

This is CS50



```
#include <stdio.h>

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

- functions
 - arguments, return values
- conditionals
- Boolean expressions
- loops
- variables
- ...



ANSEL IOSTAN
ALUMNUS CINCINNATIENSIS
ADVENTUS IN CINCINNATI
QUI CINCINNATI
AD ANNO CCCXXXV NATUS
AD ANNO CCCXLII
TEST. FIERI MEST.

ANSEL IOSTAN
ADVENTUS IN CINCINNATI
QUI CINCINNATI
AD ANNO CCCXXXV NATUS
AD ANNO CCCXLII
TEST. FIERI MEST.



correctness

design

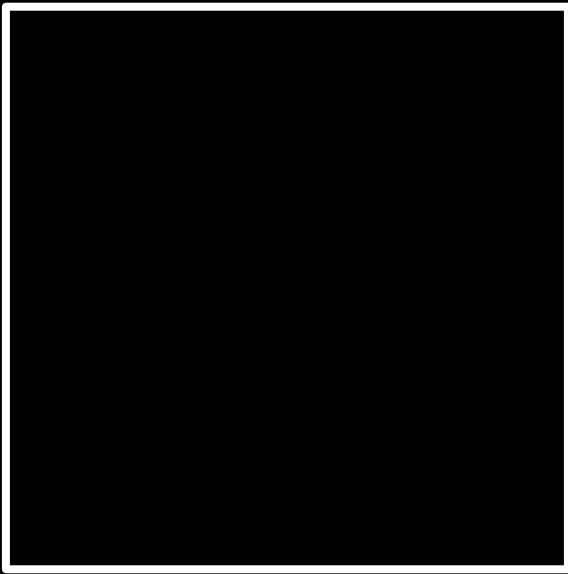
style

```
#include <stdio.h>

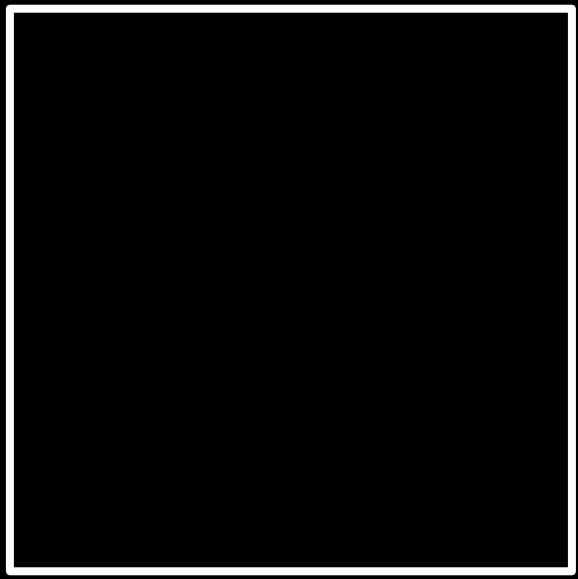
int main(void)
{
    printf("hello, world\n");
}
```

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
...

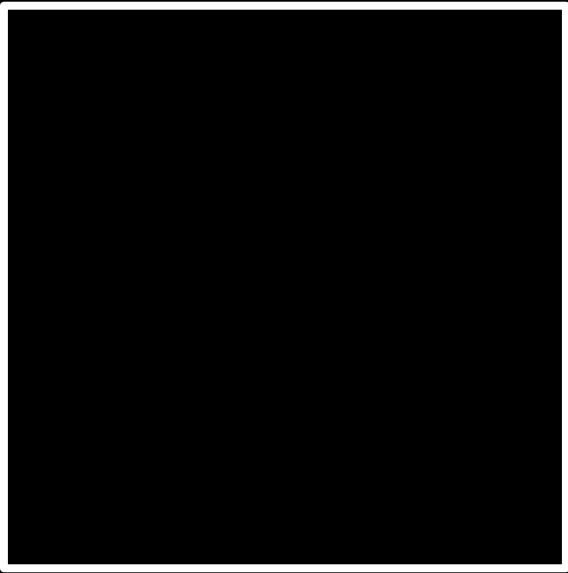
input →



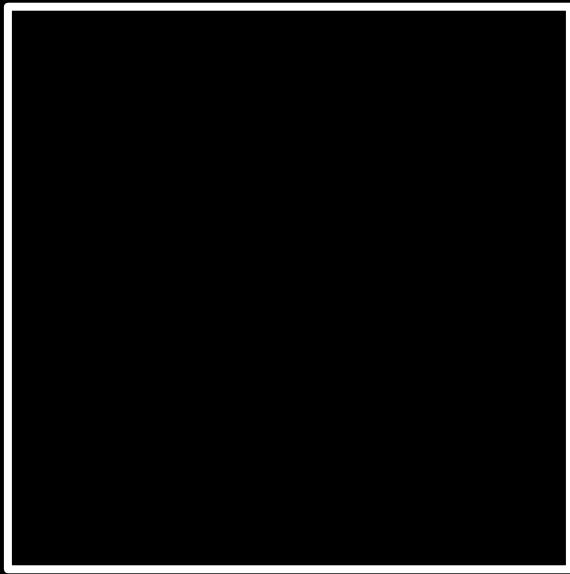
→ output



source code →

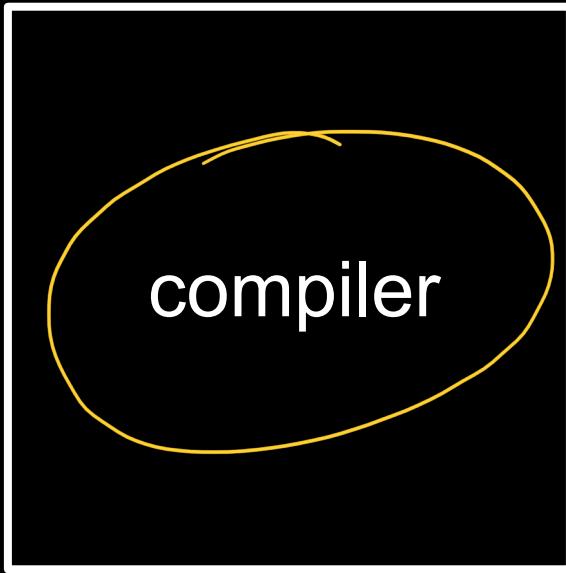


source code →



→ machine code

source code →



→ machine code

```
make hello
```

```
./hello
```

functions, arguments

say

hello, world

say hello, world

print ()



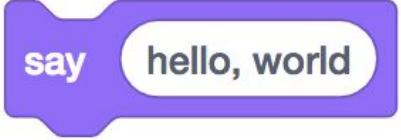
function

say hello, world

printf()

C

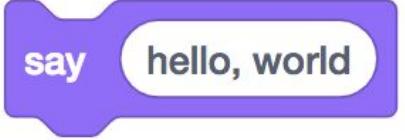
formatted



say

hello, world

```
printf( hello, world )
```



printf("hello, world")
for string.
double quote

A handwritten note in yellow ink. It starts with the C function "printf("hello, world")" in white. Two circles are drawn around the opening and closing double quotes. A curved arrow points from the word "double quote" at the top right down towards these two circles. Below the function, the words "for string." are written, followed by a period.



printf("hello, world");



semi colon
finish.

functions

arguments

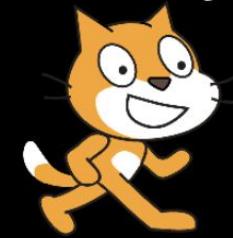
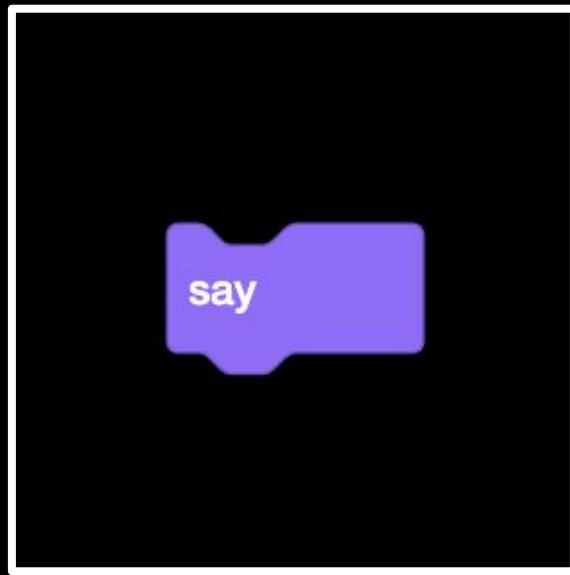
functions

arguments →

functions

→ side effects

hello, world



hello, world

return values, variables

ask

What's your name? and wait

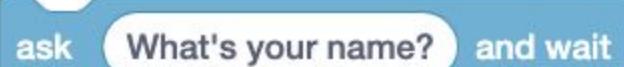
answer

ask What's your name? and wait

answer

function

get_string()



```
ask [What's your name?] and wait
```



```
answer
```

arguments

```
get_string("What's your name? ")
```

ask What's your name? and wait

answer

variable

✓ ✓
answer = get_string("What's your name? ")

assignment
operator.

ask What's your name? and wait

answer

type of value

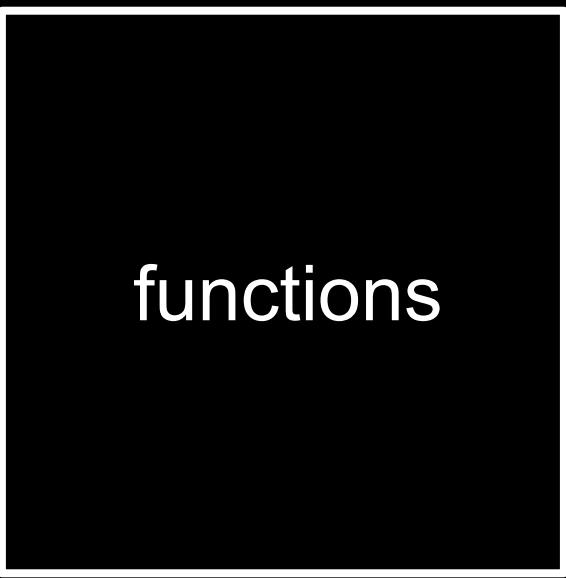
string answer = get_string("What's your name? ")

ask What's your name? and wait

answer

string answer = get_string("What's your name? ");

Standard I/O



<stdio.h>
library

<sstream>
string

%s
string

arguments →

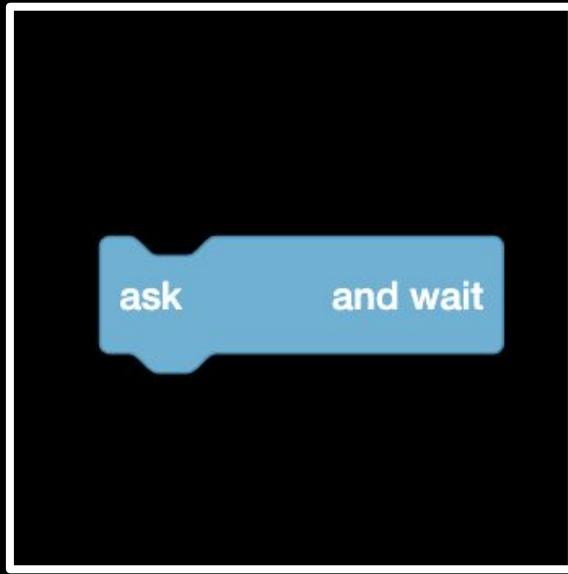
functions

arguments →

functions

→ return value

What's your name?



answer

say

join

hello,

answer

say join

hello,

answer

```
printf( );
```

say

join

hello,

answer

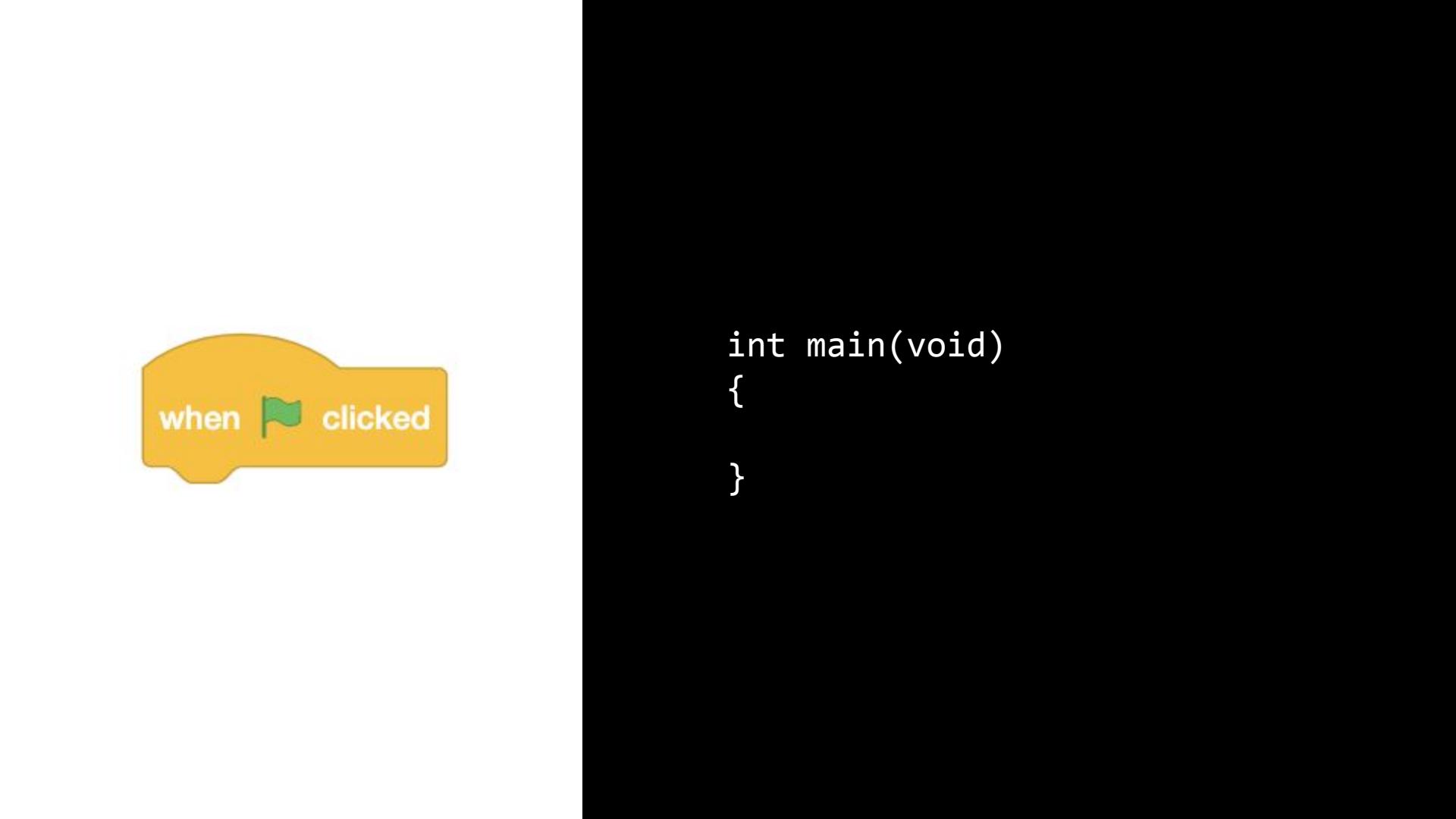
```
printf("hello, %s" );
```



```
printf("hello, %s", answer);
```

main

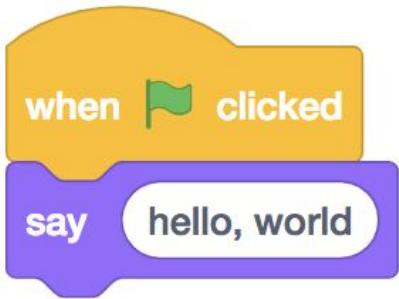
when  clicked



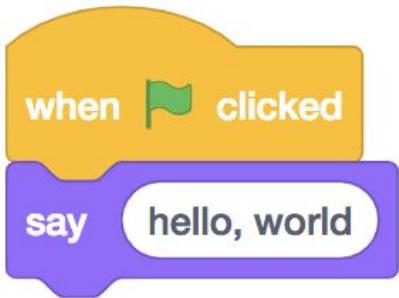
when  clicked

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

header files



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



```
#include <stdio.h>

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

cd

cp

ls

mkdir

mv

rm

rmdir

...

types

bool

char

double

float

int

long

string

...

`get_char`

`get_double`

`get_float`

`get_int`

`get_long`

`get_string`

`...`

format codes

%c

%f

float

%i

int

%li

long int

%s

string

%c char

%f float, double

%i int

%li long

%s string

operators

+

-

*

/

%

Gemander

- + addition
- subtraction
- * multiplication
- / division
- % remainder

variables, syntactic sugar





counter = 0



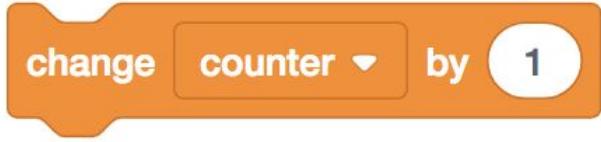


int counter = 0

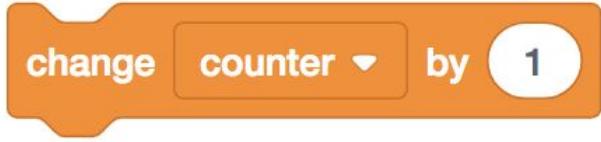
data type



int counter = 0;



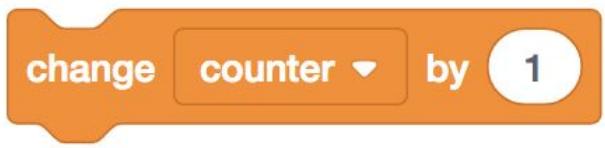
← increment



counter = counter + 1



The text "counter = counter + 1" is displayed on a black background. A yellow curved arrow points from the word "counter" in the first term of the assignment statement to the word "counter" in the second term. A horizontal yellow line underlines the entire assignment statement.



counter = counter + 1;

change counter by 1

sugar

+1

counter += 1;



change counter by 1

+ \
counter++;



8 bits \rightarrow 256

3 2 bits
 \rightarrow 4 billions
2 billion ~~per~~ $\times 1 -$

conditions



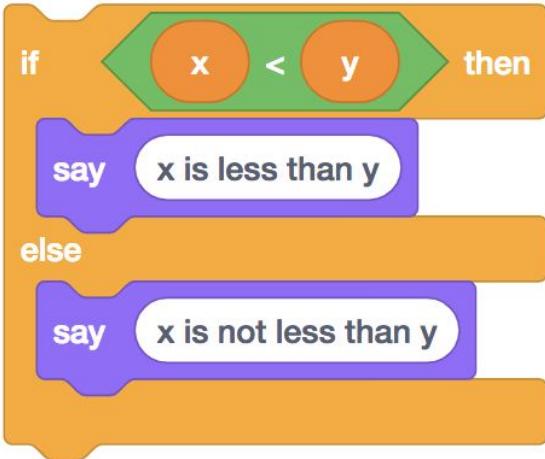


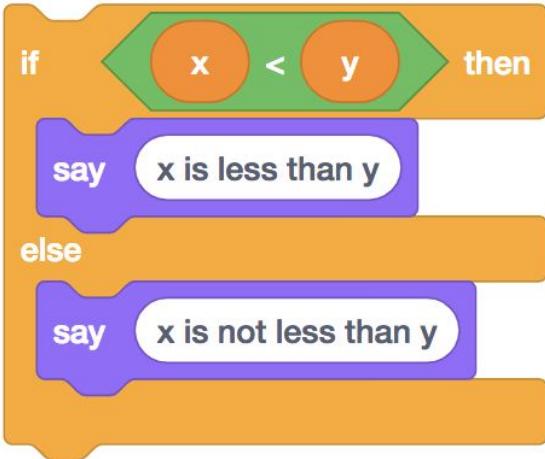
```
if (x < y)
{
}
```



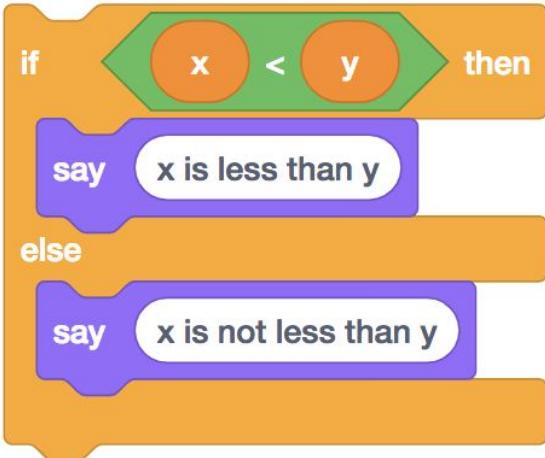
if ($x < y$)
{
}
printf("x is less than y\n");

↑
construct



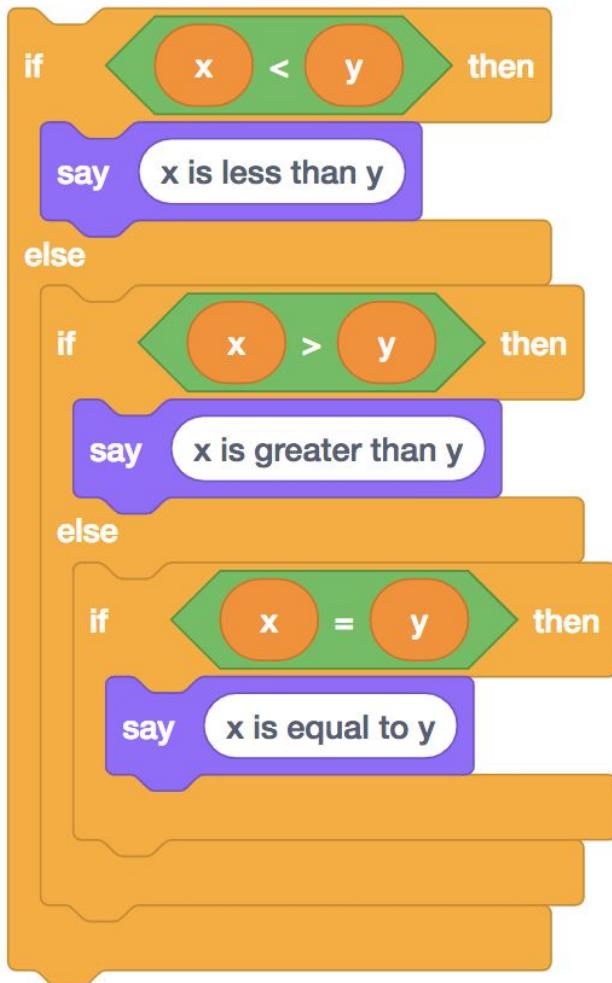


```
if (x < y)
{
}
else
{
}
```

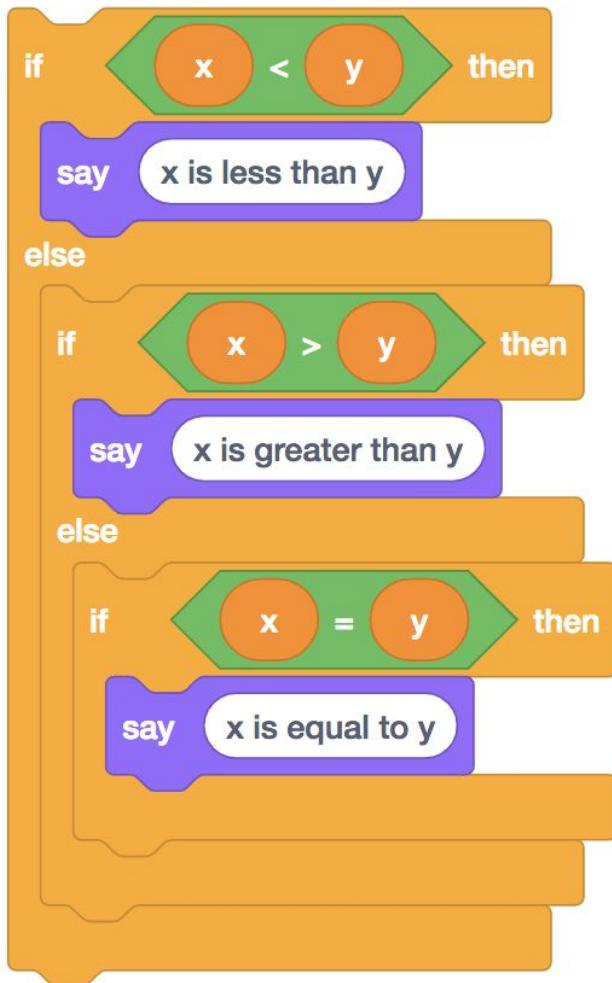


```
if (x < y)
{
    printf("x is less than y\n");
}
else
{
    printf("x is not less than y\n");
}
```

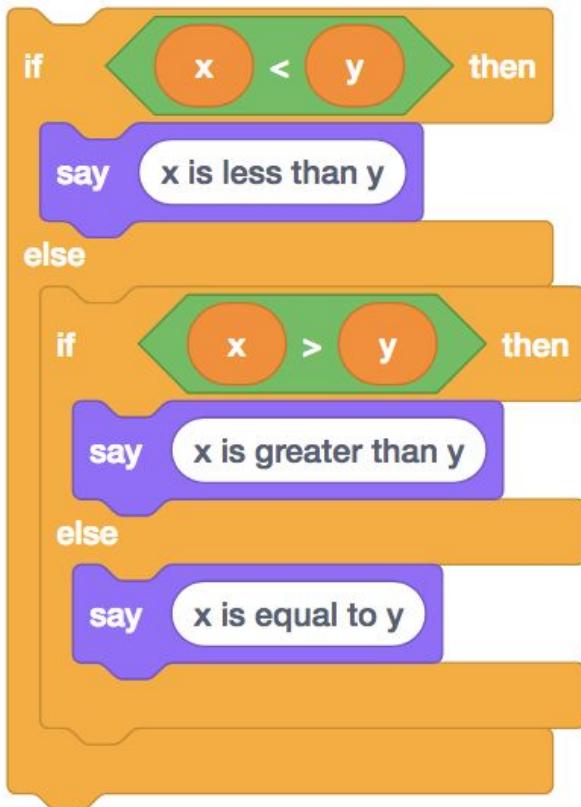
```
if x < y then  
  say x is less than y  
else  
  if x > y then  
    say x is greater than y  
  else  
    if x = y then  
      say x is equal to y
```



```
if (x < y)
{
}
else if (x > y)
{
}
else if (x == y)
{
}
```



```
if (x < y)
{
    printf("x is less than y\n");
}
else if (x > y)
{
    printf("x is greater than y\n");
}
else if (x == y)
{
    printf("x is equal to y\n");
}
```



```
if (x < y)
{
    printf("x is less than y\n");
}
else if (x > y)
{
    printf("x is greater than y\n");
}
else
{
    printf("x is equal to y\n");
}
```

better design

loops





```
while (true)
{
}
```



```
while (true)
{
    printf("meow\n");
}
```





```
int counter = 0;  
while (counter < 3)  
{  
}  
}
```



```
int counter = 0;  
while (counter < 3)  
{  
    printf("meow\n");  
}  
}
```



```
int counter = 0;  
while (counter < 3)  
{  
    printf("meow\n");  
    counter = counter + 1;  
}
```



int i
int
int i = 0;
while (i < 3)
{
 printf("meow\n");
 i = i + 1;
}



```
int i = 0;  
while (i < 3)  
{  
    printf("meow\n");  
    i += 1;  
}
```



```
int i = 0;  
while (i < 3)  
{  
    printf("meow\n");  
    i++;  
}
```

sugar



```
int i = 0;  
while (i < 3)  
{  
    printf("meow\n");  
    i++;  
}
```



```
int i = 0;  
while (i < 3)  
{  
    printf("meow\n");  
    i++;  
}
```



```
int i = 0;  
while (i < 3)  
{  
    printf("meow\n");  
    i++;  
}
```



loop

```
int i = 0;  
while (i < 3)  
{  
    printf("meow\n");  
    i++;  
}
```



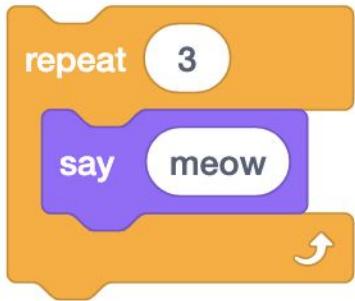
```
int i = 0;  
while (i < 3)  
{  
    printf("meow\n");  
    i++;  
}
```



```
int i = 0;  
while (i < 3)  
{  
    printf("meow\n");  
    i++;  
}
```



```
int i = 0;  
while (i < 3)  
{  
    printf("meow\n");  
    i++;  
}
```



```
int i = 1;  
while (i <= 3)  
{  
    printf("meow\n");  
    i++;  
}
```



```
int i = 3;  
while (i > 0)  
{  
    printf("meow\n");  
    i--;
```

T

decrement

Count down





```
for (int i = 0; i < 3; i++)  
{  
}  
}
```



```
for (int i = 0; i < 3; i++)  
{  
    printf("meow\\n");  
}
```



```
for (int i = 0; i < 3; i++)  
{  
    printf("meow\\n");  
}
```



```
for (int i = 0; i < 3; i++)
{
    printf("meow\n");
}
```



```
for (int i = 0; i < 3; i++)  
{  
    printf("meow\n");  
}
```



```
for (int i = 0; i < 3; i++)  
{  
    printf("meow\n");  
}
```



```
for (int i = 0; i < 3; i++)
{
    printf("meow\n");
}
```

for | while loop
can do the same thing



```
for (int i = 0; i < 3; i++)  
{  
    printf("meow\n");  
}
```

FPS : 46.04 . RFPS : 46.04

MARIO
OOOOOO

0x00

WORLD
1-1

TIME

SUPER MARIO BROS.

©1985 NINTENDO



1 PLAYER GAME

2 PLAYER GAME

TOP - OOOOOO



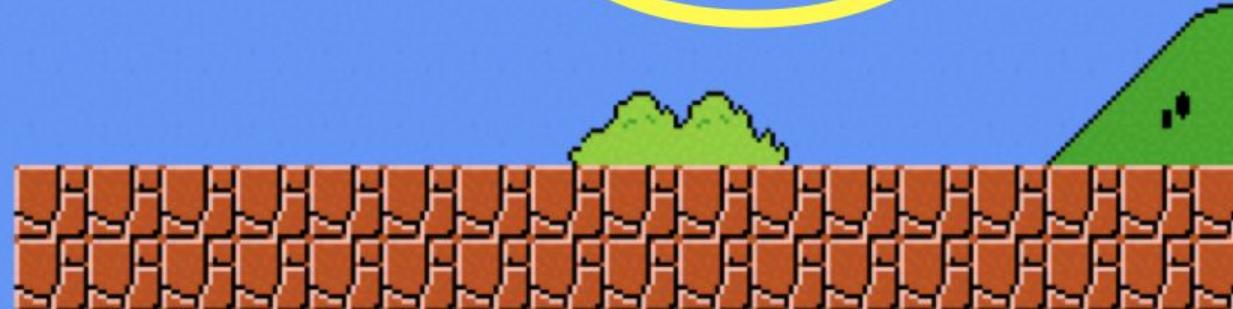
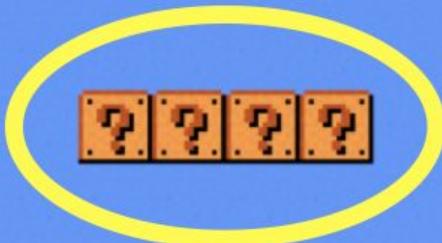


?????

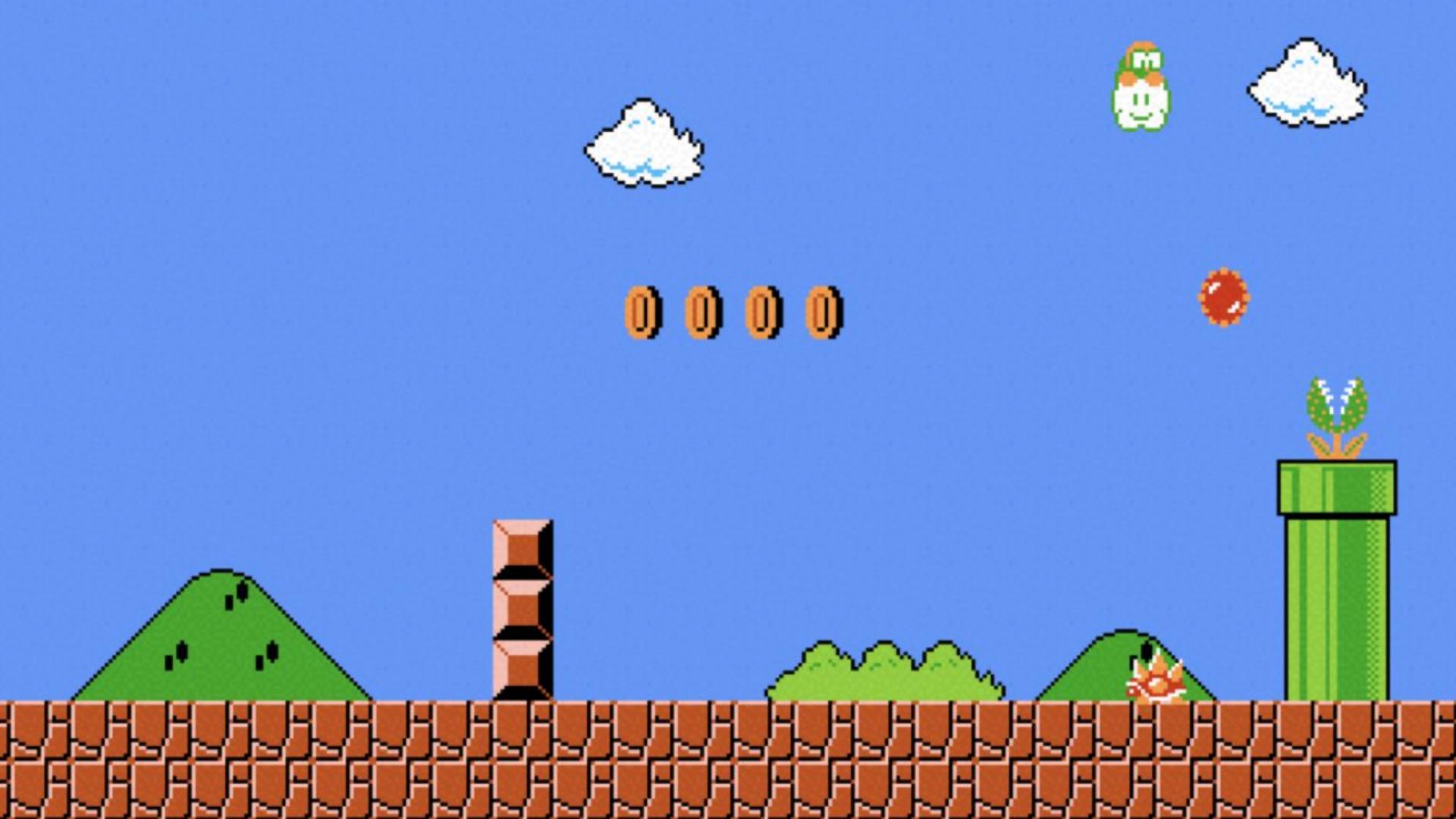
A row of five question mark blocks, each enclosed in an orange border, arranged horizontally in the center of the stage.



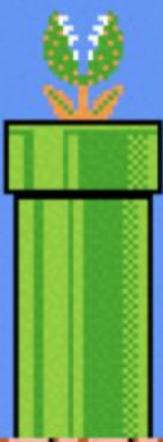
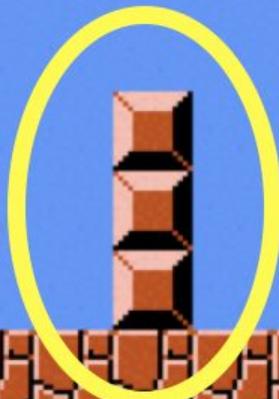
?????

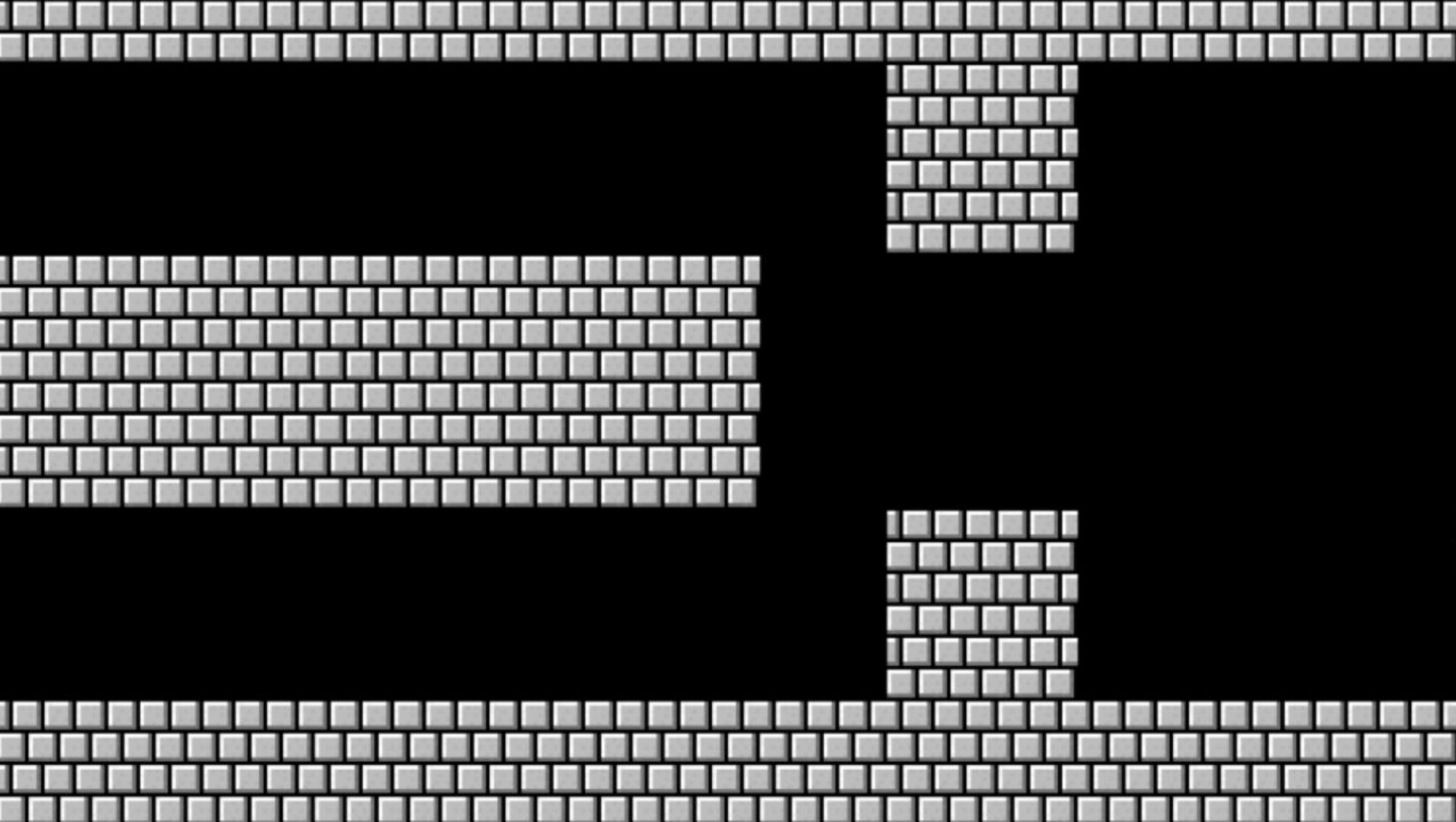


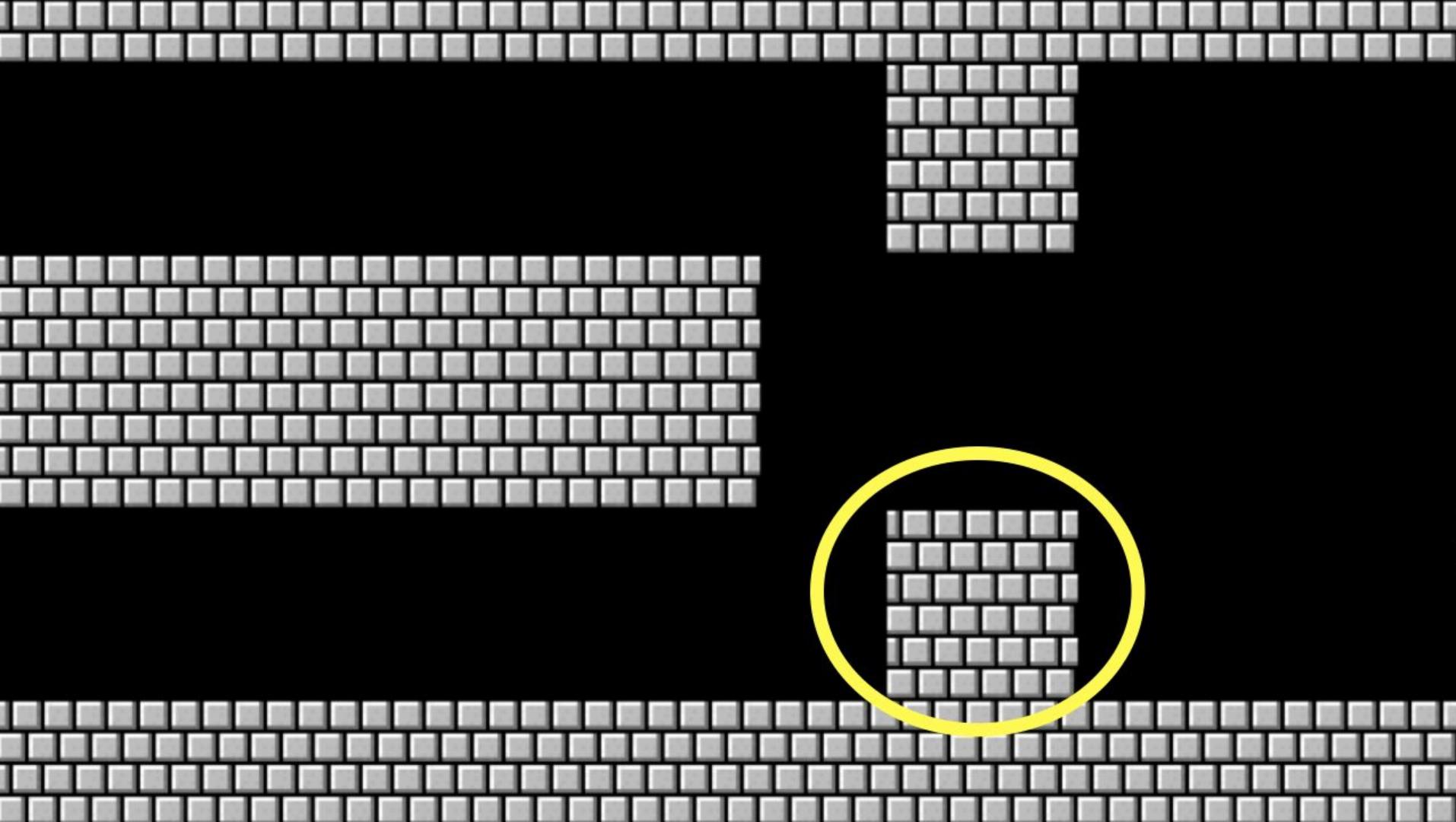
0 0 0 0



0 0 0 0









floating-point imprecision

integer overflow

000

001

010

011

100

101

110

111

1000

3 bits

A

000

1 January 2000

1999

1999

1900

1 Jan 1970
second.
2 billion

19 January 2038

2147483647

011

011

1

1

01100

011111111111111111111111111111110000

0111111111111111111111111111111100000

01111111111111111111111111111111000000

011111111111111111111111111111110000000

0111111111111111111111111111111100000000

011111111111111111111111110000000000

01111111111111111111111111111111000000000000

011111111111111111111111000000000000

0111111111111111111111111111000000000000000

011111111111111111110000000000000000

011111111111111111110000000000000000

01111111111111110000000000000000

011111111111111100000000000000000000

01111111111111000000000000000000000000

0111111111111000000000000000000000000

01111111111100000000000000000000000000

011111111110000000000000000000000000000

01111111110000000000000000000000000000

011111111000000000000000000000000000000

0111111100000000000000000000000000000000

0111111100

011111100

01111100

0111100

0111000

01100

01000

64 bits → solve issues

-2147483648

13 December 1901





This is CS50