

```
const W: u32 = 10;
const H: u32 = 10;
```

```
for y:u32 in 0..<H {
  for x:u32 in 0..<W {
    let is_horizontal:bool = y == 0 || y == H - 1;
    let is_vertical:bool = x == 0 || x == W - 1;
    let is_diagonal:bool = x == y;

    let c:char = if is_horizontal || is_vertical || is_diagonal {
      '*'
    } else {
      ' '
    };
    print!("{}", c);
  }
  println!();
}
```

$$\begin{aligned}
 y=0 & \quad x = W-1 \\
 y=1 & \quad x = W-2 \\
 y=2 & \quad x = 7
 \end{aligned}$$



```
for y:u32 in 0..<H {  
  for x:u32 in 0..<W {  
    let is_horizontal:bool = y == 0 || y == H - 1;  
    let is_vertical:bool = x == 0 || x == W - 1;  
    let is_diagonal:bool = x == y;  
    let is_co_diagonal:bool = x == W - 1 - y;  
  
    let c:char = if is_horizontal || is_vertical || is_diagonal || is_co_diagonal {  
      '*'  
    } else {  
      ' '  
    };  
    print!("{}", c);  
  }  
  println!();  
}
```

```
*****  
**               **  
*  *           *  *  
*    *       *    *  
*      **      *  
*      **      *  
*    *       *    *  
*  *           *  *  
**               **  
*****
```



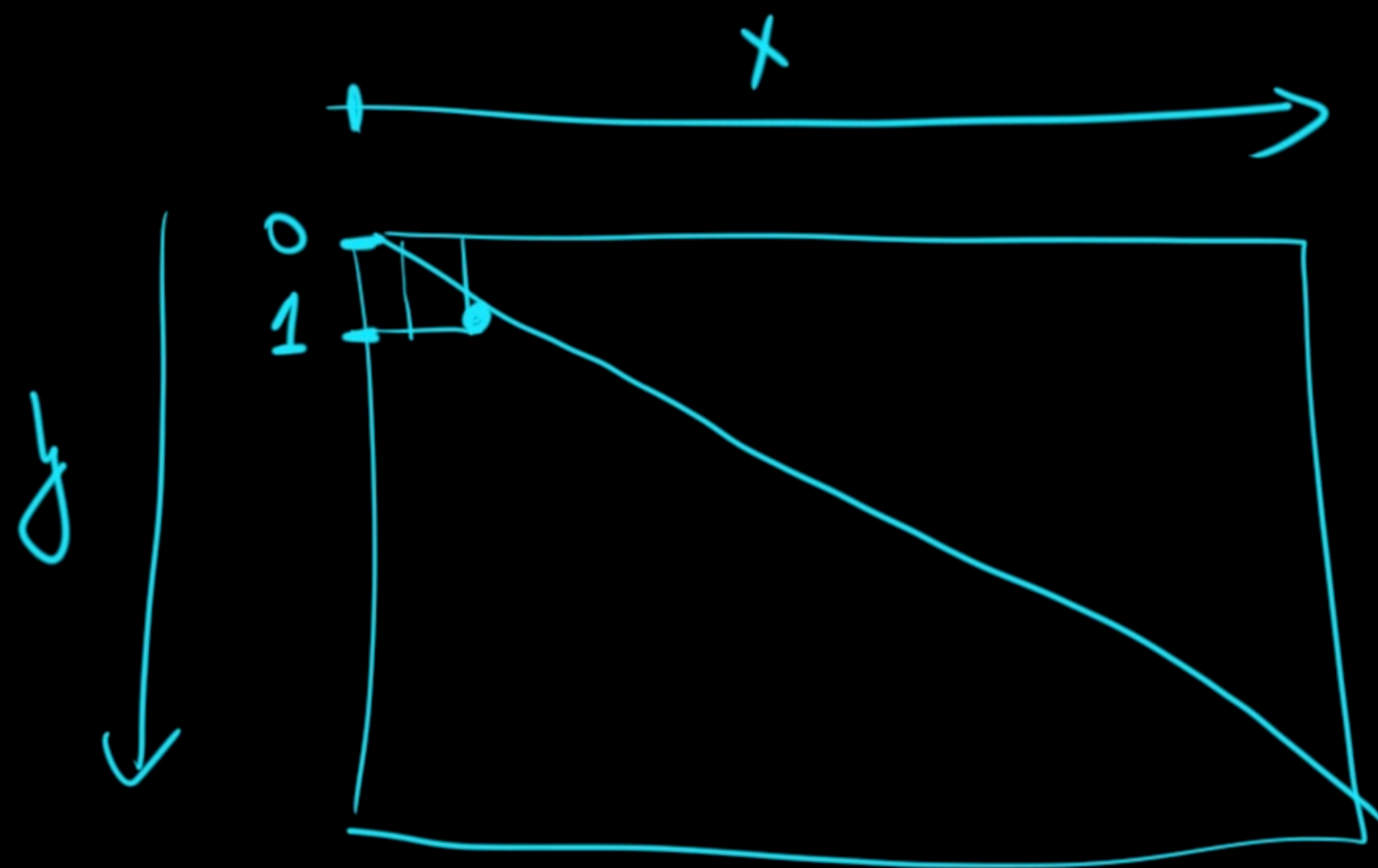
```

const W: u32 = 25;
const H: u32 = 10;

for y: u32 in 0..H {
  for x: u32 in 0..W {
    let is_horizontal: bool = y == 0 || y == H - 1;
    let is_vertical: bool = x == 0 || x == W - 1;
    let is_diagonal: bool = x == y;
    let is_co_diagonal: bool = x == W - 1 - y;
  }
}

```

$W = 20$   
 $H = 10$



```

*****
**                                     **
*  *                               *  *
*   *                             *   *
*    *                           *    *
*     *                         *     *
*      *                       *      *
*       *                     *       *
*        *                   *        *
*         *                 *         *
*****

```

$$x = y \cdot k$$

$$k = \frac{W}{H}$$

$$x = W - 1 - y \cdot k$$



```

let is_horizontal:bool = y == 0 || y == H - 1;
let is_vertical:bool = x == 0 || x == W - 1;
let is_diagonal:bool = x == (y as f32 * k).round() as u32;
let is_co_diagonal:bool = x == W - 1 - (y as f32 * k).round() as u32;

let c:char = if is_horizontal || is_vertical || is_diagonal || is_co_diagonal {
    '*'
} else {
    ' '
};

```

```

let is_diagonal:bool = x == (y as f32 * k).round() as u32;
let is_co_diagonal:bool = x == W - (y as f32 * k).round() as u32;

let c:char = match (x, y) {
    (1, _) => '*',
    (_, 1) => '*', true
    (W, _) => '*',
    (_, H) => '*',
    _ if is_diagonal => '*',
    _ if is_co_diagonal => '*',
    _ => ' ', false
};

```

```

let c:char = match (x, y) {
    (1 | W, _) => '|',
    (_, 1 | H) => '-',
    _ if is_diagonal => '\\',
    _ if is_co_diagonal => '/',
    _ => ' ',
};

```

if (x == 1 || x == W) then |



```
if x == 5 {  
    print!("*")  
} else {  
    print!(" ")  
}
```

```
match x {  
    5 => print!("*"),  
    _ => print!(" "),  
}
```