



← Back to graph

quad

AVAILABLE

Repo to create for this project

<https://learn.01founders.co/git/swin/quad>

</>

XP

👤

go

1.85 kB

3

quadA

Instructions

Write a function `QuadA` that prints a **valid** rectangle with a given width of `x` and height of `y`.

The function must draw the rectangles as in the examples.

If `x` and `y` are positive numbers, the program should print the rectangles as seen in the examples, otherwise, the function should print nothing.

Make sure you submit all the necessary files to run the program.

Expected function



Usage

Here are possible programs to test your function :

Program #1

```
package main

import "piscine"

func main() {
    piscine.QuadA(5, 3)
}
```

And its output :

```
$ go run .
o --- o
|       |
o --- o
$
```

Program #2

```
package main

import "piscine"

func main() {
    piscine.QuadA(5, 1)
}
```

And its output :

```
$ go run .
o --- o
$
```

Program #3



```
func main() {
    piscine.QuadA(1,1)
}
```

And its output :

```
$ go run .
o
$
```

Program #4

```
package main

import "piscine"

func main() {
    piscine.QuadA(1,5)
}
```

And its output :

```
$ go run .
o
|
|
|
|
o
$
```

quadB

Instructions

Write a function `QuadB` that prints a **valid** rectangle of width `x` and of height `y`.

The function must draw the rectangles as in the examples.



```
func QuadB(x, y int) {  
}
```

Usage

Here are possible programs to test your function :

Program #1

```
package main  
  
import "piscine"  
  
func main() {  
    piscine.QuadB(5, 3)  
}
```

And its output :

```
$ go run .  
/* */  
* *  
\***/  
$
```

Program #2

```
package main  
  
import "piscine"  
  
func main() {  
    piscine.QuadB(5, 1)  
}
```

And its output :

```
$ go run .  
/* */
```



```
package main

import "piscine"

func main() {
    piscine.QuadB(1,1)
}
```

And its output :

```
$ go run .
/
$
```

Program #4

```
package main

import "piscine"

func main() {
    piscine.QuadB(1,5)
}
```

And its output :

```
$ go run .
/
*
*
*
\
$
```

quadC

Instructions



seen in the examples, otherwise, the function should print nothing.

Expected function

```
func QuadC(x, y int) {  
}
```

Usage

Here are possible programs to test your function :

Program #1

```
package main  
  
import "piscine"  
  
func main() {  
    piscine.QuadC(5, 3)  
}
```

And its output :

```
$ go run .  
ABBA  
B B  
CBBC  
$
```

Program #2

```
package main  
  
import "piscine"  
  
func main() {  
    piscine.QuadC(5, 1)  
}
```



\$

Program #3

```
package main

import "piscine"

func main() {
    piscine.QuadC(1,1)
}
```

And its output :

```
$ go run .
A
$
```

Program #4

```
package main

import "piscine"

func main() {
    piscine.QuadC(1,5)
}
```

And its output :

```
$ go run .
A
B
B
B
C
$
```



Write a function `Quadd` that prints a **valid** rectangle of width `x` and of height `y`.

The function must draw the rectangles as in the examples.

If `x` and `y` are positive numbers, the program should print the rectangles as seen in the examples, otherwise, the function should print nothing.

Expected function

```
func Quadd(x, y int) {  
}
```

Usage

Here are possible programs to test your function :

Program #1

```
package main  
  
import "piscine"  
  
func main() {  
    piscine.Quadd(5, 3)  
}
```

And its output :

```
$ go run .  
ABBC  
B B  
ABBC  
$
```

Program #2

```
package main  
  
import "piscine"
```



And its output :

```
$ go run .
ABBC
$
```

Program #3

```
package main

import "piscine"

func main() {
    piscine.Quadd(1,1)
}
```

And its output :

```
$ go run .
A
$
```

Program #4

```
package main

import "piscine"

func main() {
    piscine.Quadd(1,5)
}
```

And its output :



B
A
\$

quadE

Instructions

Write a function `QuadE` that prints a **valid** rectangle of width `x` and of height `y`.

The function must draw the rectangles as in the examples.

If `x` and `y` are positive numbers, the program should print the rectangles as seen in the examples, otherwise, the function should print nothing.

Expected function

```
func QuadE(x,y int) {  
}
```

Usage

Here are possible programs to test your function :

Program #1

```
package main  
  
import "piscine"  
  
func main() {  
    piscine.QuadE(5,3)  
}
```

And its output :



\$

Program #2

```
package main

import "piscine"

func main() {
    piscine.QuadE(5,1)
}
```

And its output :

```
$ go run .
ABBC
$
```

Program #3

```
package main

import "piscine"

func main() {
    piscine.QuadE(1,1)
}
```

And its output :

```
$ go run .
A
$
```

Program #4

```
package main

import "piscine"

func main() {
```



```
$ go run .
A
B
B
B
C
$
```

Something is wrong ? Submit an issue or even better ! propose a change !



Your group



SEARCH VIII

SWIFT

EXTRA REWARDS



Nikolaos Makridis

nmakridi

EXTRA REWARDS