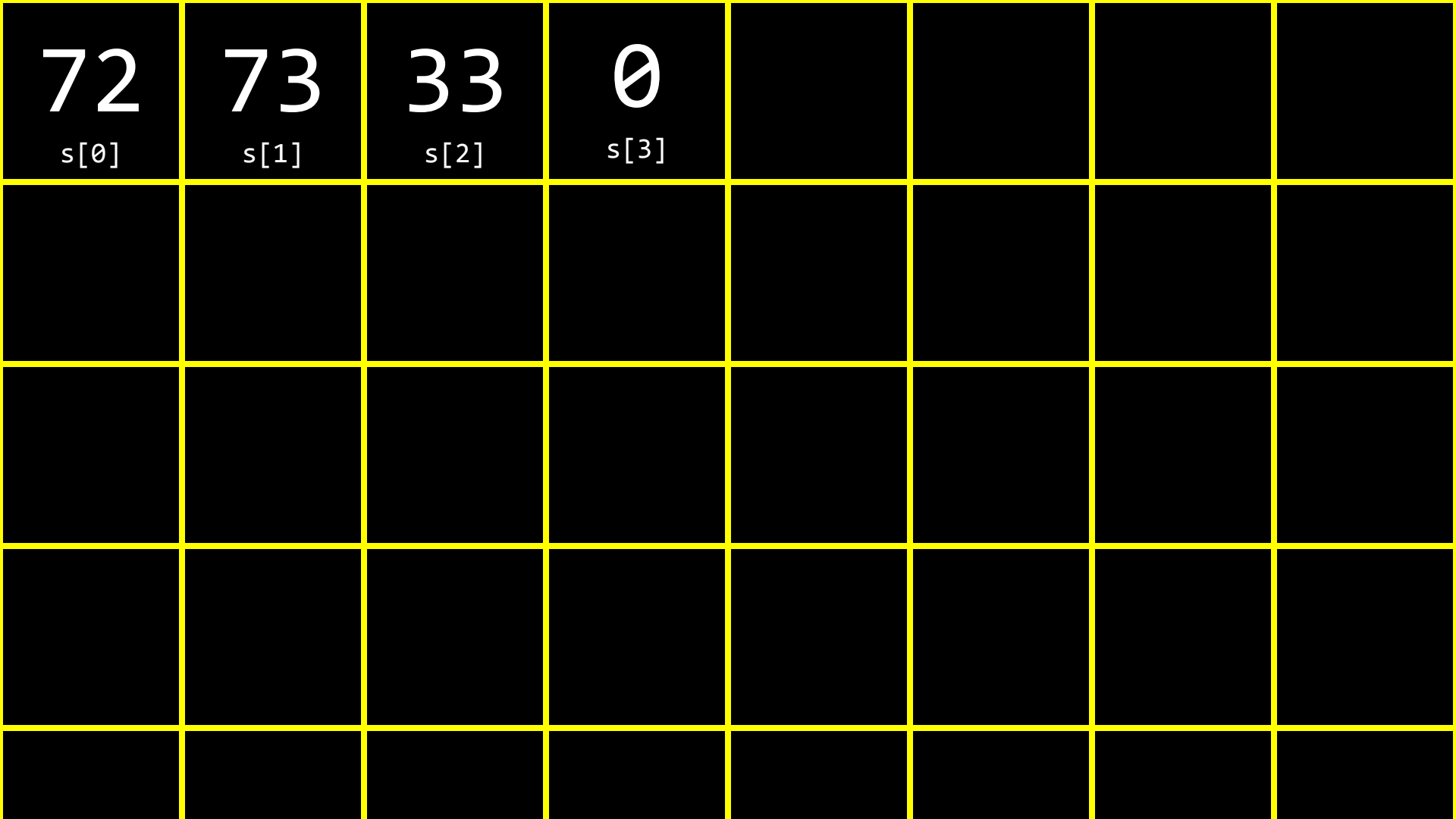
String is like array of characters.

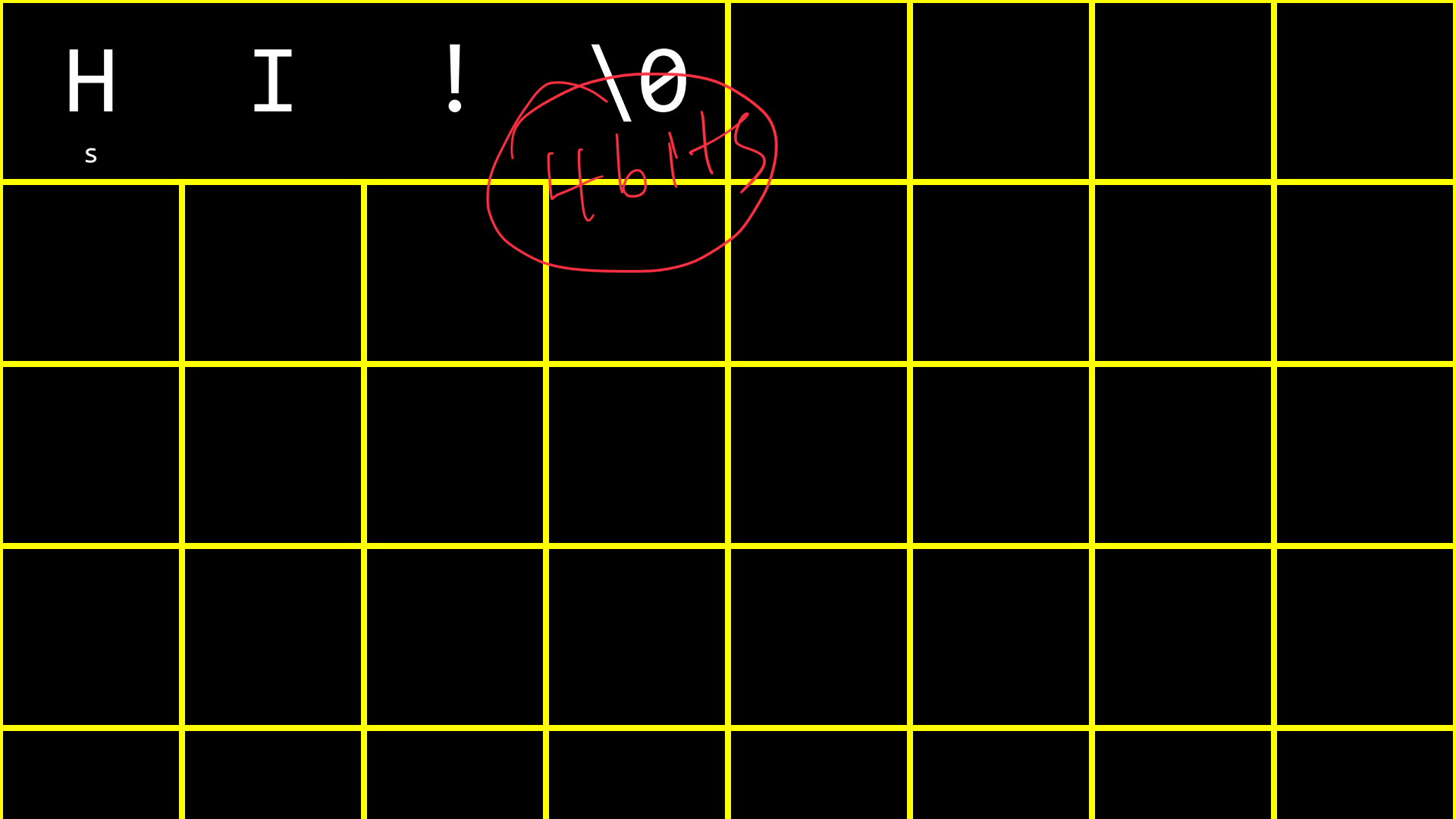
H I !

s


H
I
!
s[0]
s[1]
s[2]


H
I
!
\0
s[0]
s[1]
s[2]
s[3]
null

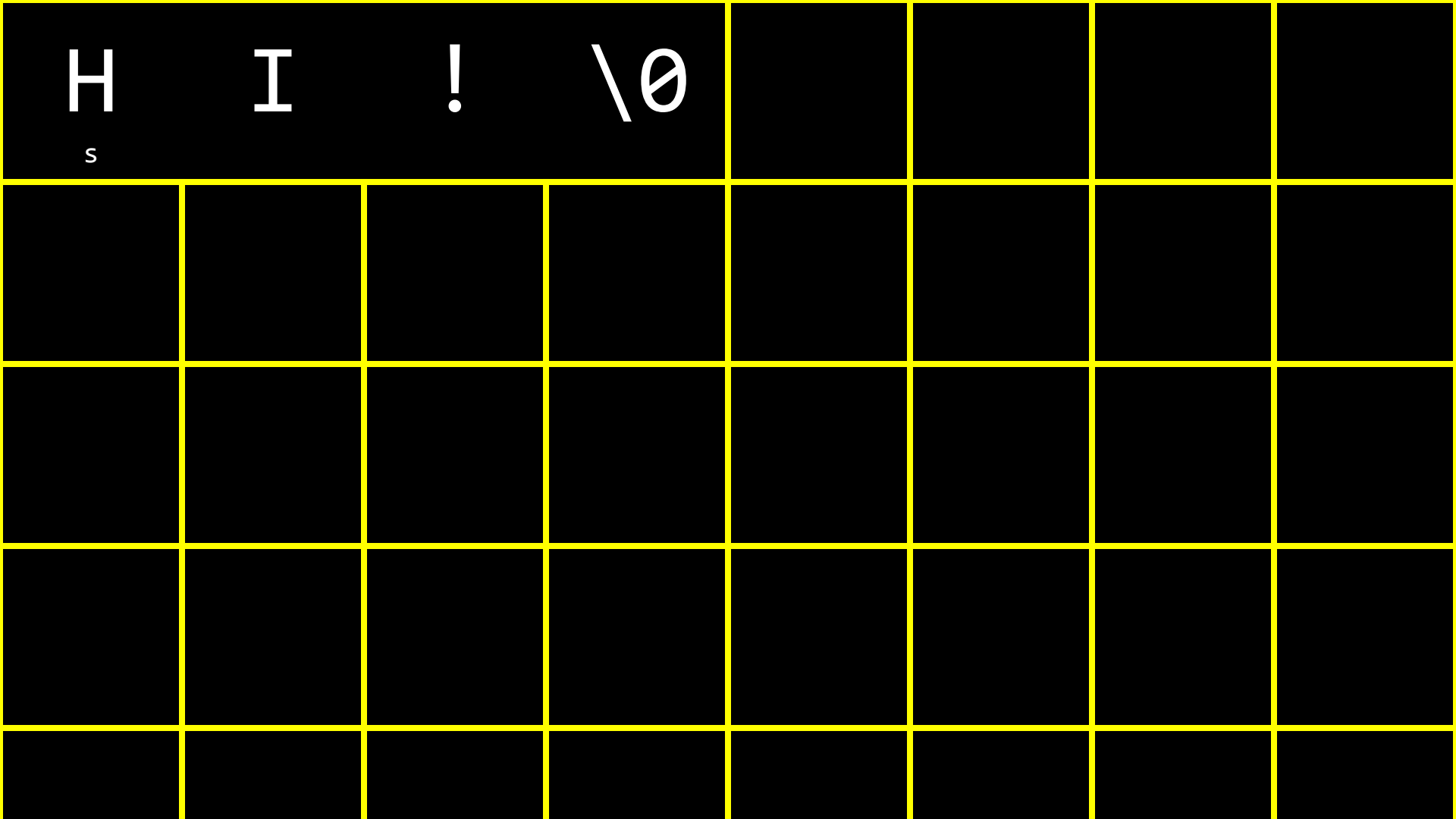

72
s[0]
73
s[1]
33
s[2]
0
s[3]

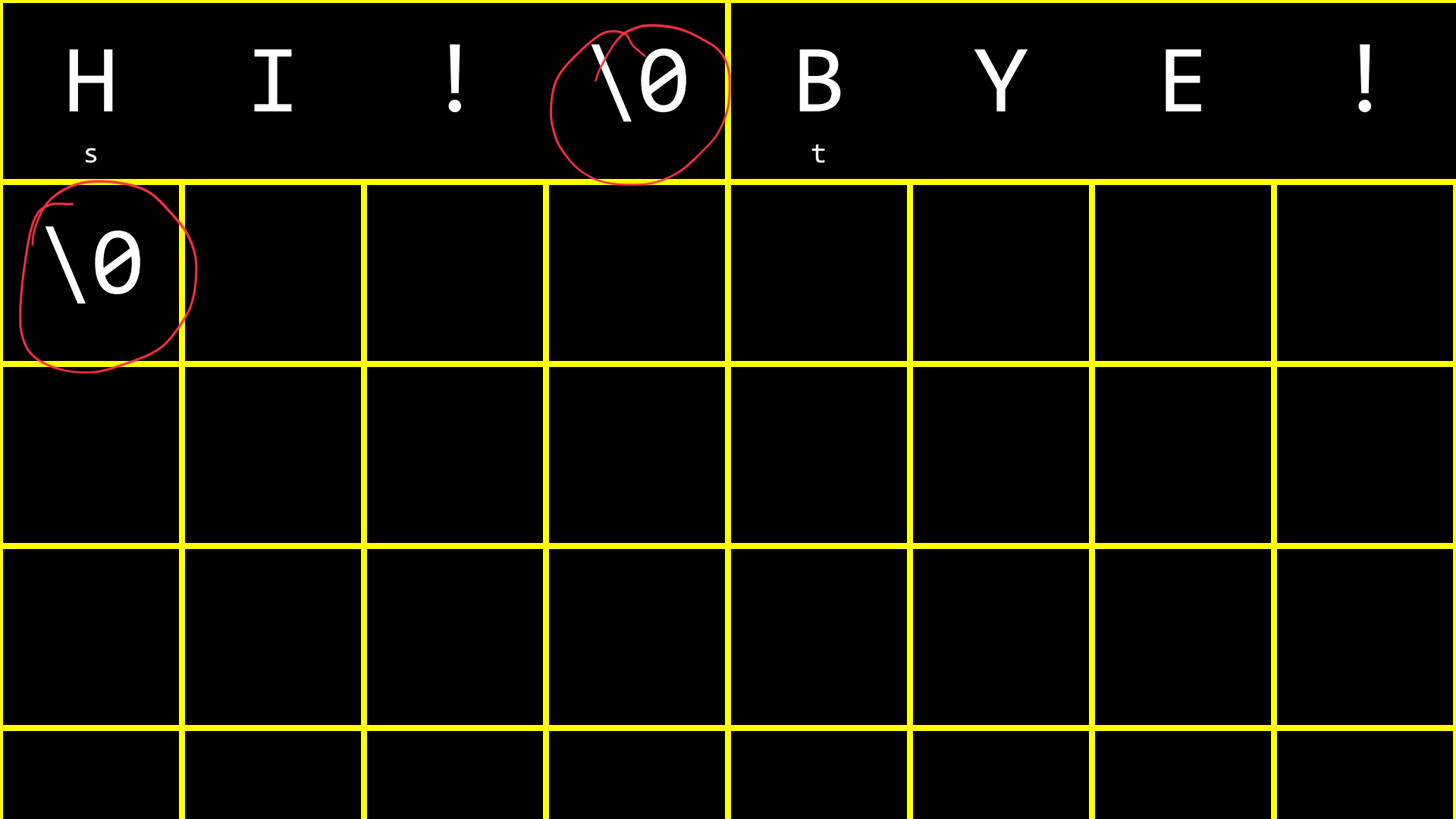

H I ! \0
s
Hints

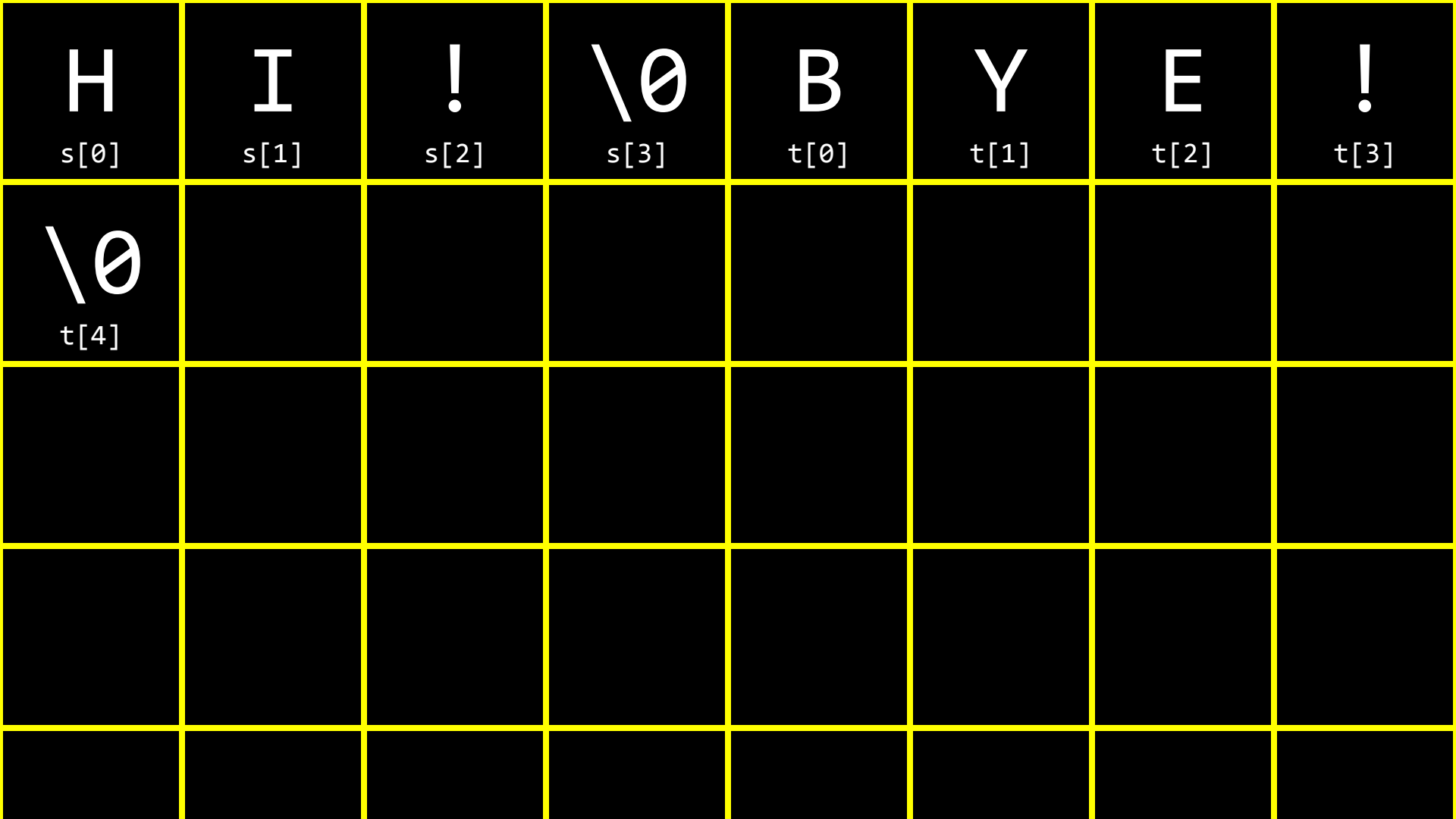
NUL

```
string s = "HI!";

string t = "BYE!";
```


H I ! \0
s


H I ! \0 B Y E !
s
t
\0


H
s[0]
I
s[1]
!
s[2]
\0
s[3]
B
t[0]
Y
t[1]
E
t[2]
!
t[3]
\0
t[4]

```
string words[2];

words[0] = "HI!";

words[1] = "BYE!";
```

string

manual pages

command-line arguments

```c
#include <stdio.h>

int main(void)
{
    ...
}
```

```c
#include <stdio.h>

int main(void)
{
    ...
}
```

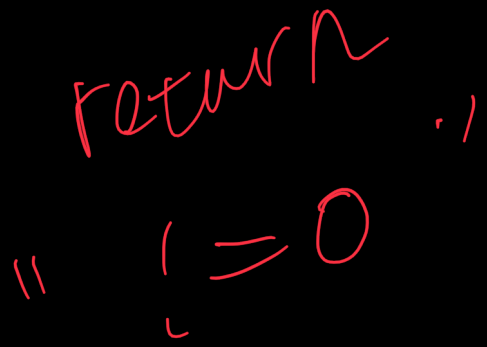
```
#include <stdio.h>

int main(int argc, string argv[])
{
    ...
}
```

argument vector

argument count

exit status



```c
#include <stdio.h>

int main(void)
{
    ...
}
```

```c
#include <stdio.h>

int main(void)
{
    ...
}
```

readability

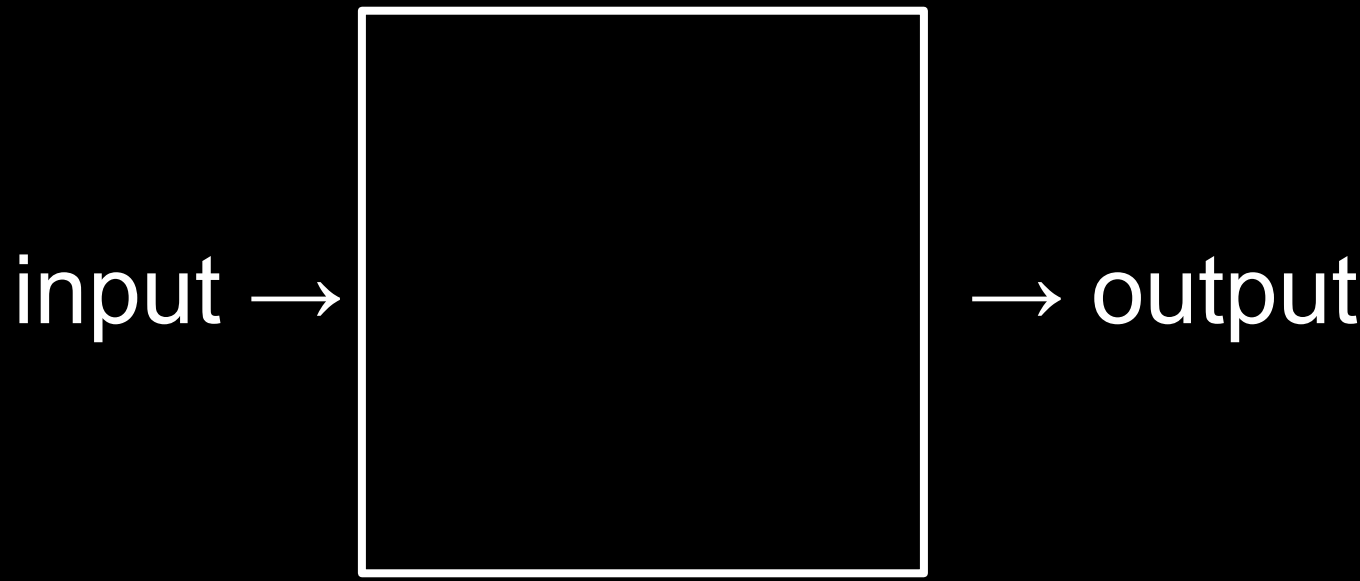
One fish. Two fish. Red fish. Blue fish.

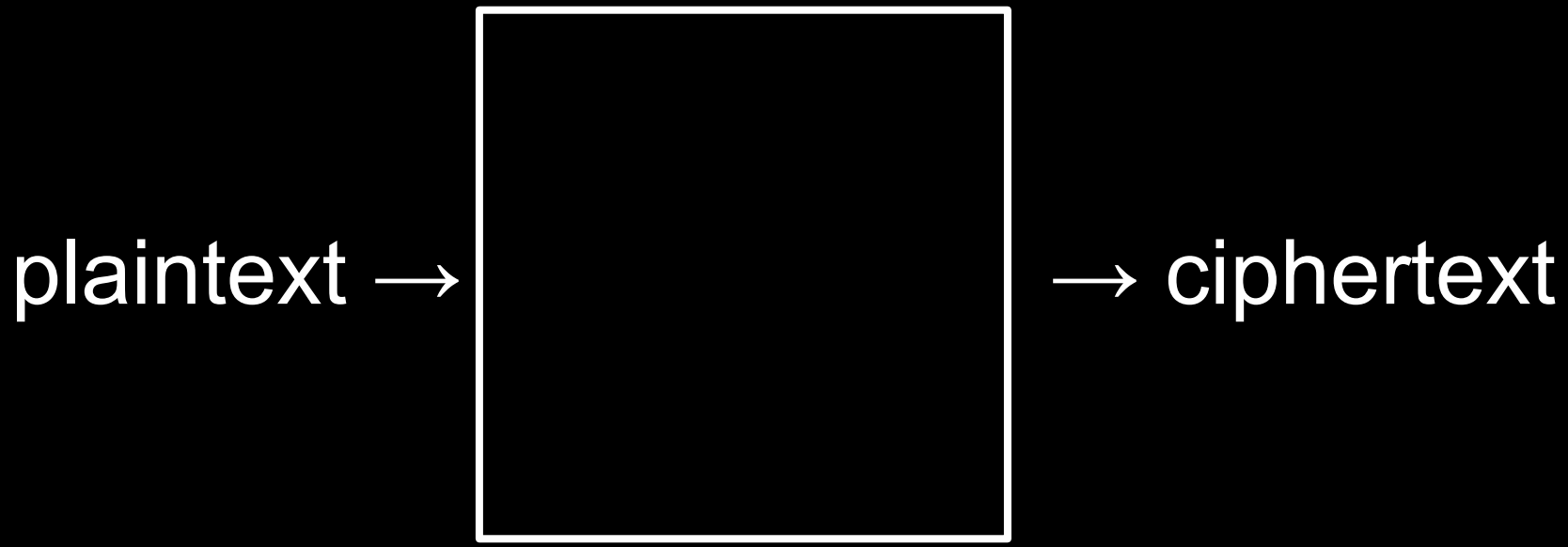
Before Grade 1

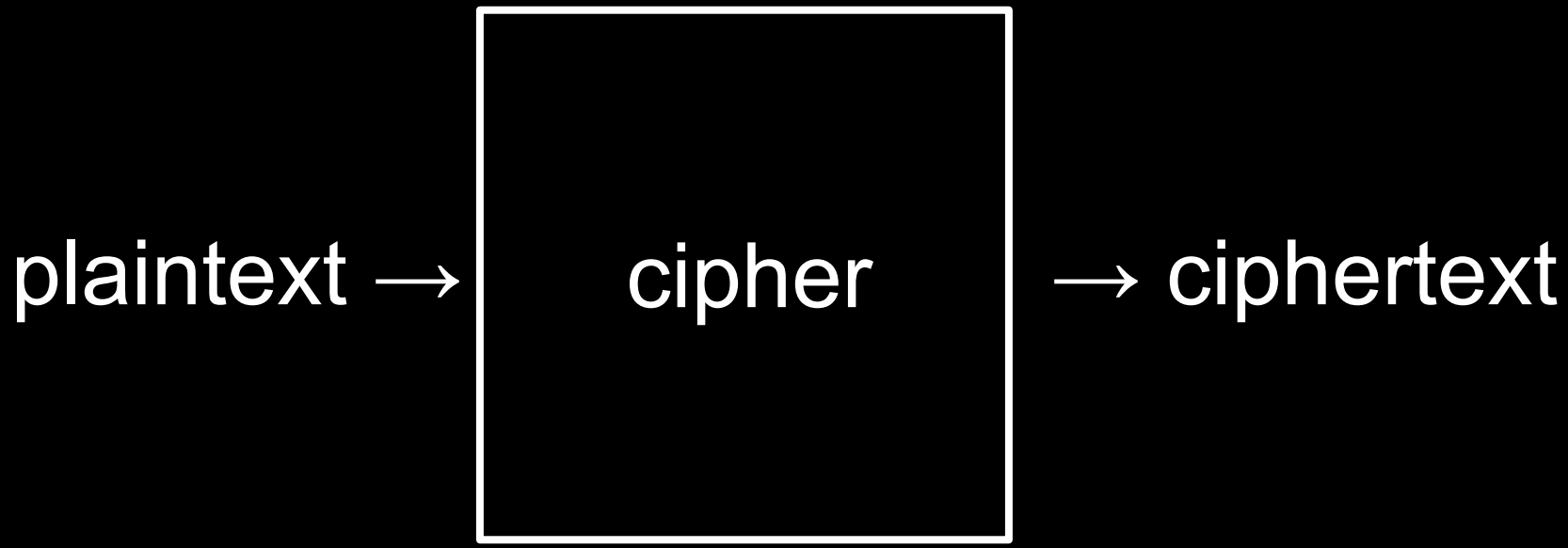
Mr. and Mrs. Dursley, of number four, Privet Drive, were proud to say that they were perfectly normal, thank you very much. They were the last people you'd expect to be involved in anything strange or mysterious, because they just didn't hold with such nonsense...

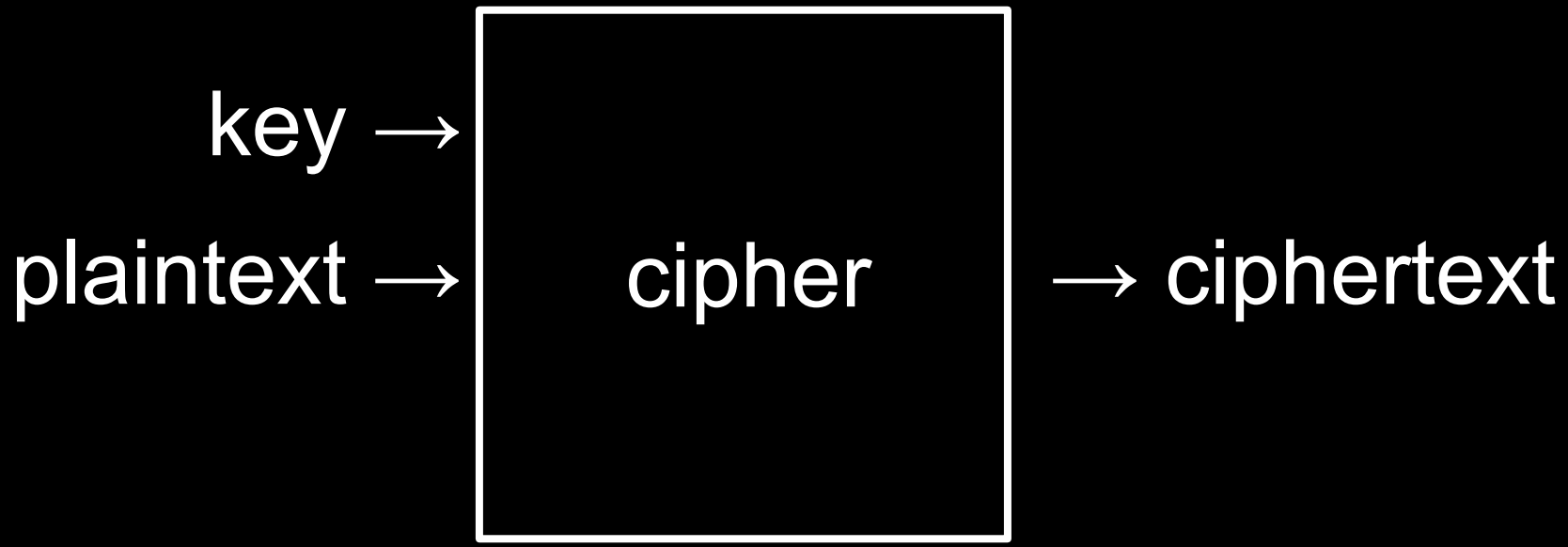
Grade 7

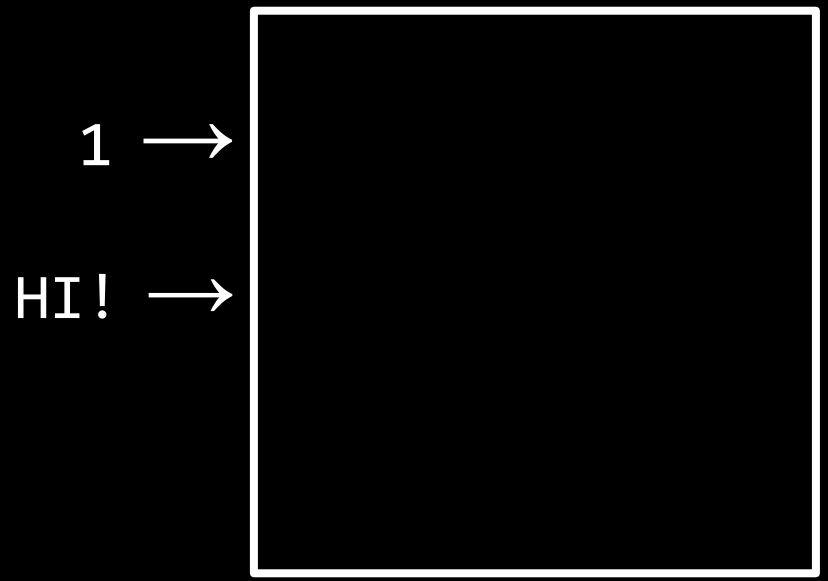
cryptography

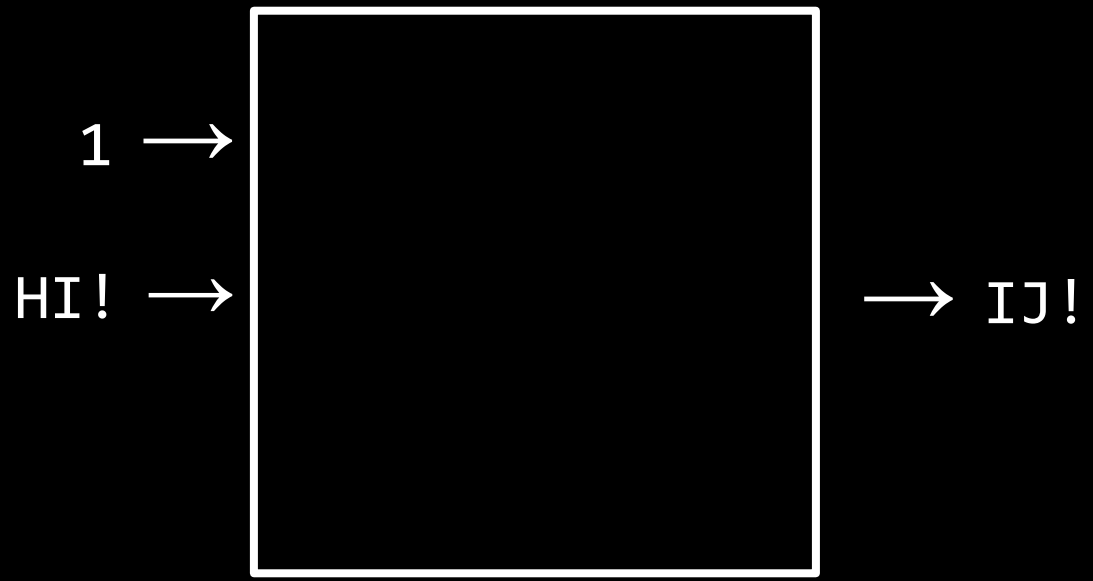

input →
→ output

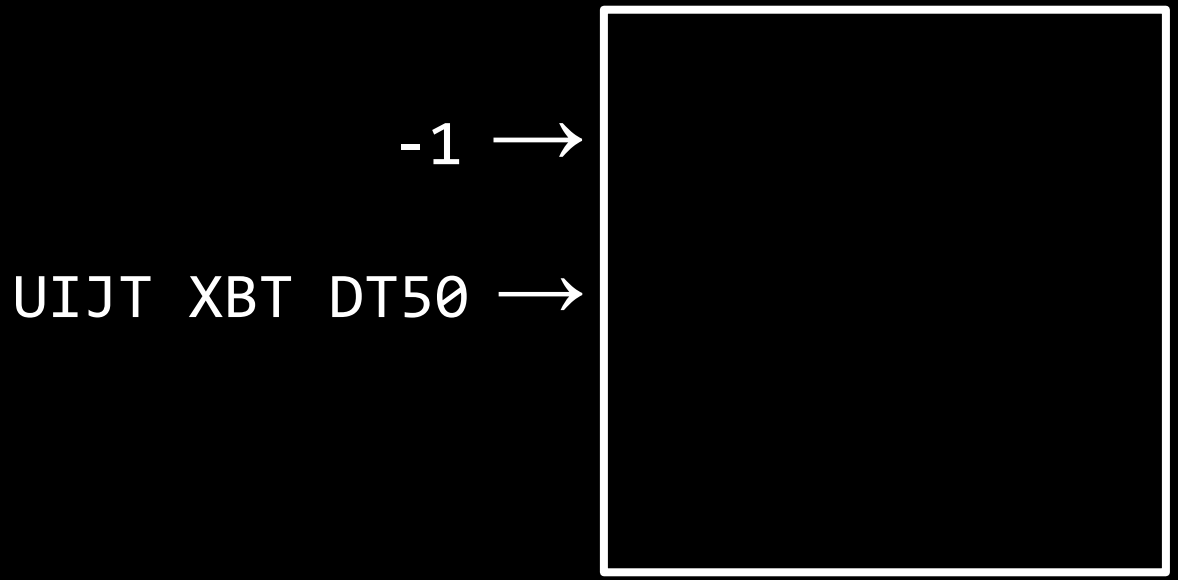

plaintext →
→ ciphertext


plaintext →
cipher
→ ciphertext


key →
plaintext →
cipher
→ ciphertext


1
HI!


1
HI!
IJ!


-1
UIJT XBT DT50

U I J T X B T D T 5 0

T I J T X B T D T 5 0

T H J T X B T D T 5 0

T H I T X B T D T 5 0

T H I S X B T D T 5 0

T H I S W B T D T 5 0

T H I S W A T D T 5 0

T H I S W A S D T 5 0

T H I S W A S C T 5 0

T H I S W A S C S 5 0