

Herzlich Willkommen bei Inf-Einf-B Block 4.

Heute geht es um Speicher.

Diese Konzepte fallen vielen anfangs schwer – Sie können auch die Tutorien nutzen, um Fragen zu klären.

Kurs-Website mit Wochenplan, Videos, Notizen, Aufgaben, FAQ, uvm.: [inf.zone](#)

Die Vorlesungen werden aufgezeichnet und im Internet veröffentlicht. Die Videos werden allerdings meist erst 3 Wochen später online sein; Folien und Notizen am Tag nach der Vorlesung. Machen Sie sich Notizen!

Ihre Fragen sind in der Aufzeichnung in der Regel nicht zu hören (ich wiederhole sie).

Es gibt eine Pause – und wir bekommen Besuch von Prof. Rickert (Robotik)

Dies ist Inf-Einf-B.

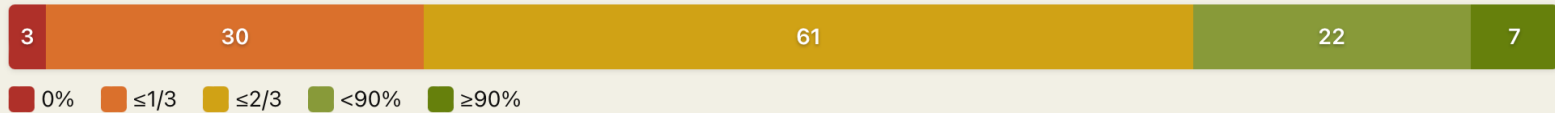
lecture-2 (Woche 2) · 104 Teilnehmende

Verteilung im Kurs:



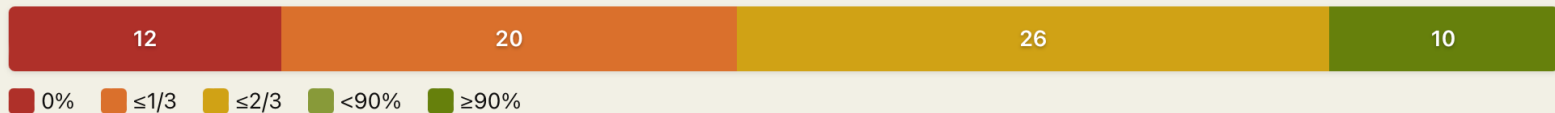
block25 (Woche 2.5) · 123 Teilnehmende

Verteilung im Kurs:



lecture-3 (Woche 3) · 68 Teilnehmende

Verteilung im Kurs:



Hochgeladene Übungsaufgaben

Woche 0
Scratch

88

Woche 1
C

130

Woche 2
Arrays

115

Woche 3
Algor.

64+

PROBLEM 1

Der folgende Code hat einen Programmierfehler.

Welchen?

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

int main(void)
{
    string word = get_string("Gib ein Wort ein: ");
    printf("Das Wort hat %i Zeichen\n", strlen(word));
    printf("Letztes Zeichen ist: %c\n", word[strlen(word)]);
}
```

PROBLEM 2

Hier fehlt etwas!
Ergänzen Sie den
Code, sodass er
kompilierbar ist.

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

int main(void)
{
    string names[3] = {"Alice", "Bob", "Charlie"};
    int scores[3] = {95, 87, 92};

    person students[3];

    for (int i = 0; i < 3; i++) {
        students[i].firstname = names[i];
        students[i].points = scores[i];
    }

    for (int i = 0; i < 3; i++) {
        printf("%s: %i points\n",
            students[i].firstname, students[i].points);
    }
}
```

**EMBRACE
THE FLOW**

INF.ZONE

**EMBRACE
THE FLOW**

INF.ZONE

**EMBRACE
THE FLOW**

INF.ZONE

INF.ZONE

**EMBRACE
THE FLOW**

INF.ZONE

**EMBRACE
THE FLOW**

INF.ZONE

**EMBRACE
THE FLOW**

INF.ZONE

**EMBRACE
THE FLOW**

INF.ZONE

**EMBRACE
THE FLOW**

**EMBRACE
THE FLOW**

THE FLOW

INF.ZONE

THE FLOW

INF.ZONE

INF.

INF.

ACE
LOW

NE

EMBRACE
THE FLOW

INF.ZONE

EMBRA
THE FL

INF.ZO

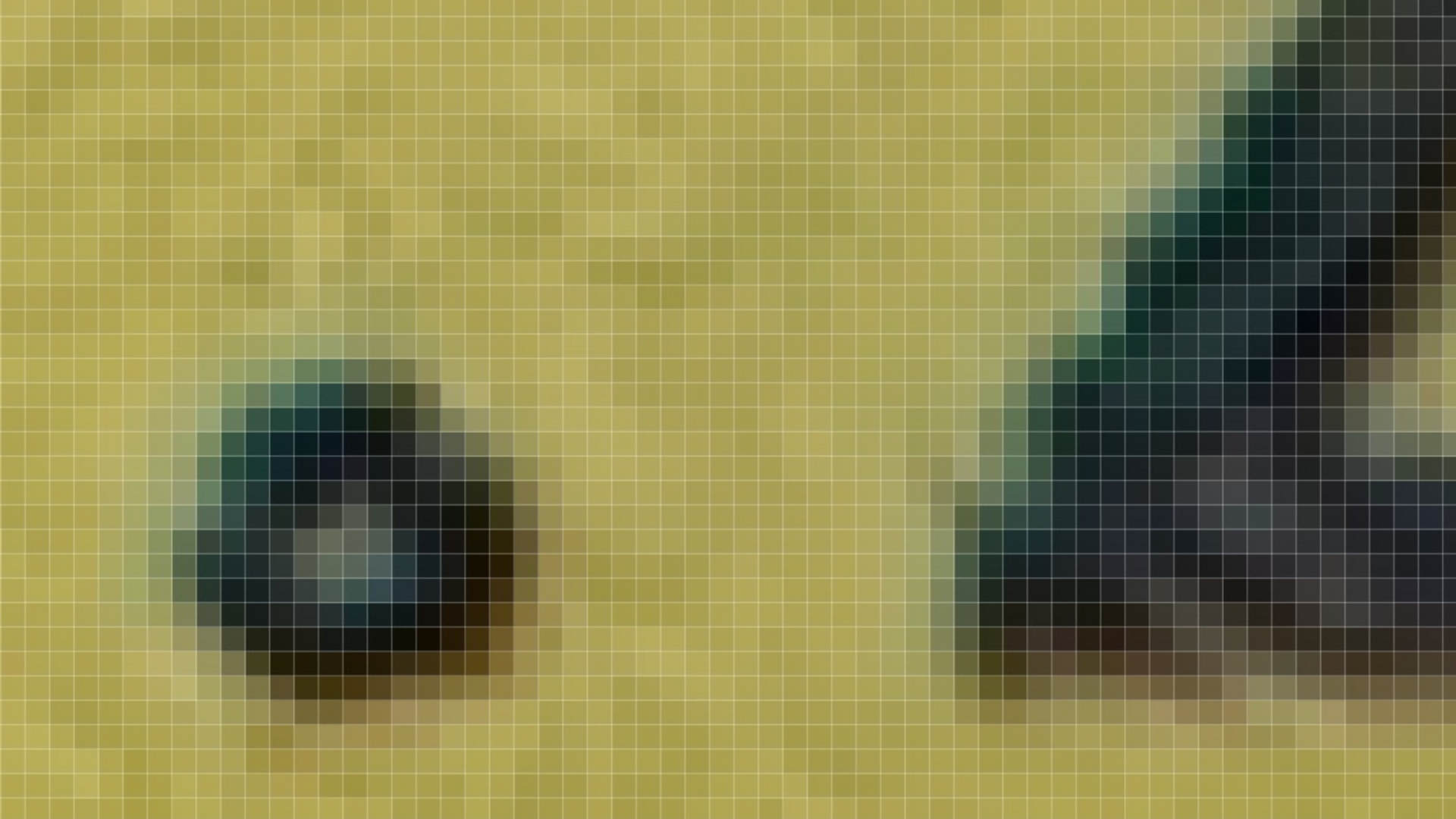
INF.ZONE

INF.ZONE

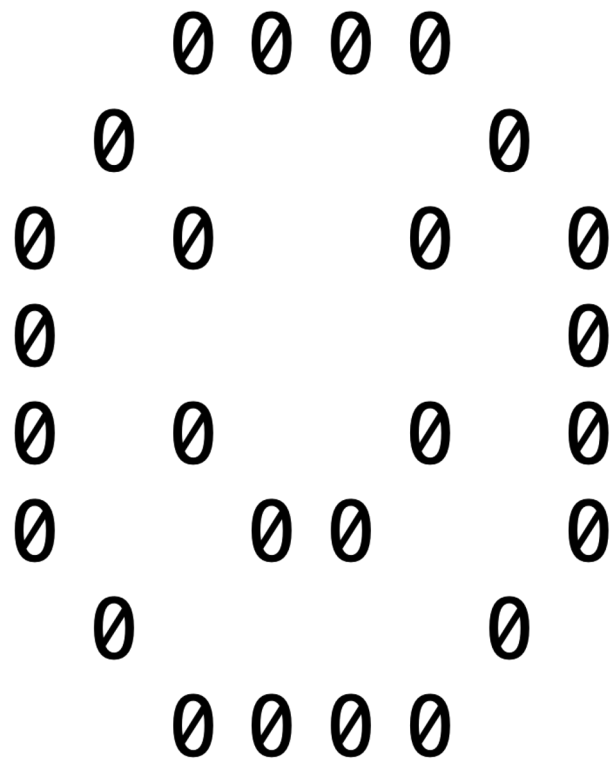
NF. ZON

F

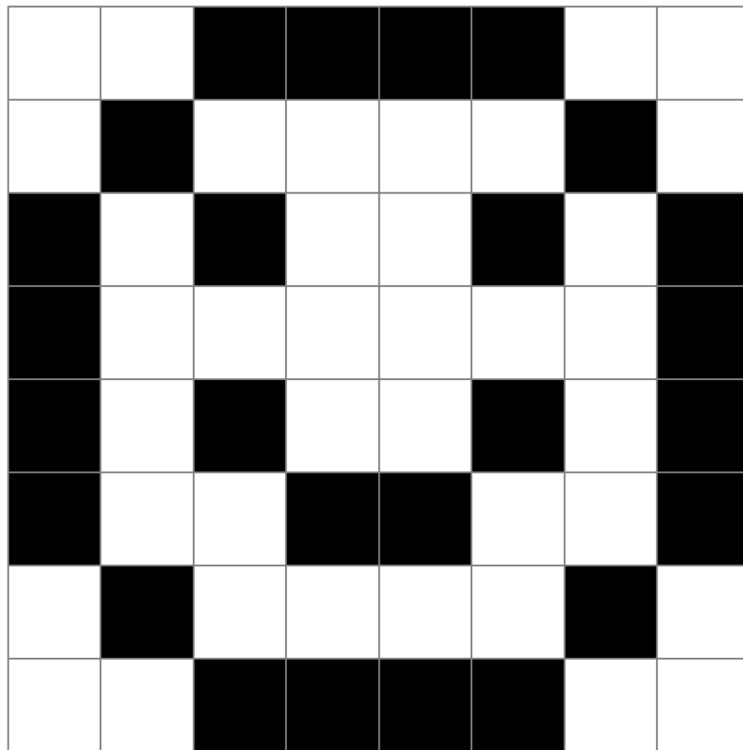
Z



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



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



```
int zahlen[5] = {10, 20, 30, 40, 50};
```

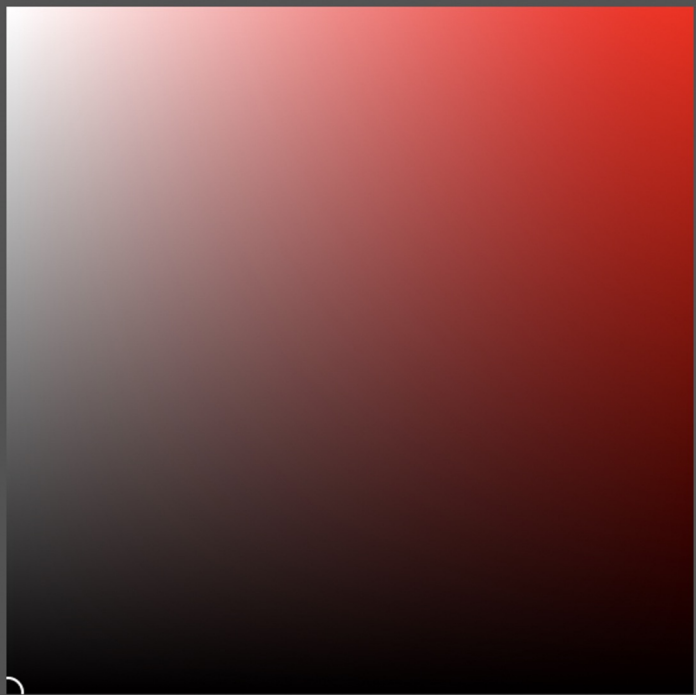
```
// Arrays müssen nicht unbedingt 1-dimensional sein:
```

```
int matrix[3][4] = {  
    {1, 2, 3, 4},  
    {5, 6, 7, 8},  
    {9, 10, 11, 12}  
};
```

```
// matrix[1][2] = 7 (Zeile 1, Spalte 2)
```

RGB

Color Picker (Foreground Color)



new



current

OK

Cancel

Add to Swatches

Color Libraries

☒ H: 0 °

☐ L: 0

☐ S: 0 %

☐ a: 0

☐ B: 0 %

☐ b: 0

☐ R: 0

C: 75 %

☐ G: 0

M: 68 %

☐ B: 0

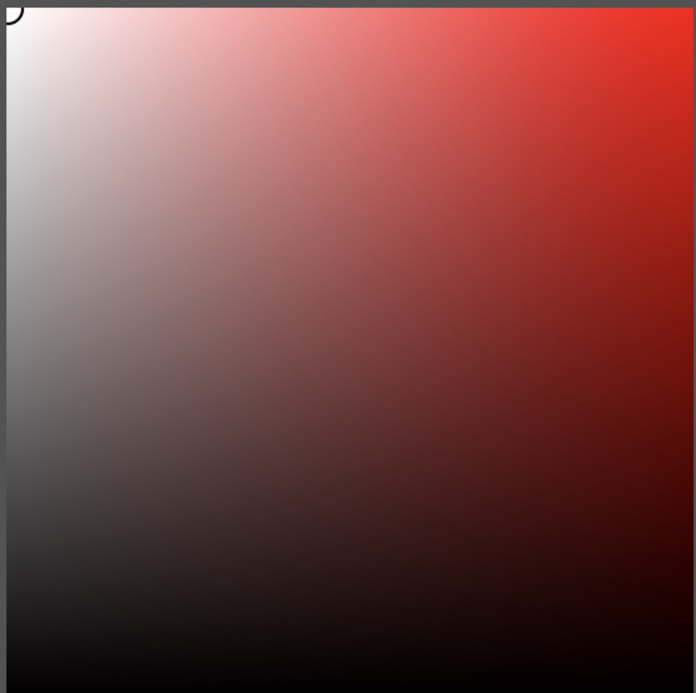
Y: 67 %

K: 90 %

☐ Only Web Colors

000000

Color Picker (Foreground Color)



new



current

OK

Cancel

Add to Swatches

Color Libraries

☐ Only Web Colors

☒ H: 0 °

☐ L: 100

☐ S: 0 %

☐ a: 0

☐ B: 100 %

☐ b: 0

☐ R: 255

C: 0 %

☐ G: 255

M: 0 %

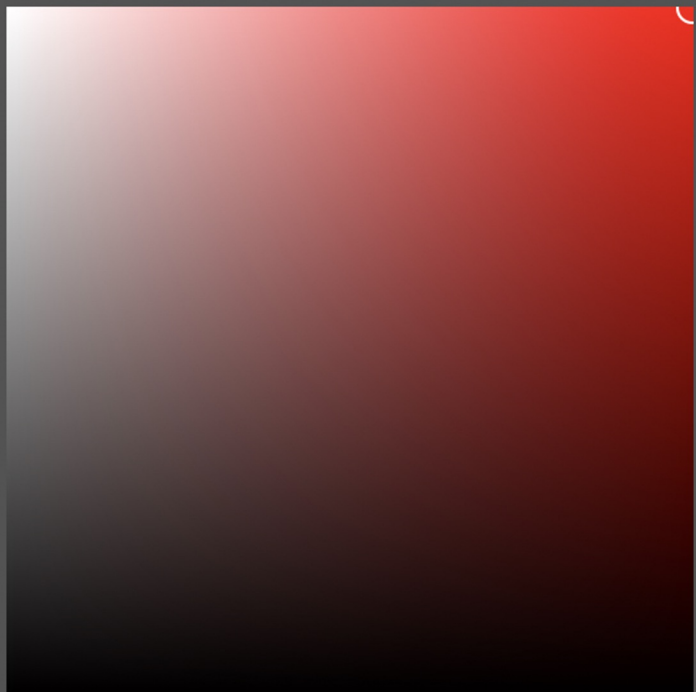
☐ B: 255

Y: 0 %

K: 0 %

FFFFFF

Color Picker (Foreground Color)



new



current

OK

Cancel

Add to Swatches

Color Libraries



H:

0

°



L:

54



S:

100

%



a:

81



B:

100

%



b:

70



R:

255

C:

0

%



G:

0

M:

99

%



B:

0

Y:

100

%

K:

0

%



Only Web Colors

#

FF0000

Color Picker (Foreground Color)



new



current

OK

Cancel

Add to Swatches

Color Libraries

☐ Only Web Colors

☒ H: 120 °

☐ L: 88

☐ S: 100 %

☐ a: -79

☐ B: 100 %

☐ b: 81

☐ R: 0

C: 63 %

☐ G: 255

M: 0 %

☐ B: 0

Y: 100 %

00FF00

K: 0 %

Color Picker (Foreground Color)



OK

Cancel

Add to Swatches

Color Libraries

☒ H: 240 °

☐ L: 30

☐ S: 100 %

☐ a: 68

☐ B: 100 %

☐ b: -112

☐ R: 0

C: 88 %

☐ G: 0

M: 77 %

☐ B: 255

Y: 0 %

K: 0 %

☐ Only Web Colors

0000FF

0 1

0 1 2 3 4 5 6 7 8 9

0 1 2 3 4 5 6 7 8 9 A B C D E F

0 1 2 3 4 5 6 7 8 9 a b c d e f

hexadezimal

“base-16”

16^1 16^0

#

16 1

#

16 1

00

16 1

01

16 1

02

16 1

03

16 1

04

16 1

05

16 1

06

16 1

07

16 1

08

16 1

09

16 1

0A

16 1

ØB

16 1

0C

16 1

ØD

16 1

0E

16 1

0F

16 1

10

16 1

11

16 1

12

16 1

13

16 1

14

16 1



16 1

FF

16 1

FF

$16 \times F + 1 \times F$

16 1

FF

$16 \times 15 + 1 \times 15$

16 1

FF

240 + 15

16 1

FF

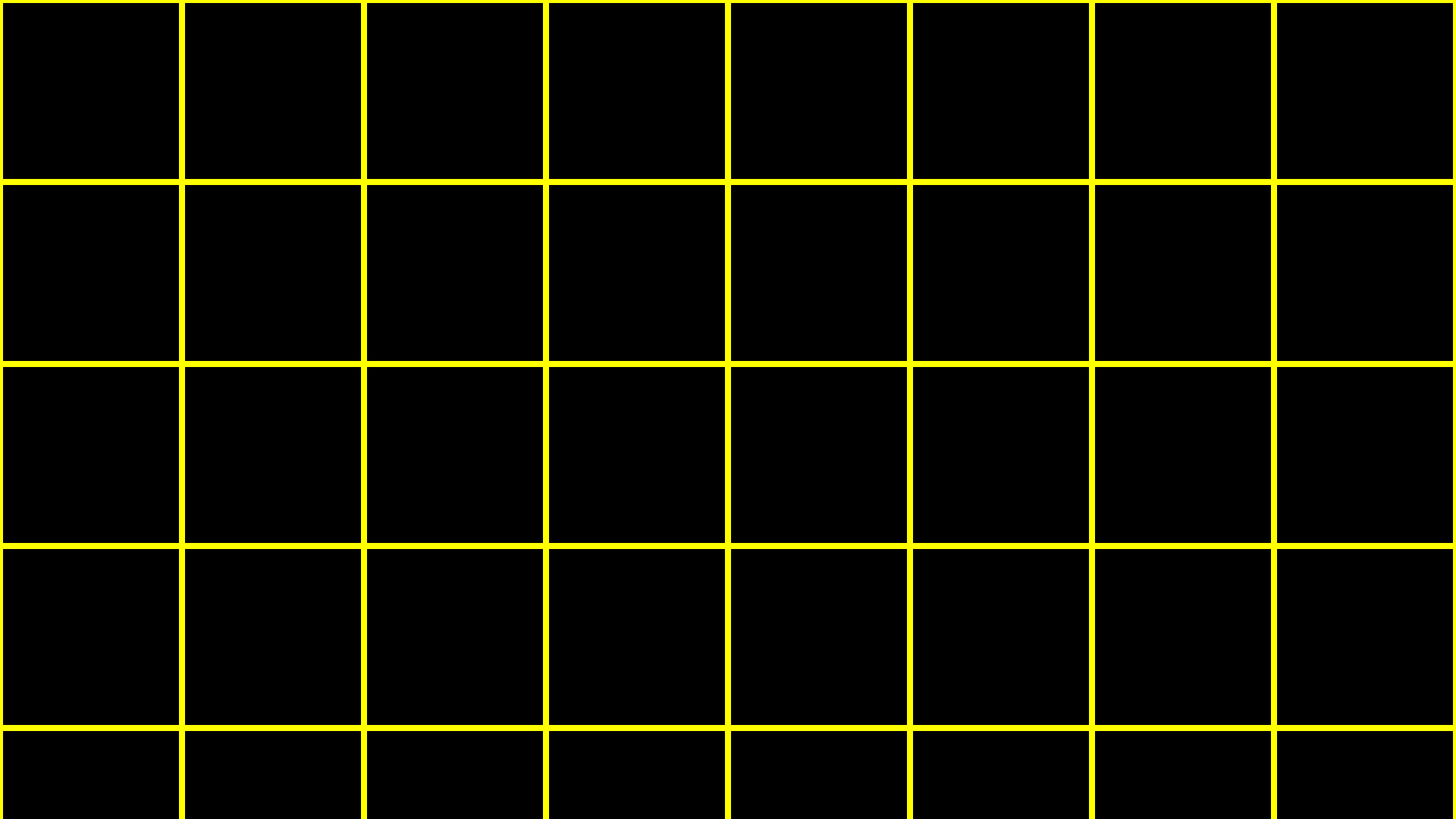
255

F

1111

11111111

FF



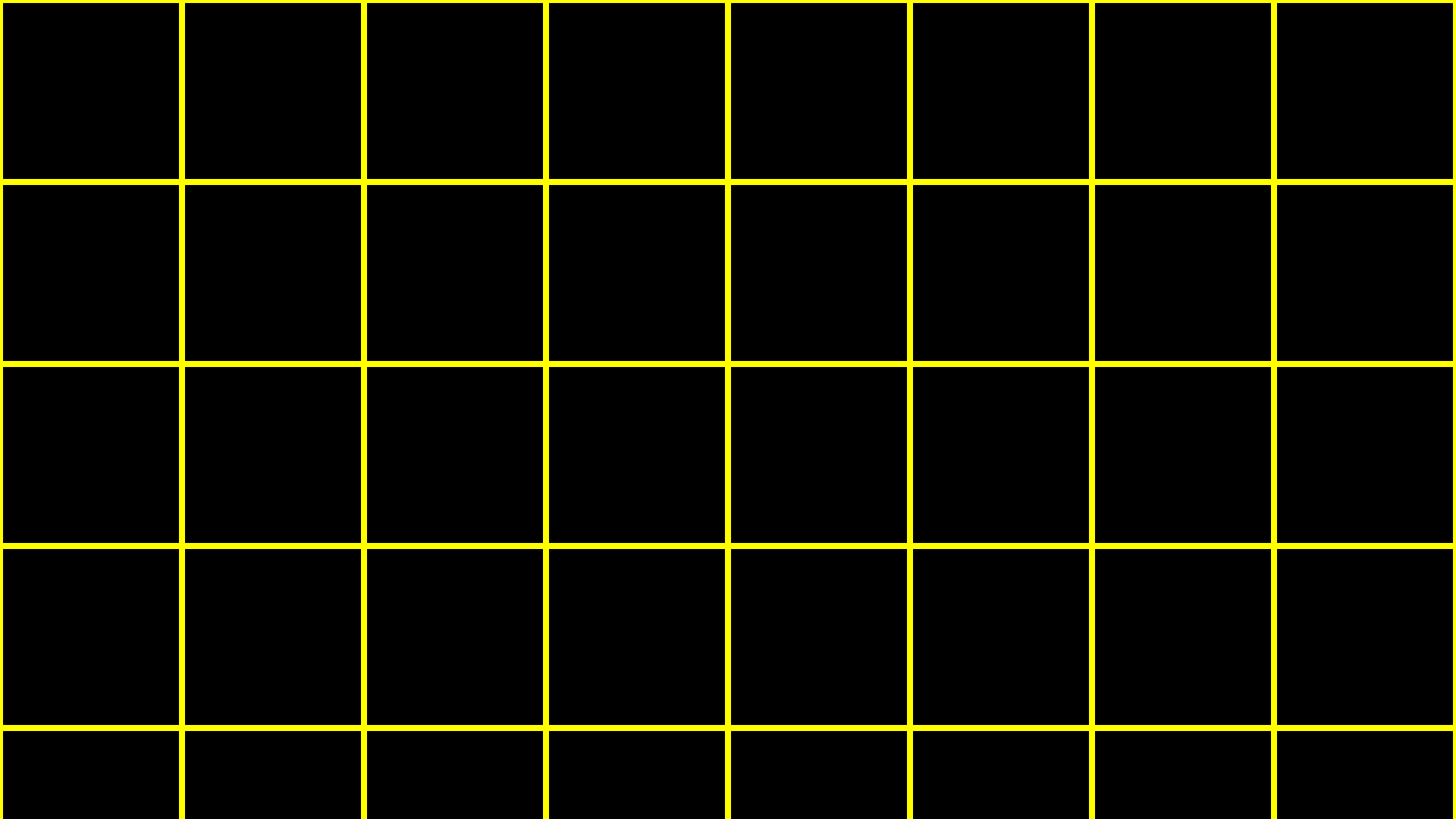
0	1	2	3	4	5	6	7
8	9	10	11	12	13	14	15

0	1	2	3	4	5	6	7
8	9	A	B	C	D	E	F

0	1	2	3	4	5	6	7
8	9	A	B	C	D	E	F
10	11	12	13	14	15	16	17
18	19	1A	1B	1C	1D	1E	1F

0x0	0x1	0x2	0x3	0x4	0x5	0x6	0x7
0x8	0x9	0xA	0xB	0xC	0xD	0xE	0xF
0x10	0x11	0x12	0x13	0x14	0x15	0x16	0x17
0x18	0x19	0x1A	0x1B	0x1C	0x1D	0x1E	0x1F

```
int n = 50;
```



				50			
						n	

				50			
				0x123			

&

*

Pointer

```
int n = 50;
```

```
int *p = &n;
```

```
int n = 50;
```

```
int *p = &n;
```

```
int n = 50;
```

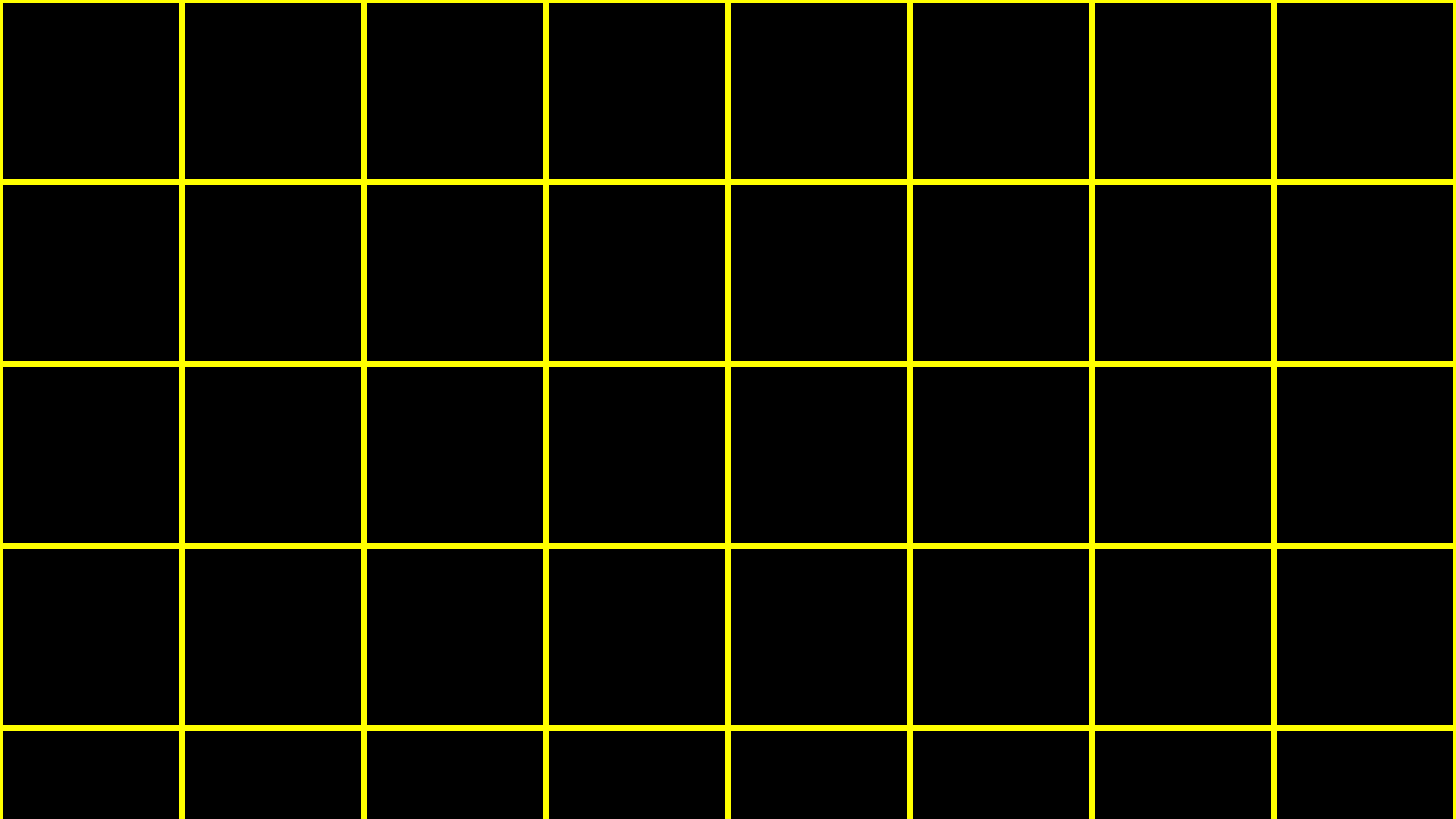
```
int* p = &n;
```

```
int n = 50;
```

```
int * p = &n;
```

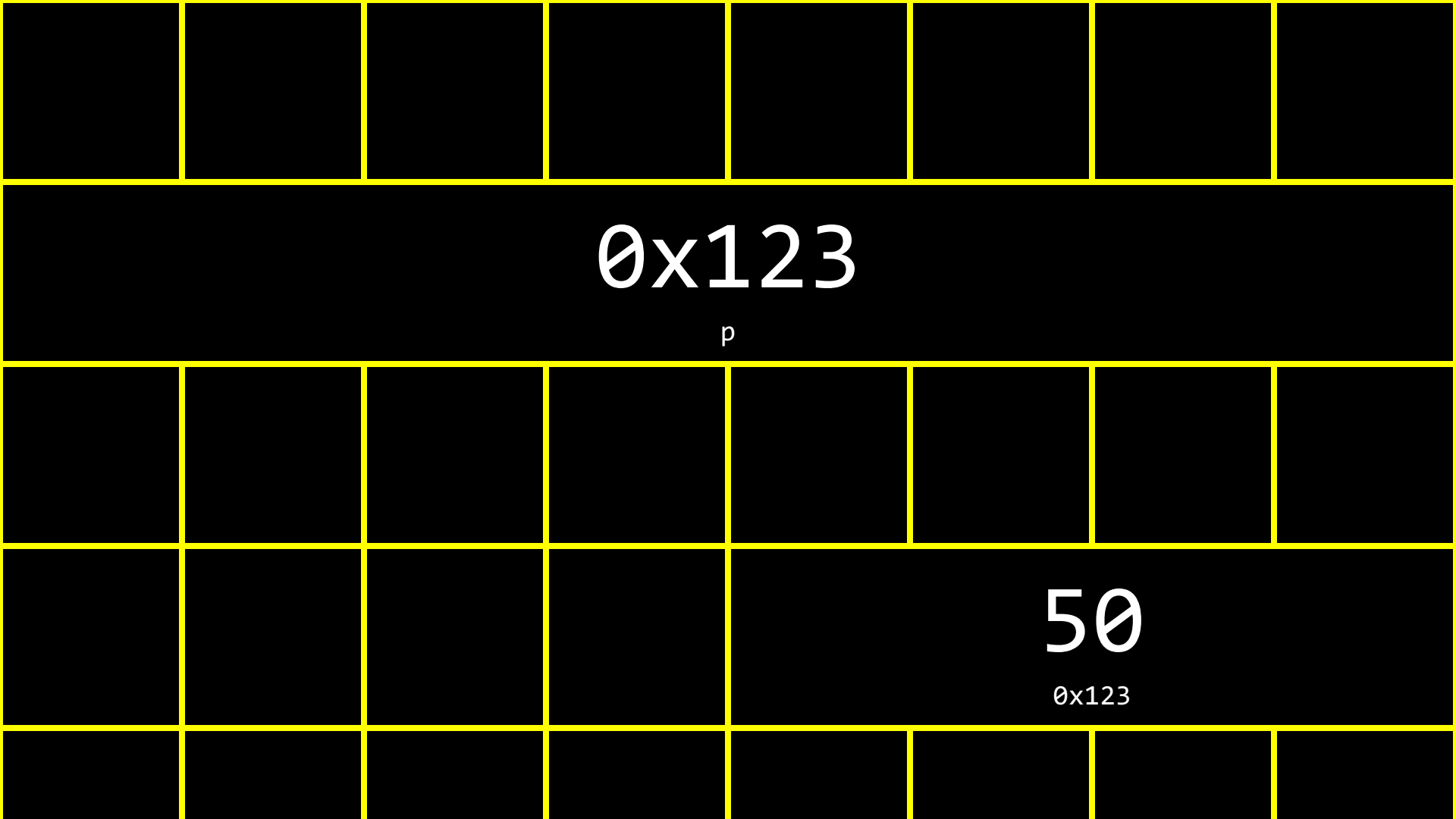
```
int n = 50;
```

```
int *p = &n;
```



				50			
						n	

				50			
				0x123			



0x123

p

50

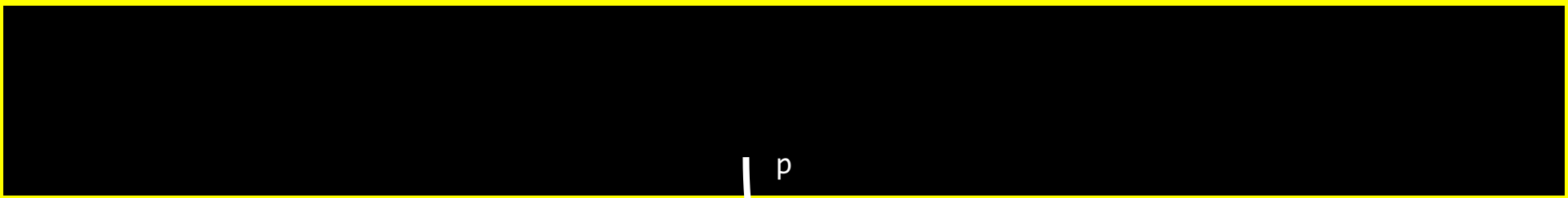
0x123

0x123

p

50

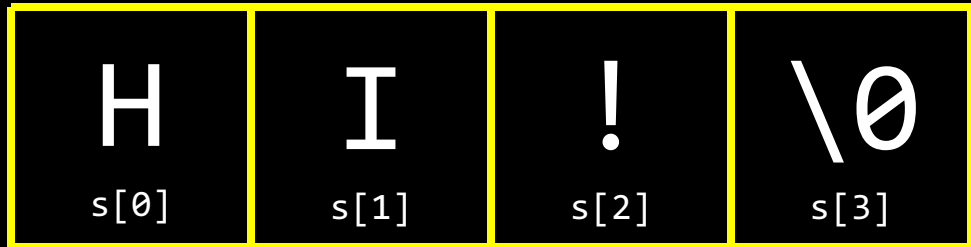
0x123



string

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

H	I	!	\0
---	---	---	----



H 0x123	I 0x124	! 0x125	\0 0x126
------------	------------	------------	-------------

s

H

0x123

I

0x124

!

0x125

\0

0x126

0x123

s

H

0x123

I

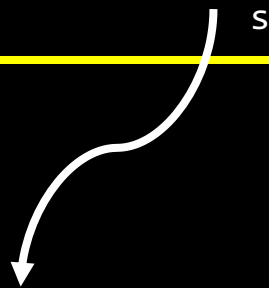
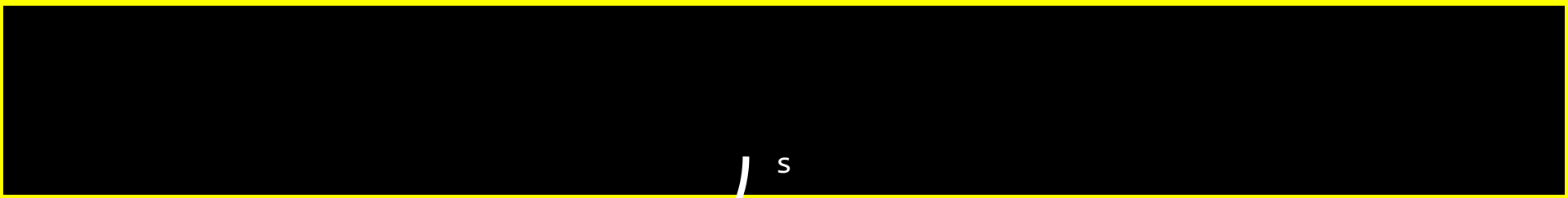
0x124

!

0x125

\0

0x126



s

H	I	!	\0
---	---	---	----

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

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

```
char *s = "HI!";
```

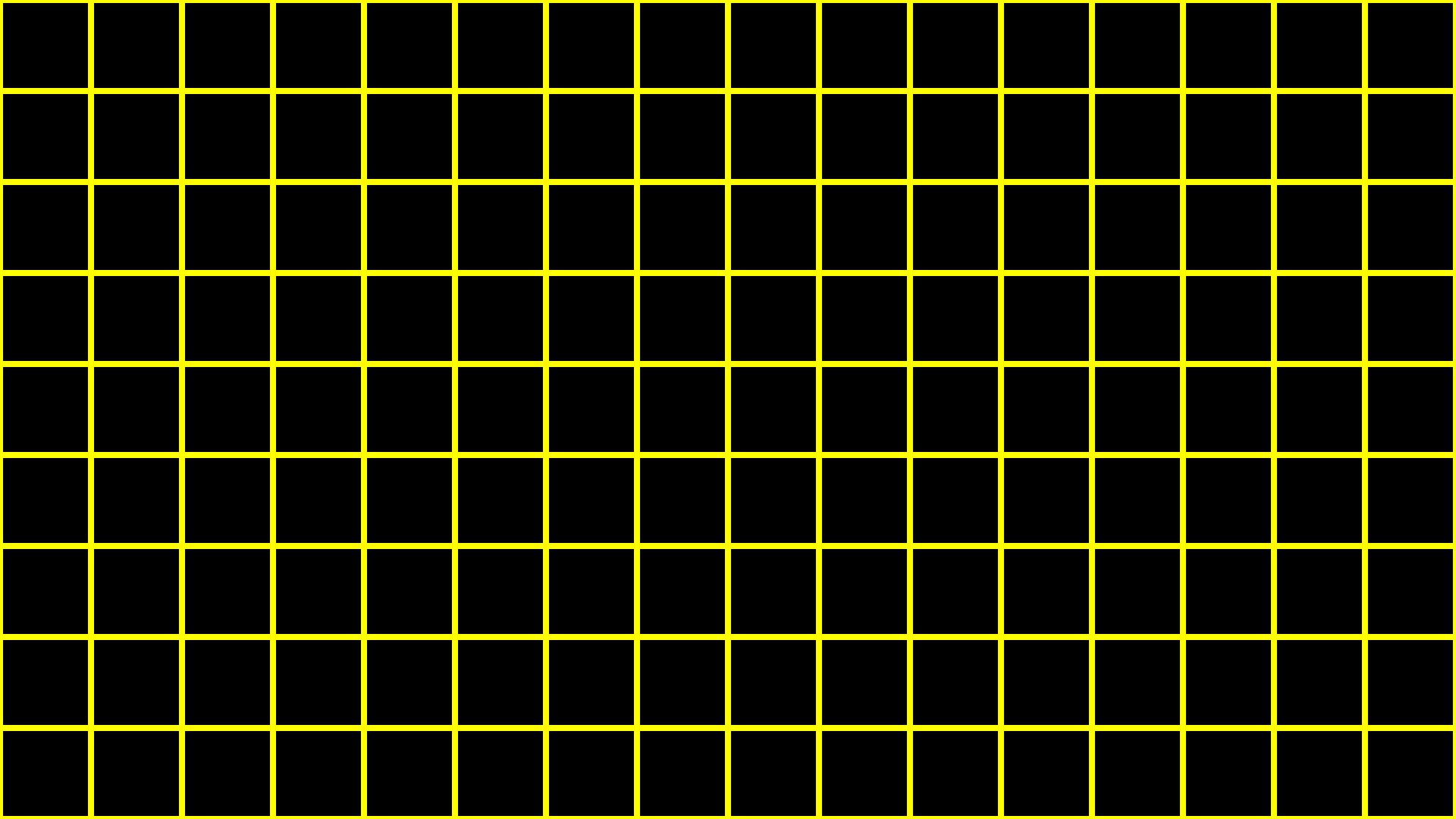
```
typedef struct  
{  
    string name;  
    string number;  
} person;
```

```
typedef int integer;
```

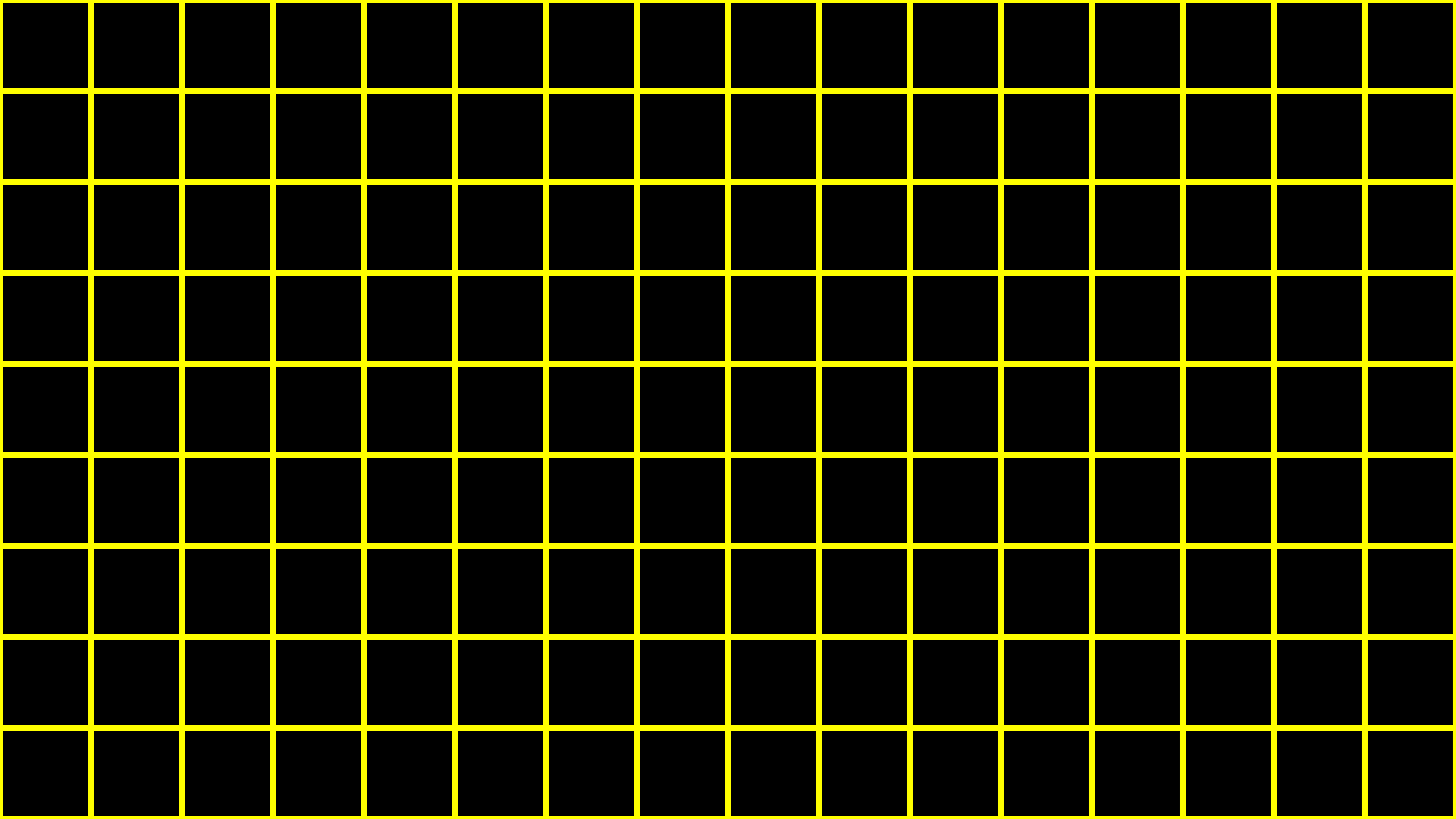
```
typedef uint8_t BYTE;
```

```
typedef char *string;
```

Pointer-Arithmetik



[illegible]



[illegible]

S

		H	I	!	\0		
--	--	---	---	---	----	--	--

		H	I	!	\0		
--	--	---	---	---	----	--	--

		H	I	!	\0		
--	--	---	---	---	----	--	--

		H	I	!	\0		
--	--	---	---	---	----	--	--

S

H

0x123

I

0x124

!

0x125

 $\backslash \theta$

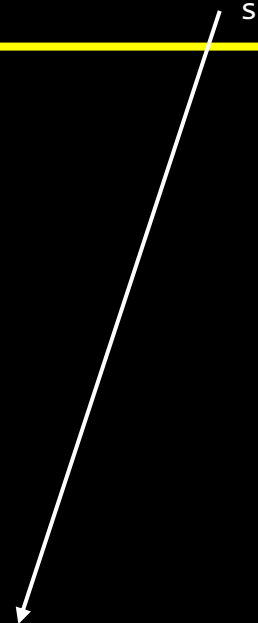
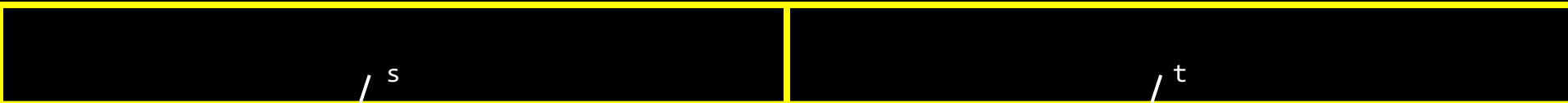
0x126

[illegible]

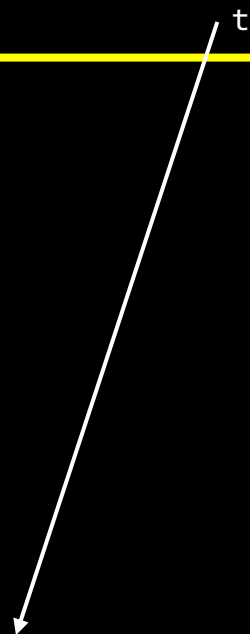
[illegible]

[illegible]

[illegible]



H	I	!	\0
0x123	0x124	0x125	0x126



H	I	!	\0
0x456	0x457	0x458	0x459

PROBLEM 3

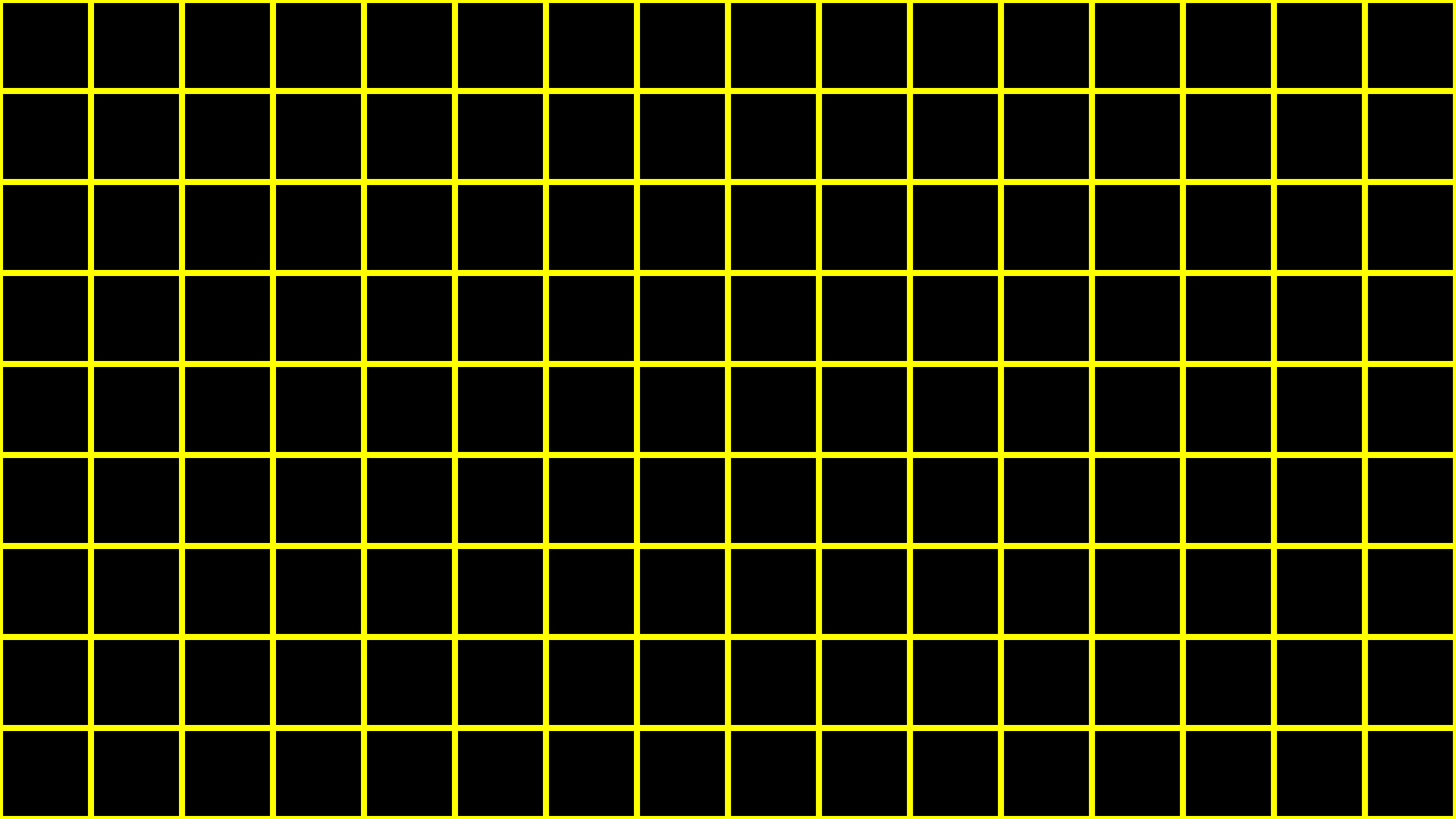
Parsons-Problem:

Bringen Sie die Zeilen in
die richtige Reihenfolge.
Gewünscht ist
folgende Ausgabe

42
100
100

- A. `printf("%i\n", n);`
- B. `int *p = &n;`
- C. `*p = 100;`
- D. `int n = 42;`
- E. `printf("%i\n", *p);`
- F. `printf("%i\n", n);`
- G. `printf("%i\n", p);`

Was man beim
Kopieren von Strings
beachten muss



[illegible]

S

h

i

!

 $\backslash \theta$

S

i 0x124	! 0x125
------------	------------

!	\0
0x125	0x126

	\0	
	0x126	

--	--	--

0x123
s

0x123
s

i 0x124	! 0x125
------------	------------

i 0x124	! 0x125
------------	------------

!	\0
0x125	0x126

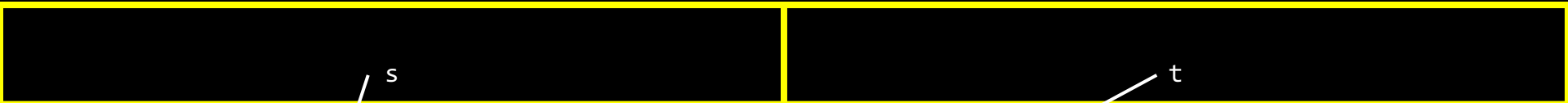
!	\0
0x125	0x126

	\0	
	0x126	

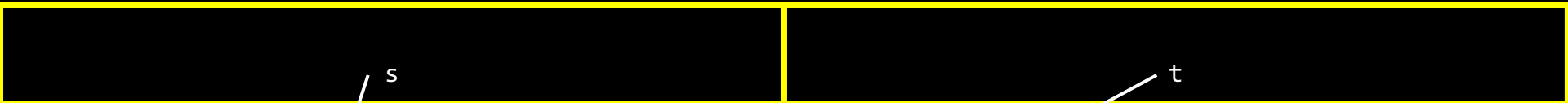
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	1	2	3	4	5</																										

[illegible]

[illegible]



h	i	!	\0
0x123	0x124	0x125	0x126



H	i	!	\0
0x123	0x124	0x125	0x126

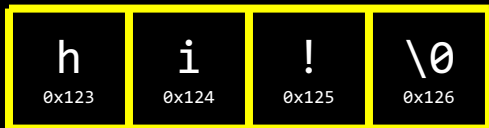
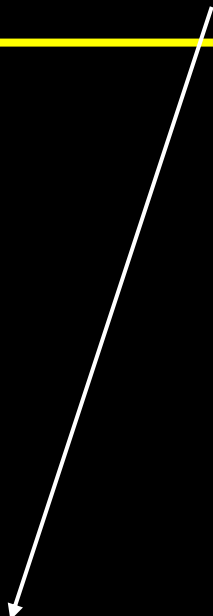
malloc

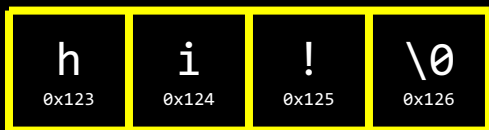
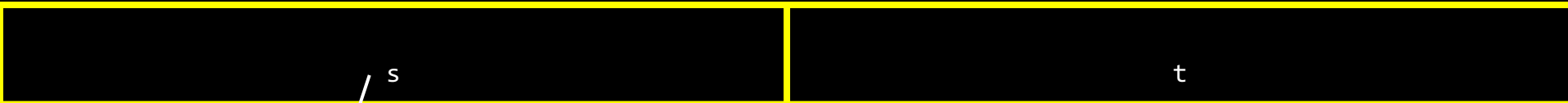
free

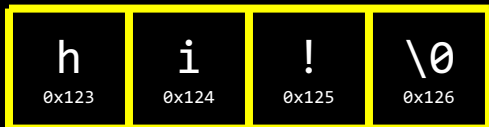
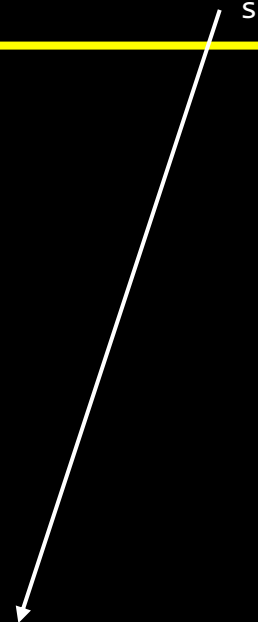
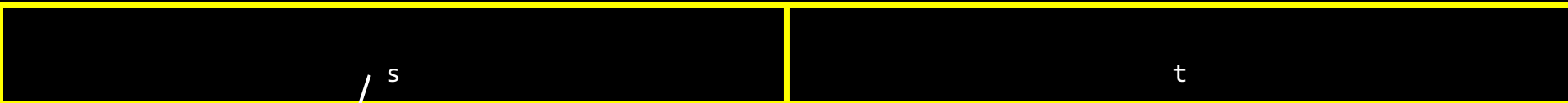
...

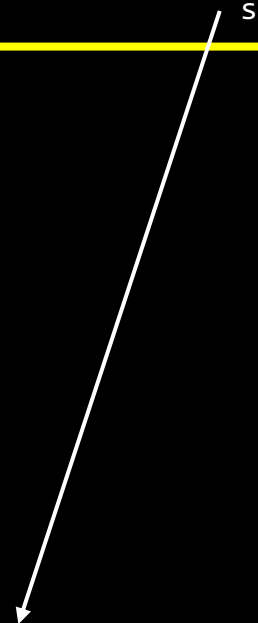
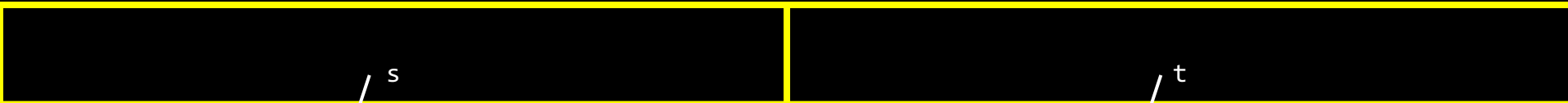


s

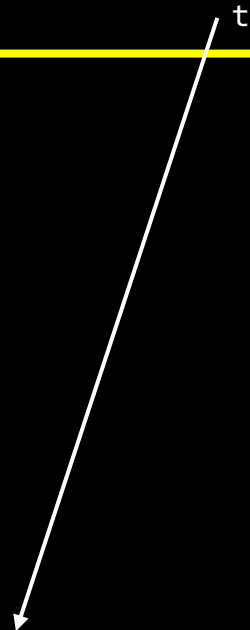




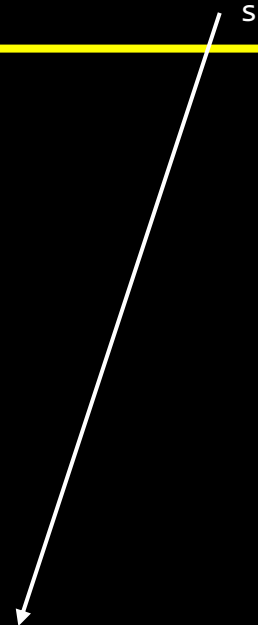
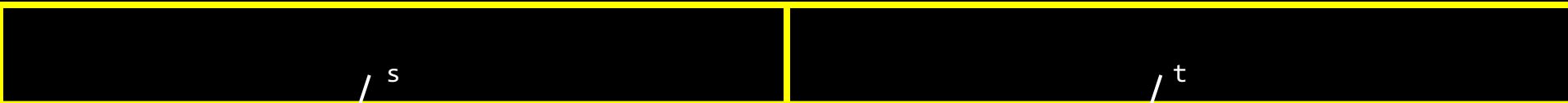




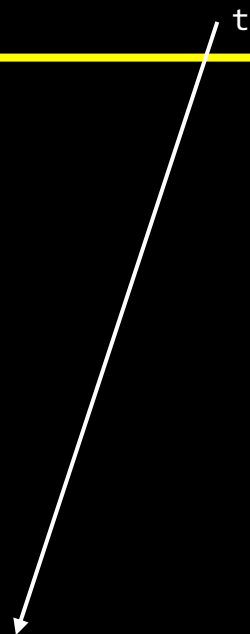
h	i	!	\0
0x123	0x124	0x125	0x126



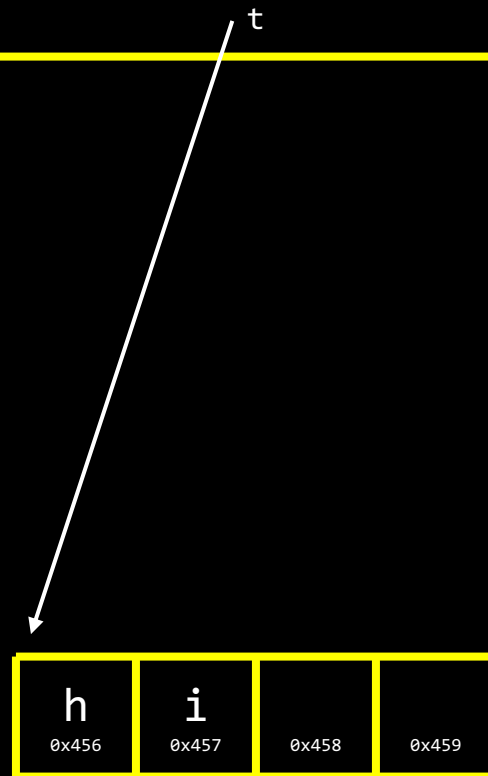
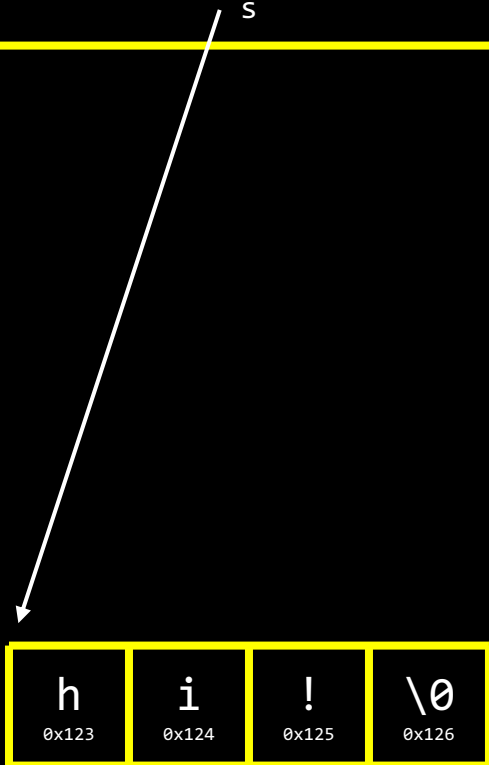
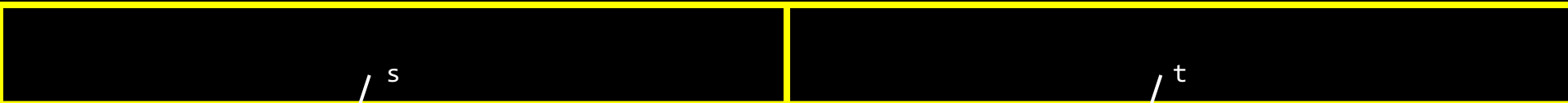
0x456	0x457	0x458	0x459

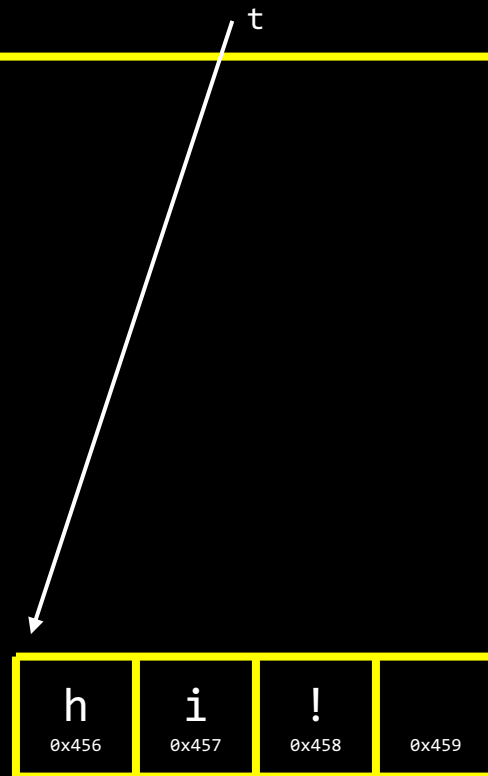
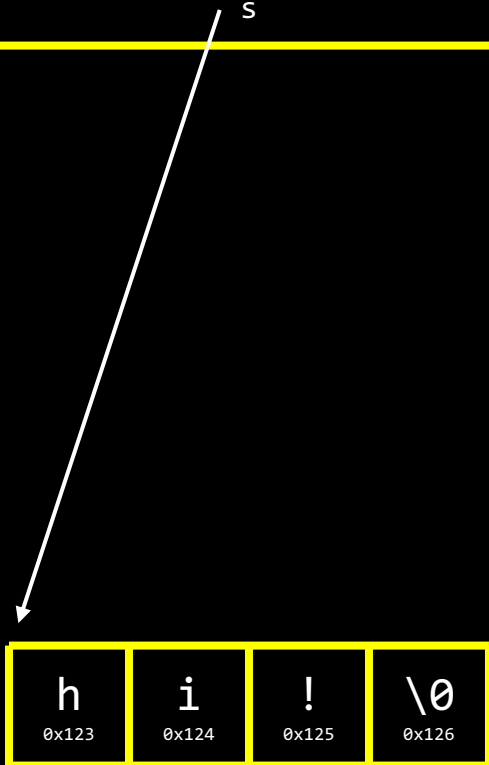
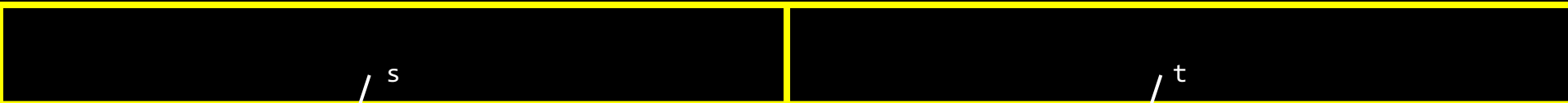


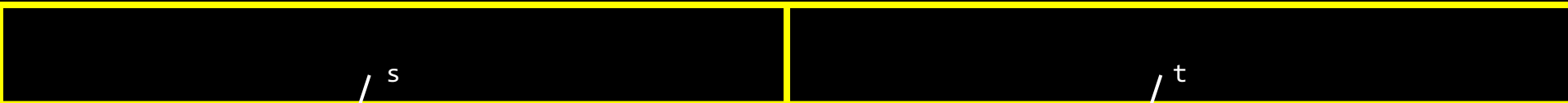
h	i	!	\0
0x123	0x124	0x125	0x126



h			
0x456	0x457	0x458	0x459

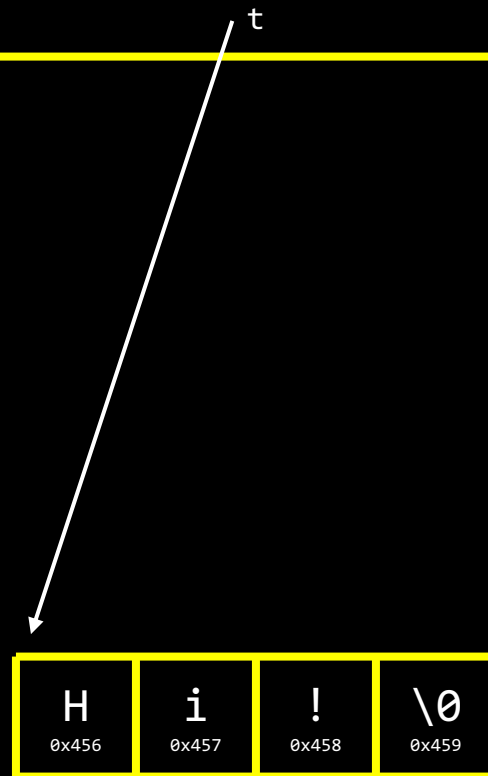
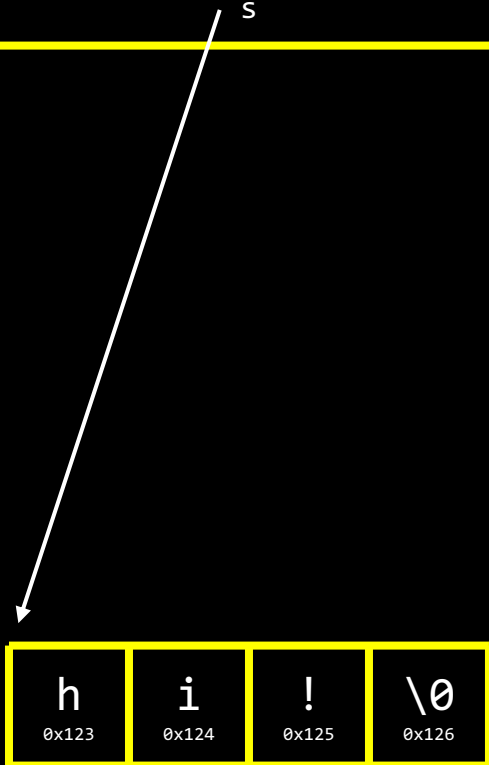
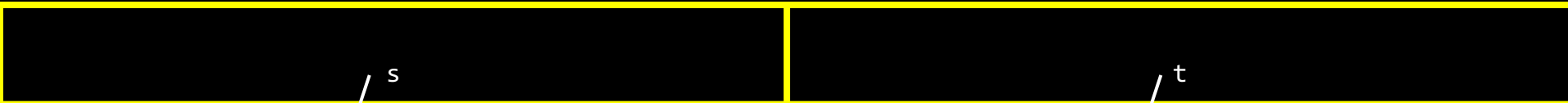






h	i	!	\0
0x123	0x124	0x125	0x126

h	i	!	\0
0x456	0x457	0x458	0x459



NULL

valgrind

Crowdstrike 2024 Buffer Overread

By Smishra1 - Own work, CC BY-SA 4.0,
<https://commons.wikimedia.org/w/index.php?curid=150535443>



Garbage Values

```
int main(void)
{
    int *x;
    int *y;

    x = malloc(sizeof(int));

    *x = 42;
    *y = 13;

    y = x;

    *y = 13;
}
```

```
int main(void)
{
    int *x;
    int *y;

    x = malloc(sizeof(int));

    *x = 42;
    *y = 13;

    y = x;

    *y = 13;
}
```

```
int main(void)
{
    int *x;
    int *y;

    x = malloc(sizeof(int));

    *x = 42;
    *y = 13;

    y = x;

    *y = 13;
}
```

```
int main(void)
{
    int *x;
    int *y;

    x = malloc(sizeof(int));

    *x = 42;
    *y = 13;

    y = x;

    *y = 13;
}
```

```
int main(void)
{
    int *x;
    int *y;

    x = malloc(sizeof(int));

    *x = 42;
    *y = 13;

    y = x;

    *y = 13;
}
```

```
int main(void)
{
    int *x;
    int *y;

    x = malloc(sizeof(int));

    *x = 42;

    y = x;

    *y = 13;
}
```

```
int main(void)
{
    int *x;
    int *y;

    x = malloc(sizeof(int));

    *x = 42;

    y = x;

    *y = 13;
}
```

```
int main(void)
{
    int *x;
    int *y;

    x = malloc(sizeof(int));

    *x = 42;

    y = x;

    *y = 13;
}
```

```
void swap(int a, int b)
```

```
{
```

```
}
```

```
void swap(int a, int b)
{
    int tmp = a;
    a = b;
    b = tmp;
}
```


scope

passing by value

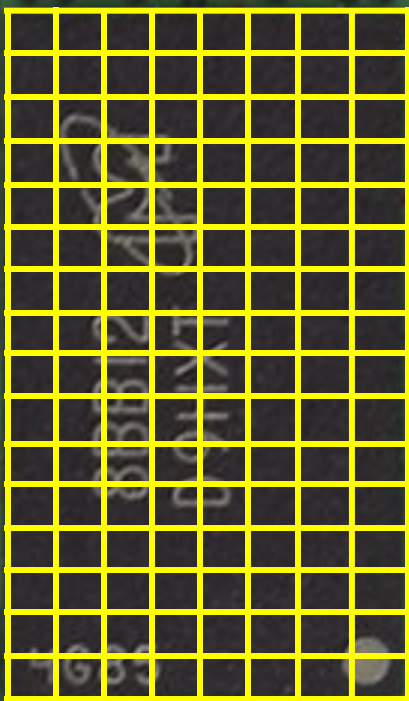
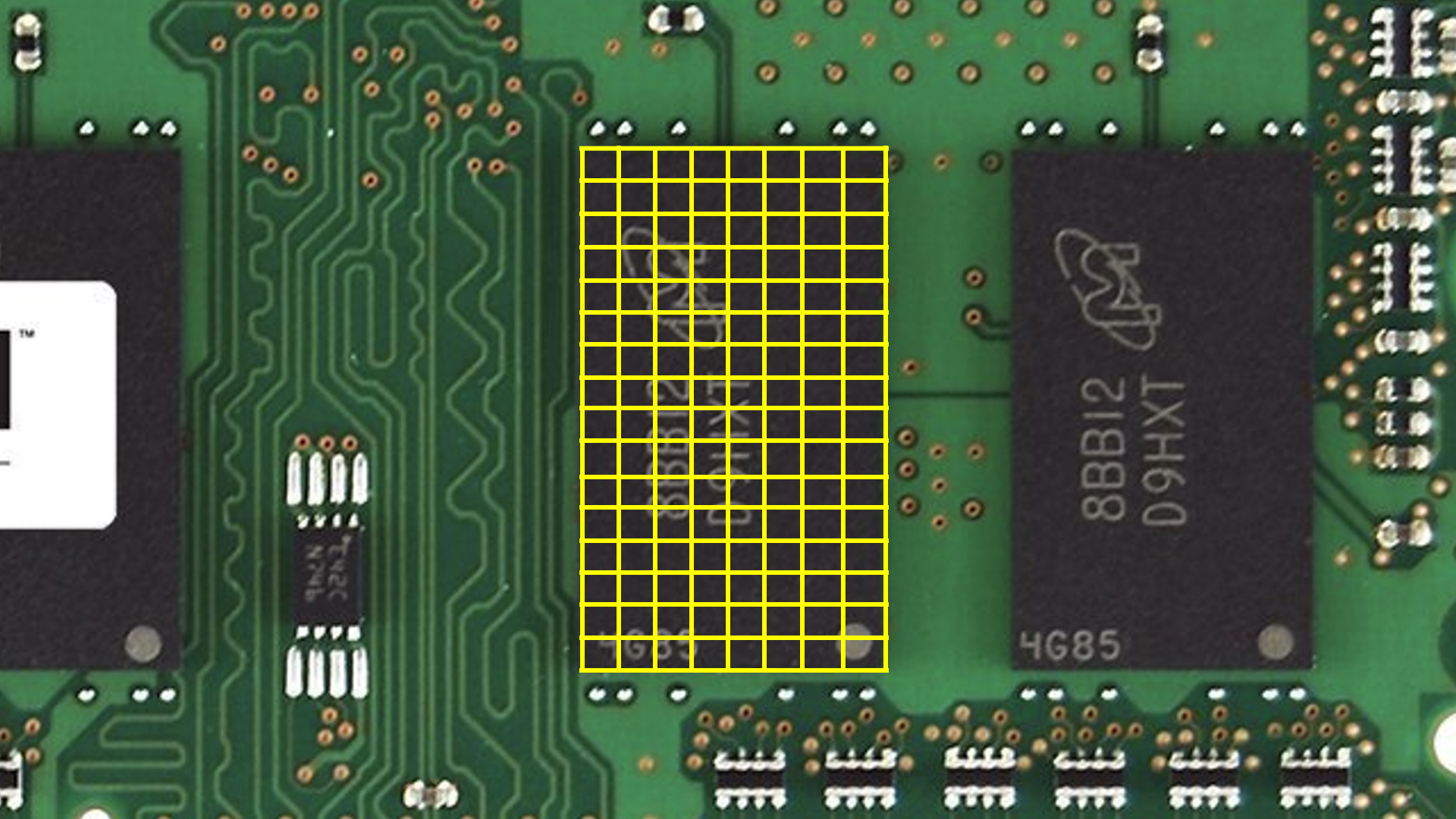


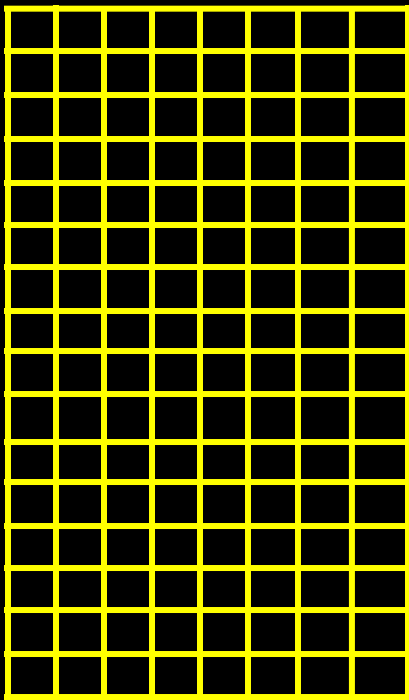
™

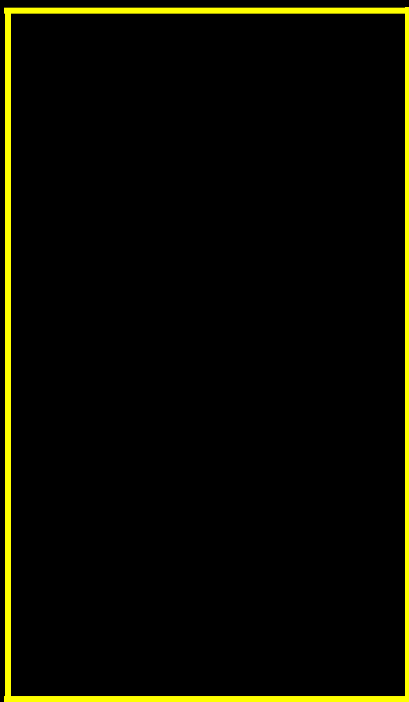
9442N
2242


8BB12
D9HXT
4G85


8BB12
D9HXT
4G85







machine code

machine code

globals

machine code

globals

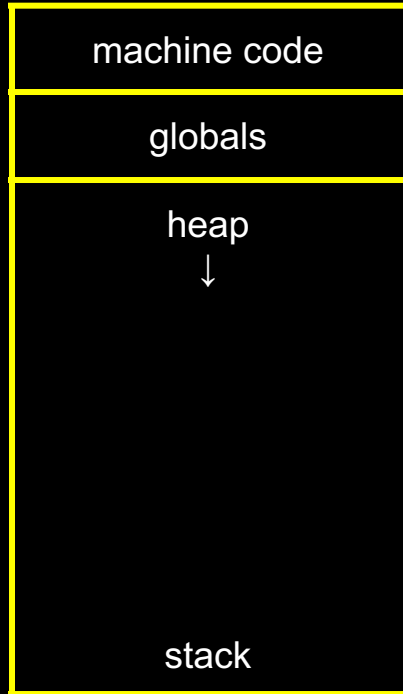
heap

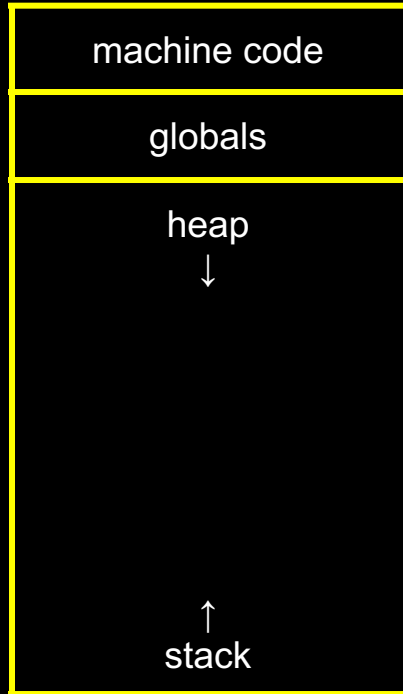
machine code

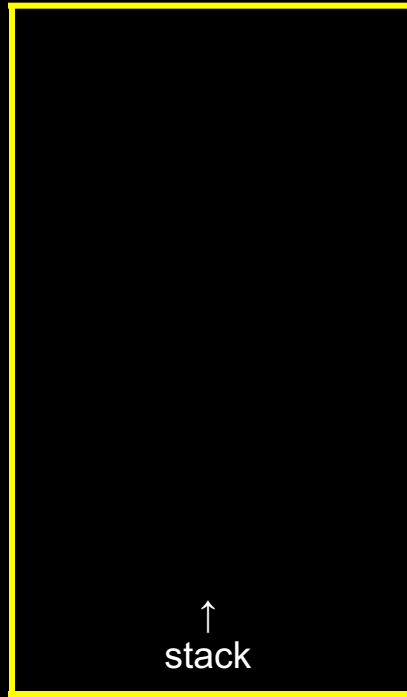
globals

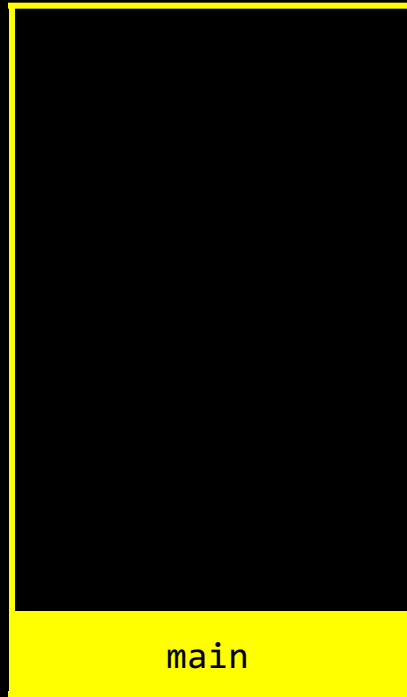
heap

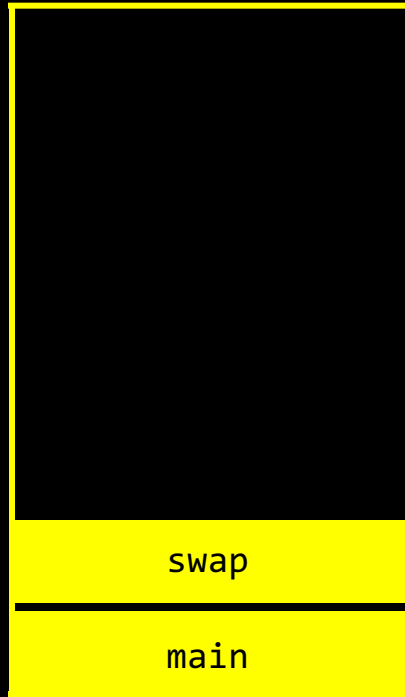


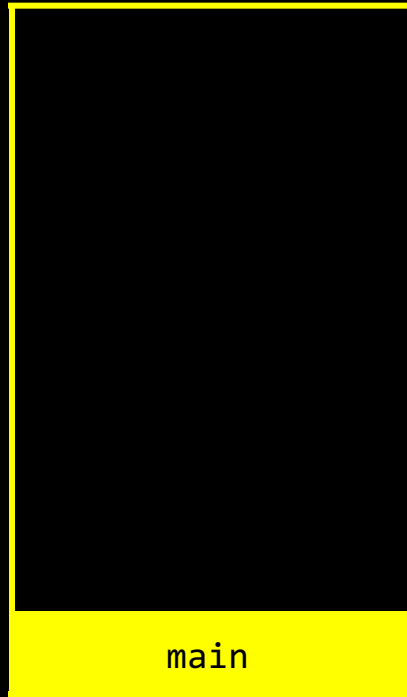


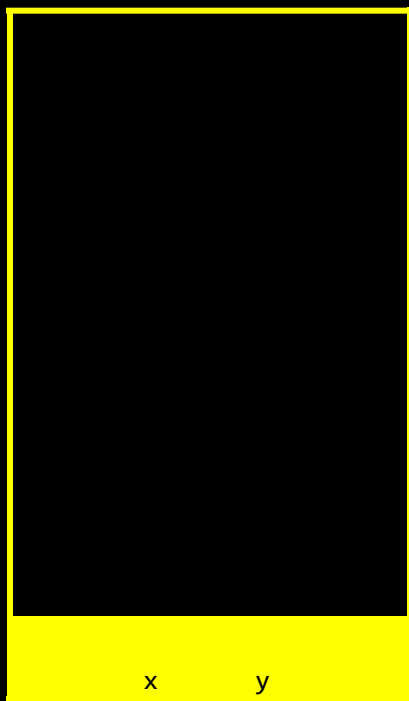


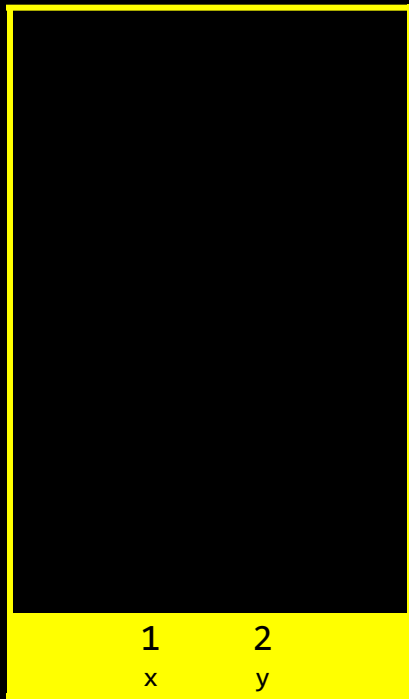


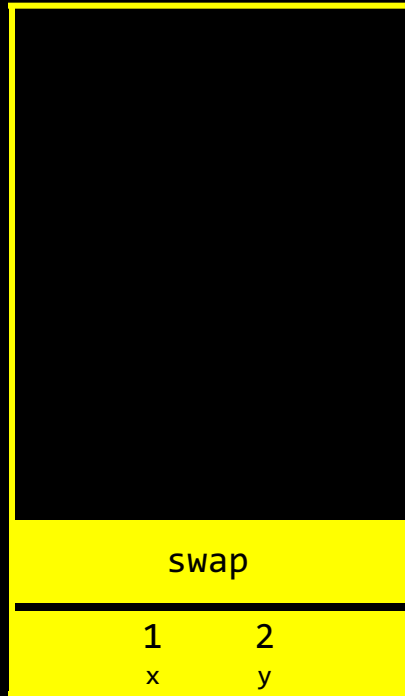




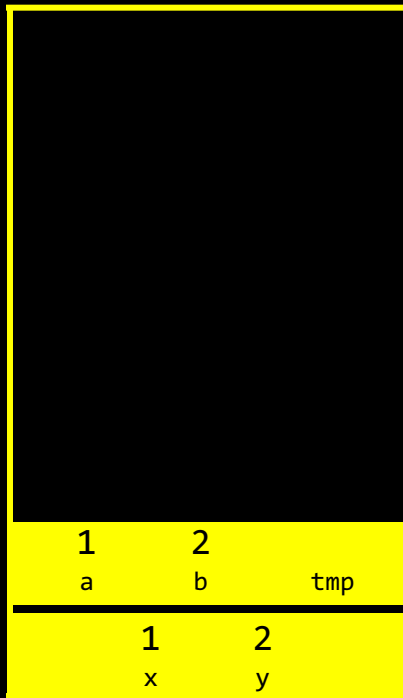




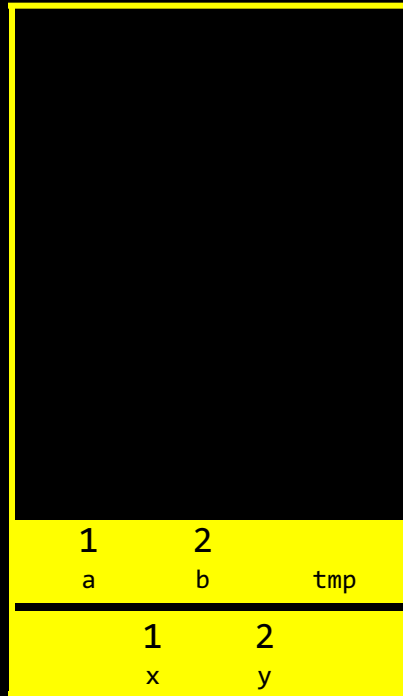




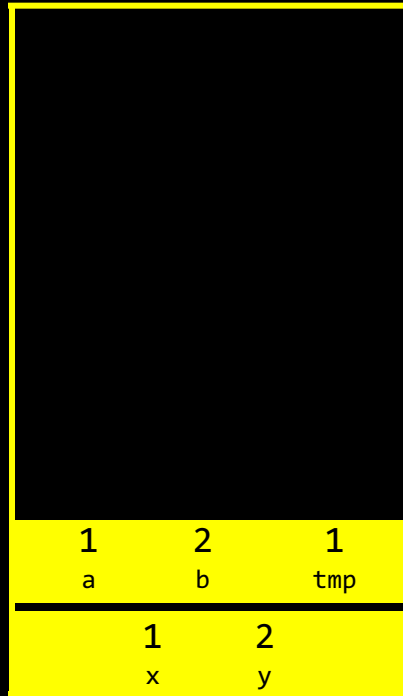
a	b	tmp
1	2	
x	y	



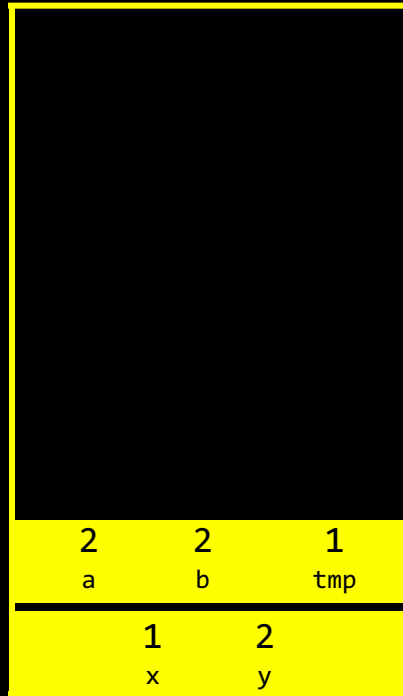
```
int tmp = a;  
a = b;  
b = tmp;
```



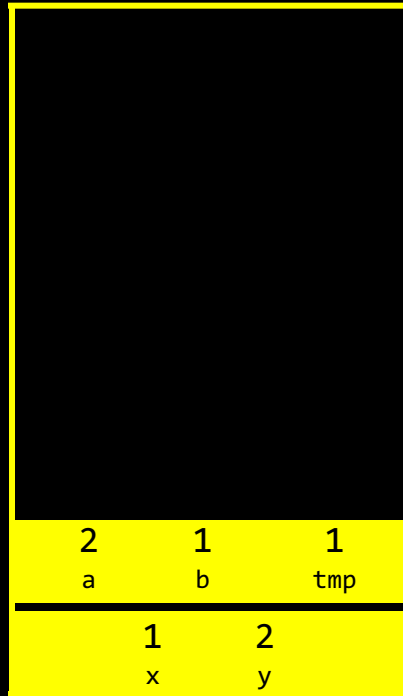
```
int tmp = a;  
a = b;  
b = tmp;
```

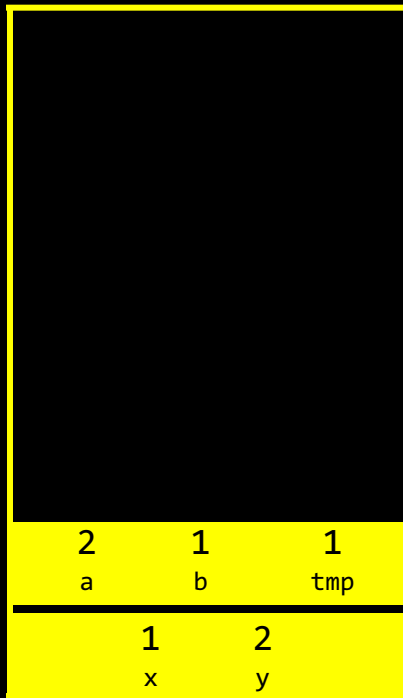


```
int tmp = a;  
a = b;  
b = tmp;
```



```
int tmp = a;  
a = b;  
b = tmp;
```



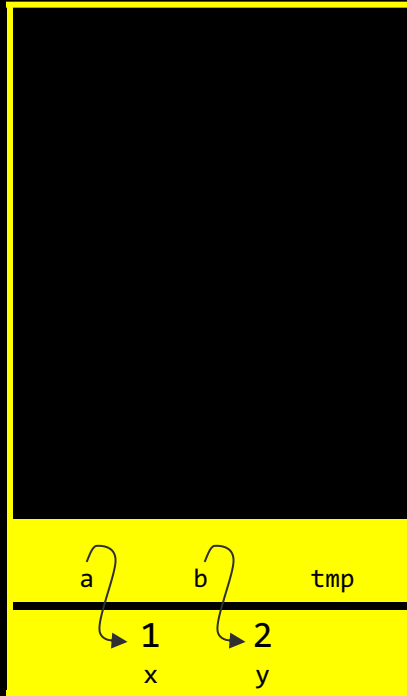


1	2
x	y

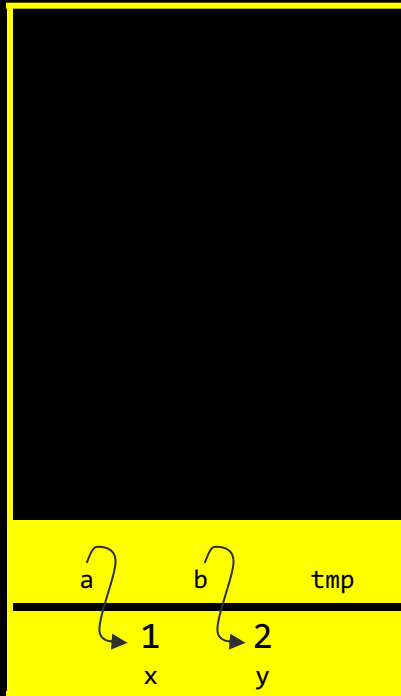
Übergabe als Referenz


```
void swap(int *a, int *b)
{
    int tmp = *a;
    *a = *b;
    *b = tmp;
}
```

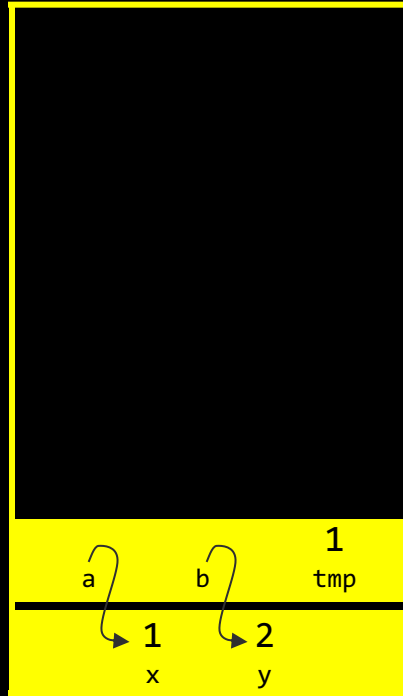
1	2
x	y



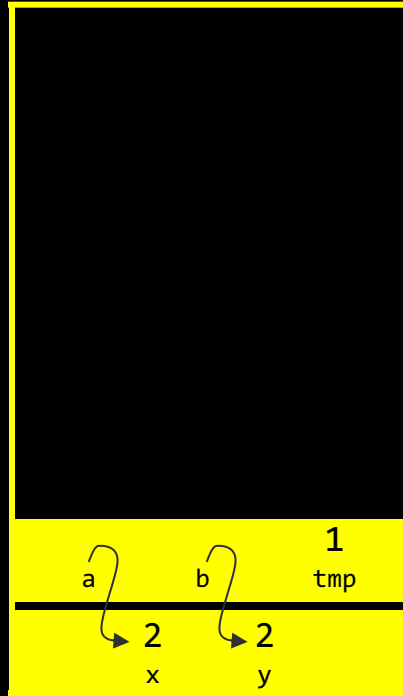
```
int tmp = *a;  
*a = *b;  
*b = tmp;
```



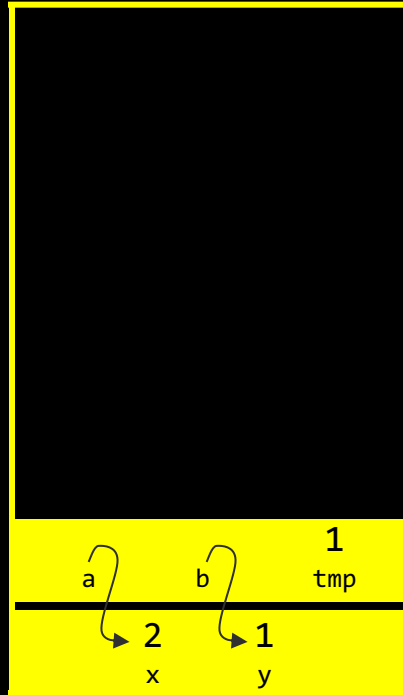
```
int tmp = *a;  
*a = *b;  
*b = tmp;
```

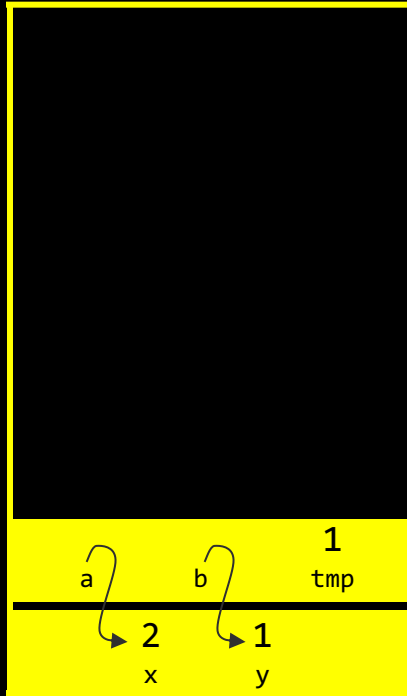


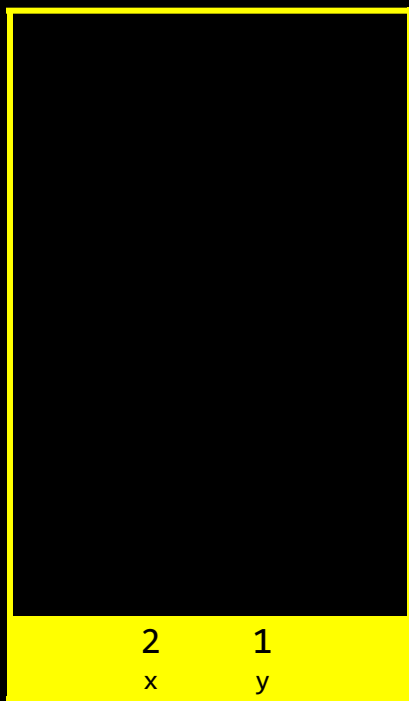
```
int tmp = *a;  
*a = *b;  
*b = tmp;
```



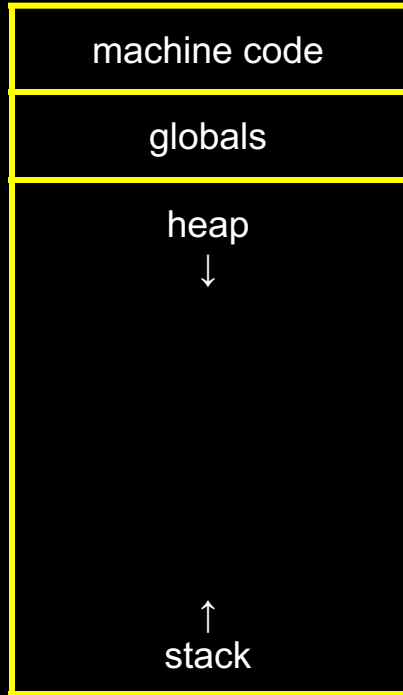
```
int tmp = *a;  
*a = *b;  
*b = tmp;
```

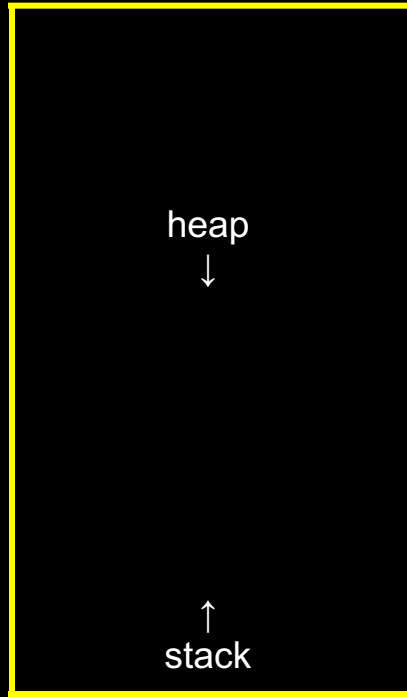







```
void swap(int *a, int *b)
{
    int tmp = *a;
    *a = *b;
    *b = tmp;
}
```





heap overflow

stack overflow

buffer overflow

get_char

get_double

get_float

get_int

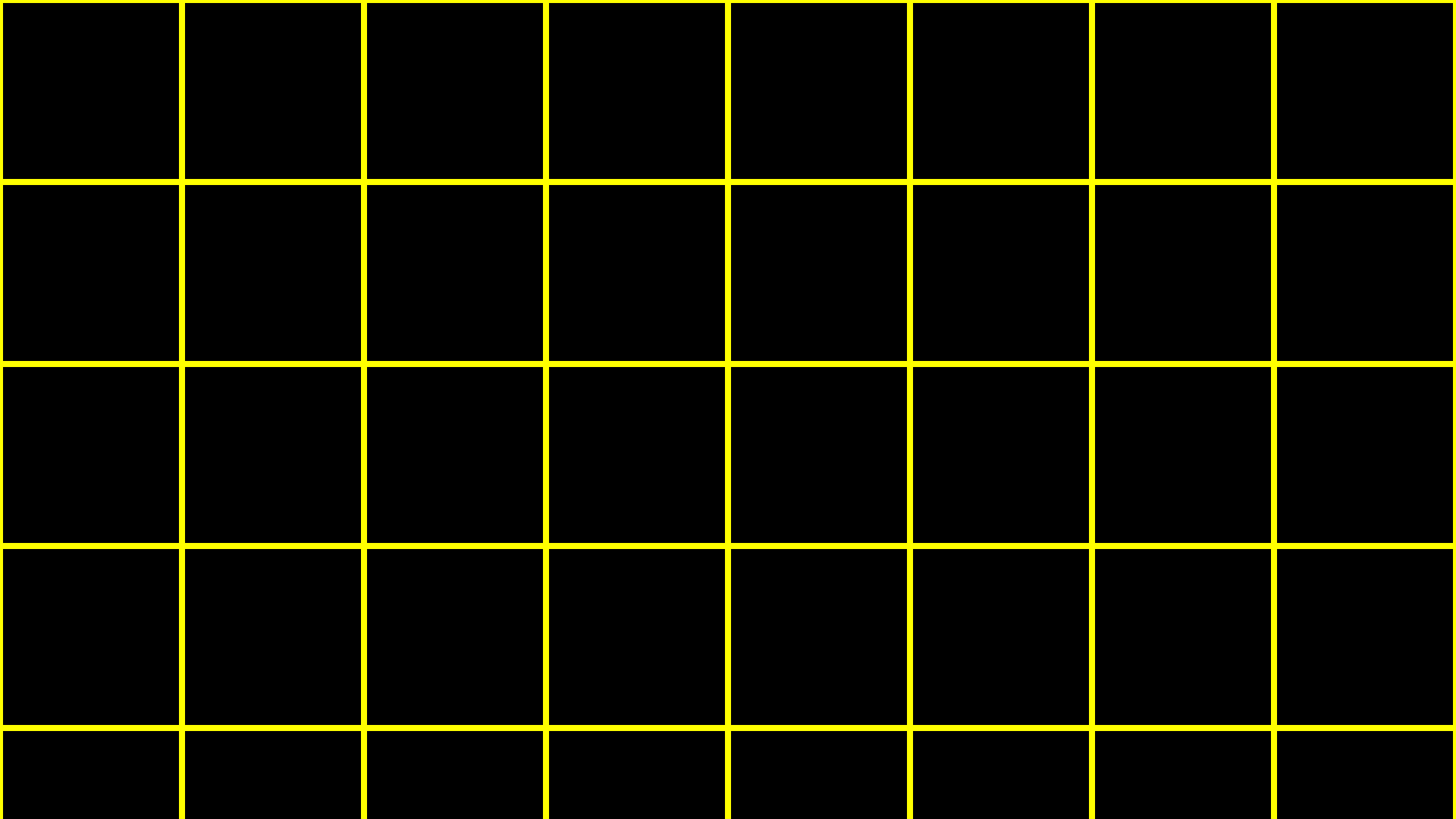
get_long

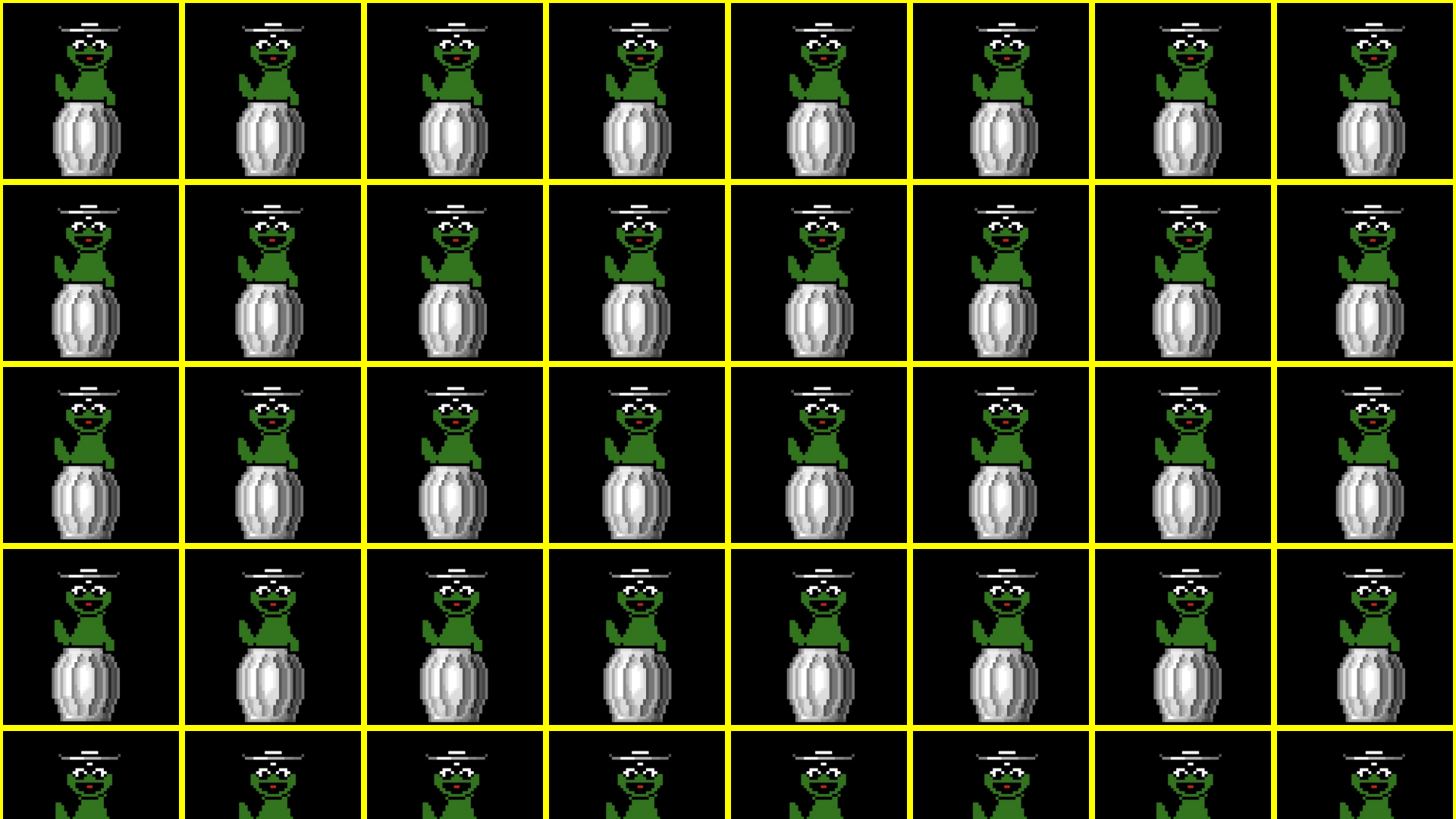
get_string

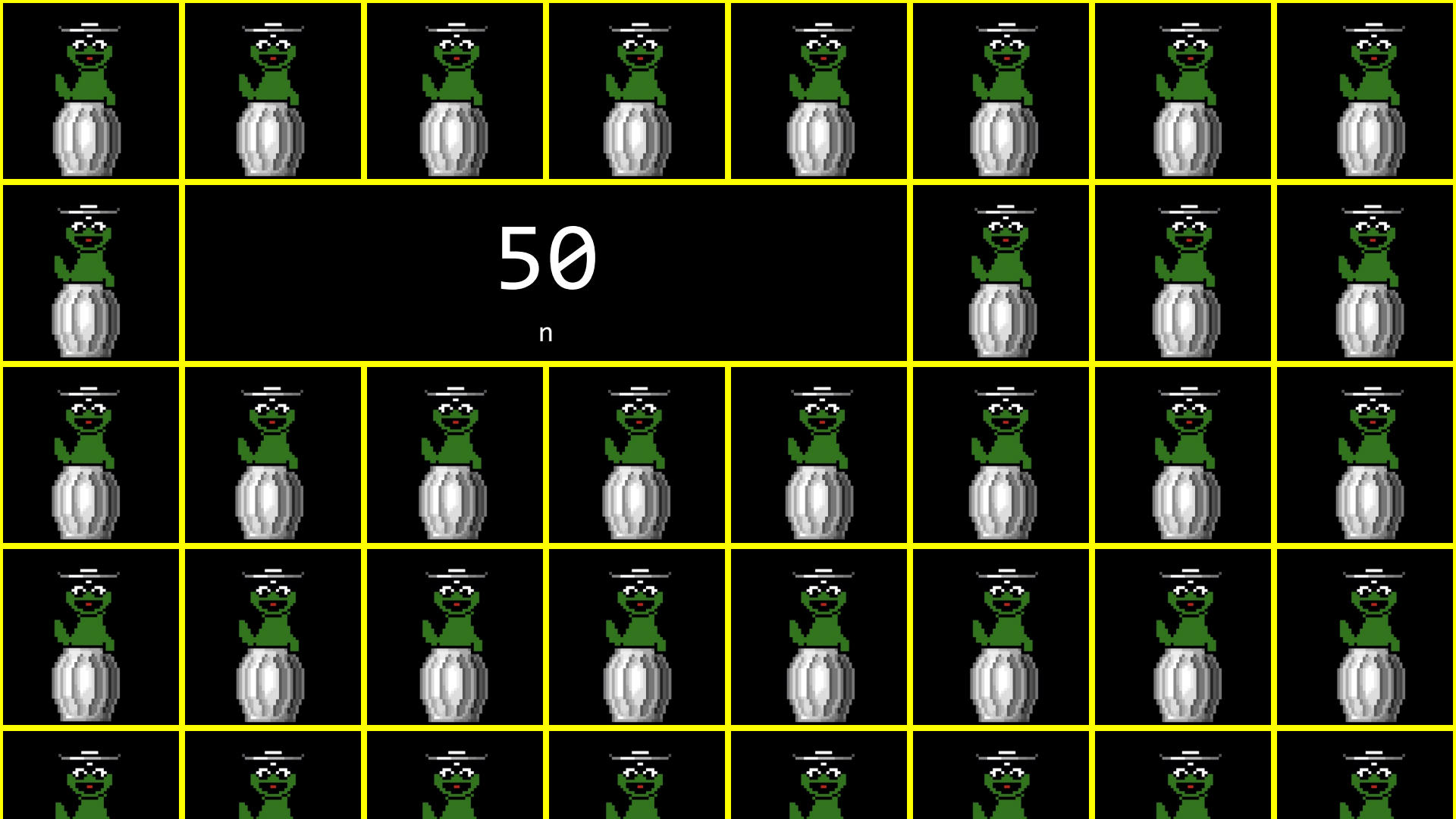
...

scanf

...

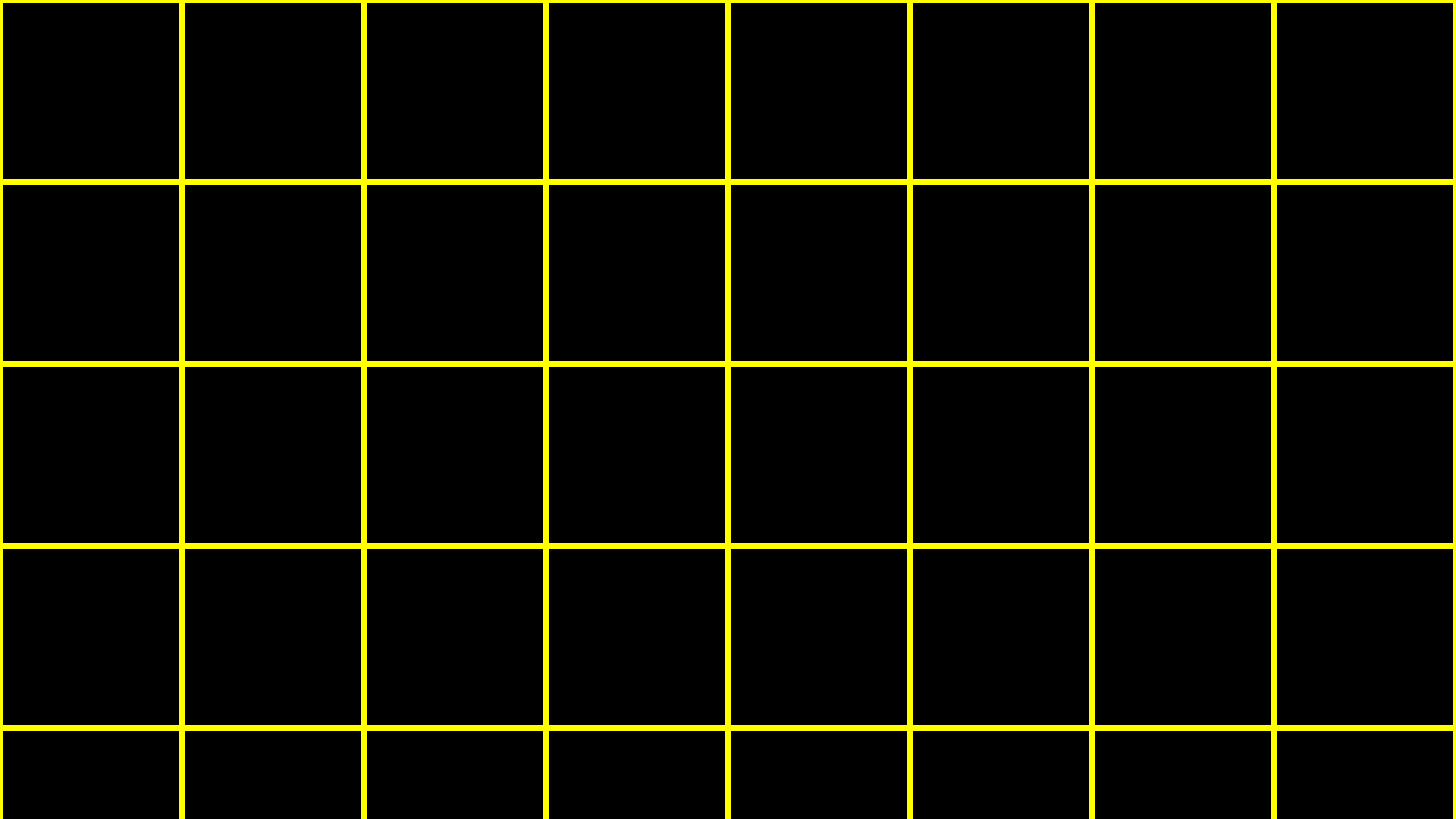


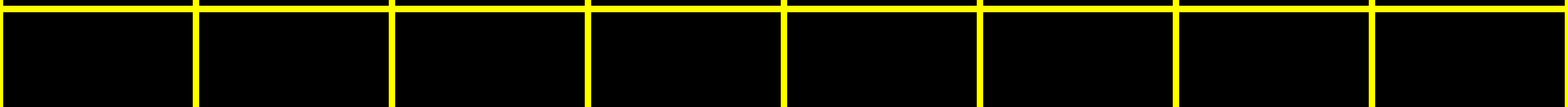
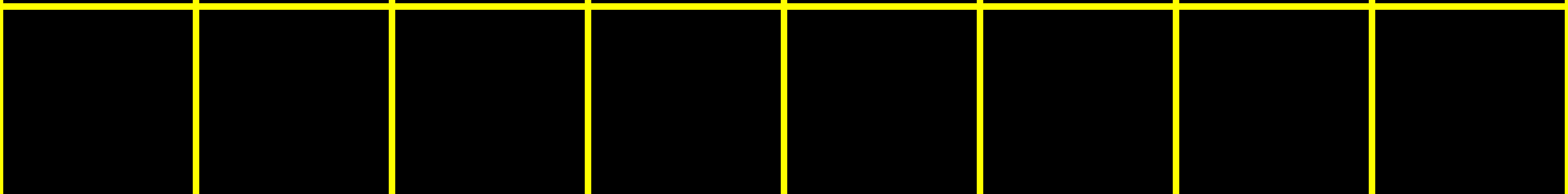
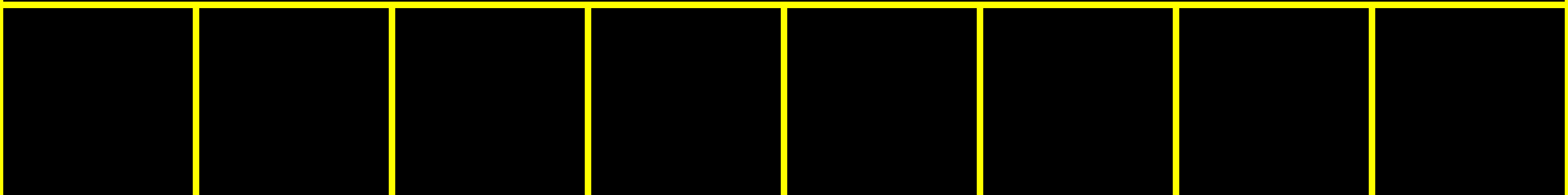
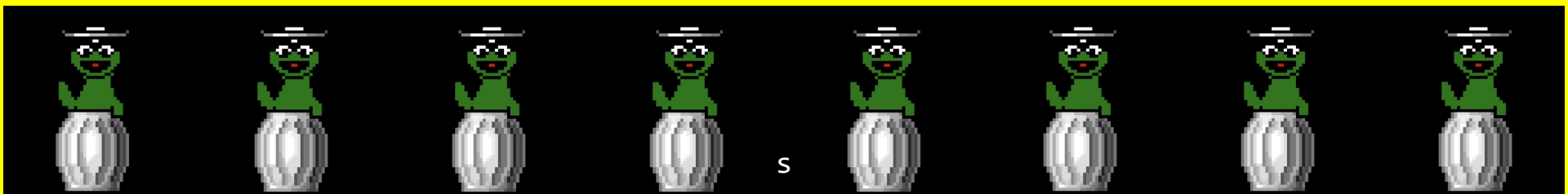
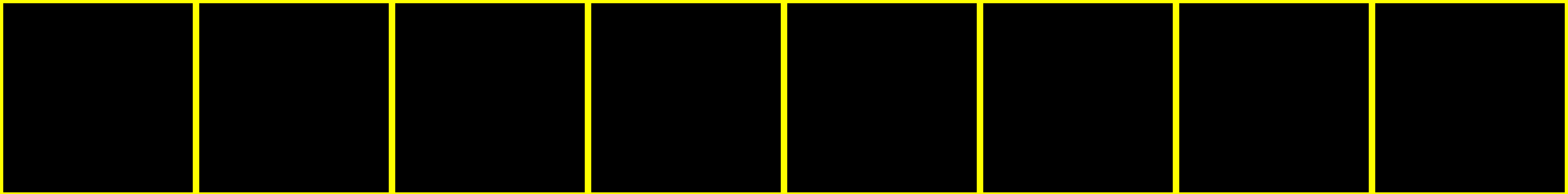


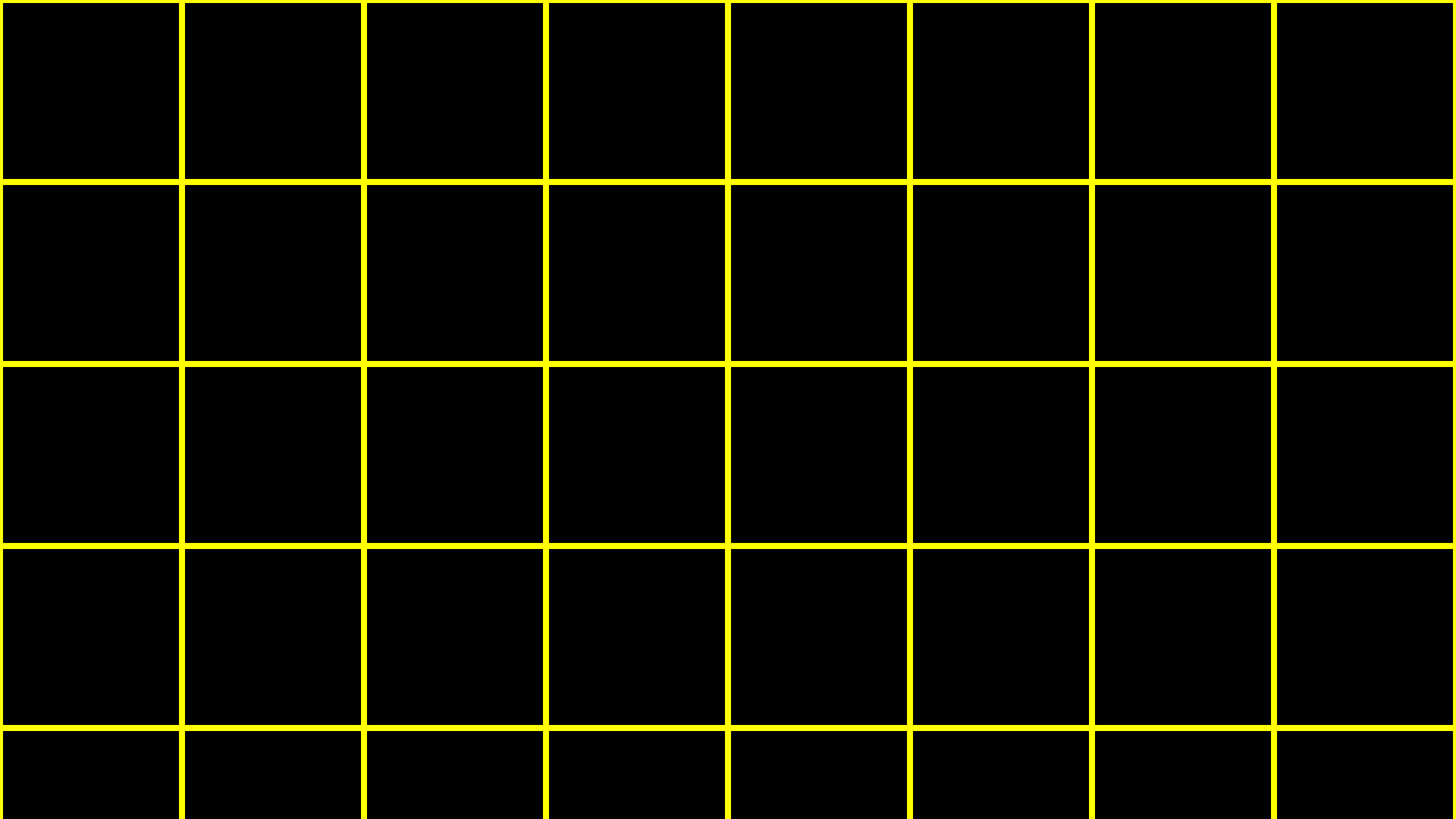


50

n







S							

File-I/O

Input und Output

fopen

fclose

fprintf

fscanf

fread

fwrite

fseek

...

BMP













MAN, I SUCK AT THIS GAME.
CAN YOU GIVE ME
A FEW POINTERS?

0x3A28213A
0x6339392C,
0x7363682E.

I HATE YOU.



Dies war Inf-Einf-B.