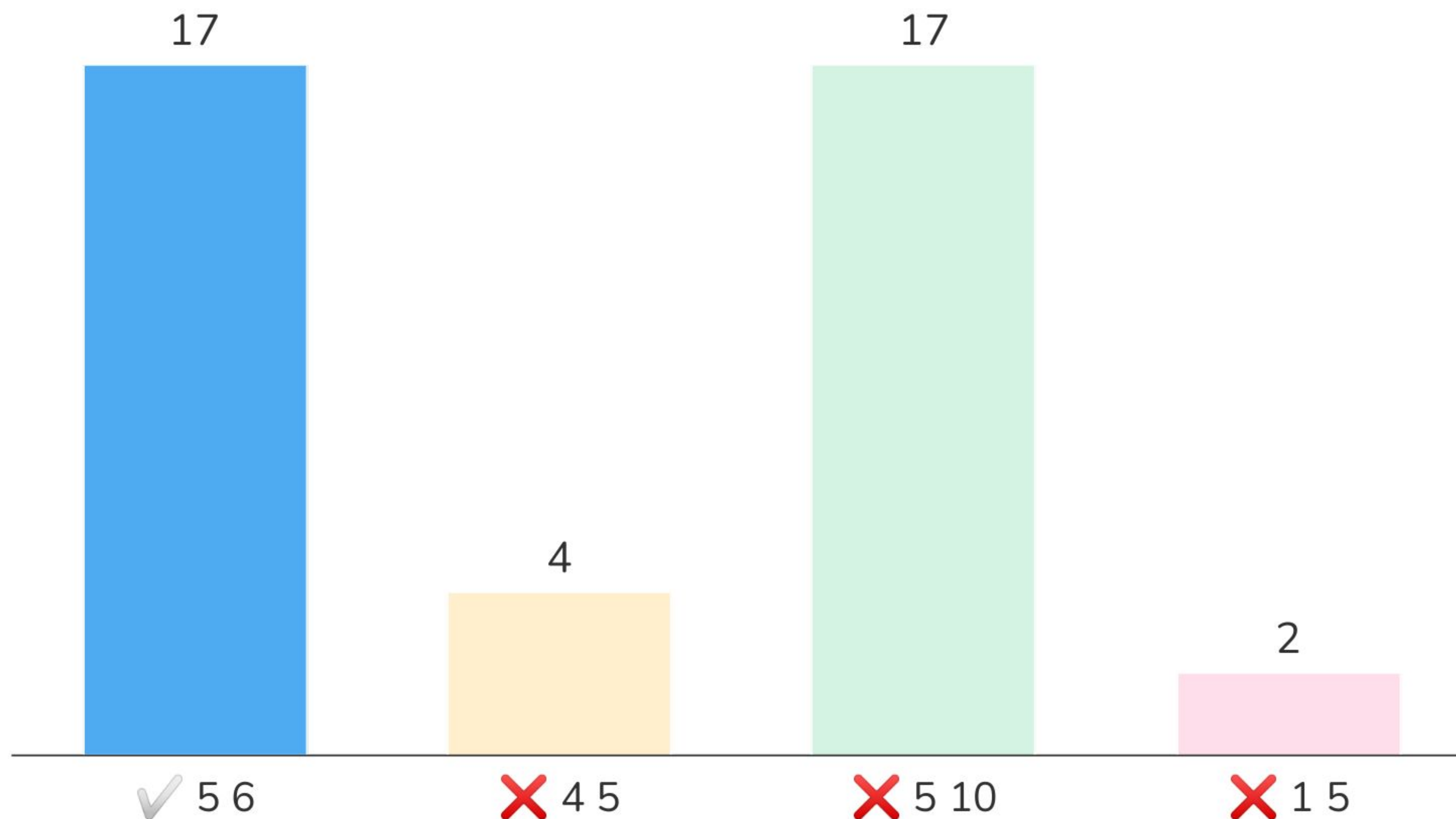


```
package main

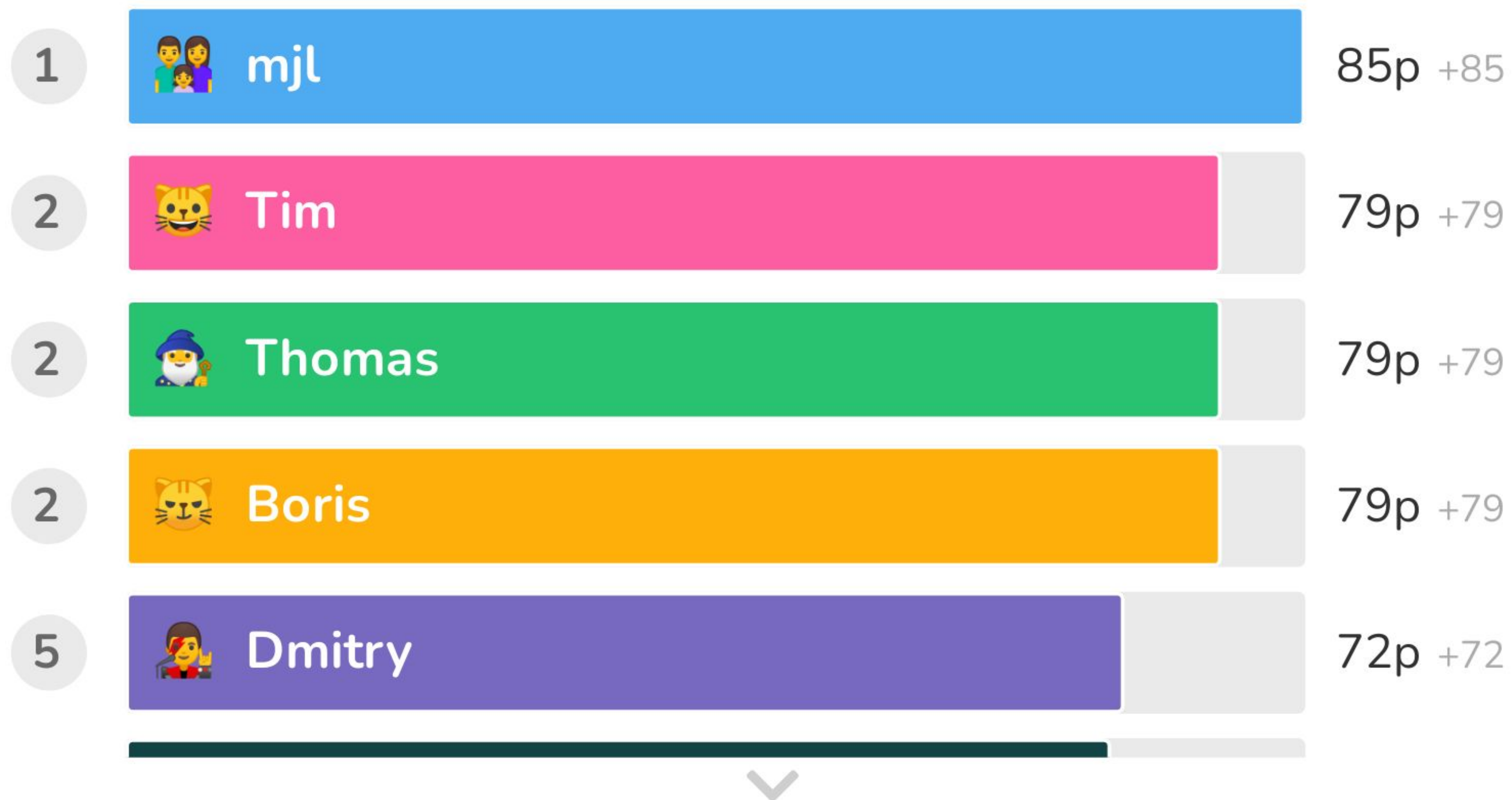
import "fmt"

func main() {
    numbers := []int{1, 2, 3}
    numbers = append(numbers, 4, 5)
    fmt.Println(len(numbers), cap(numbers))
}
```



# Leaderboard

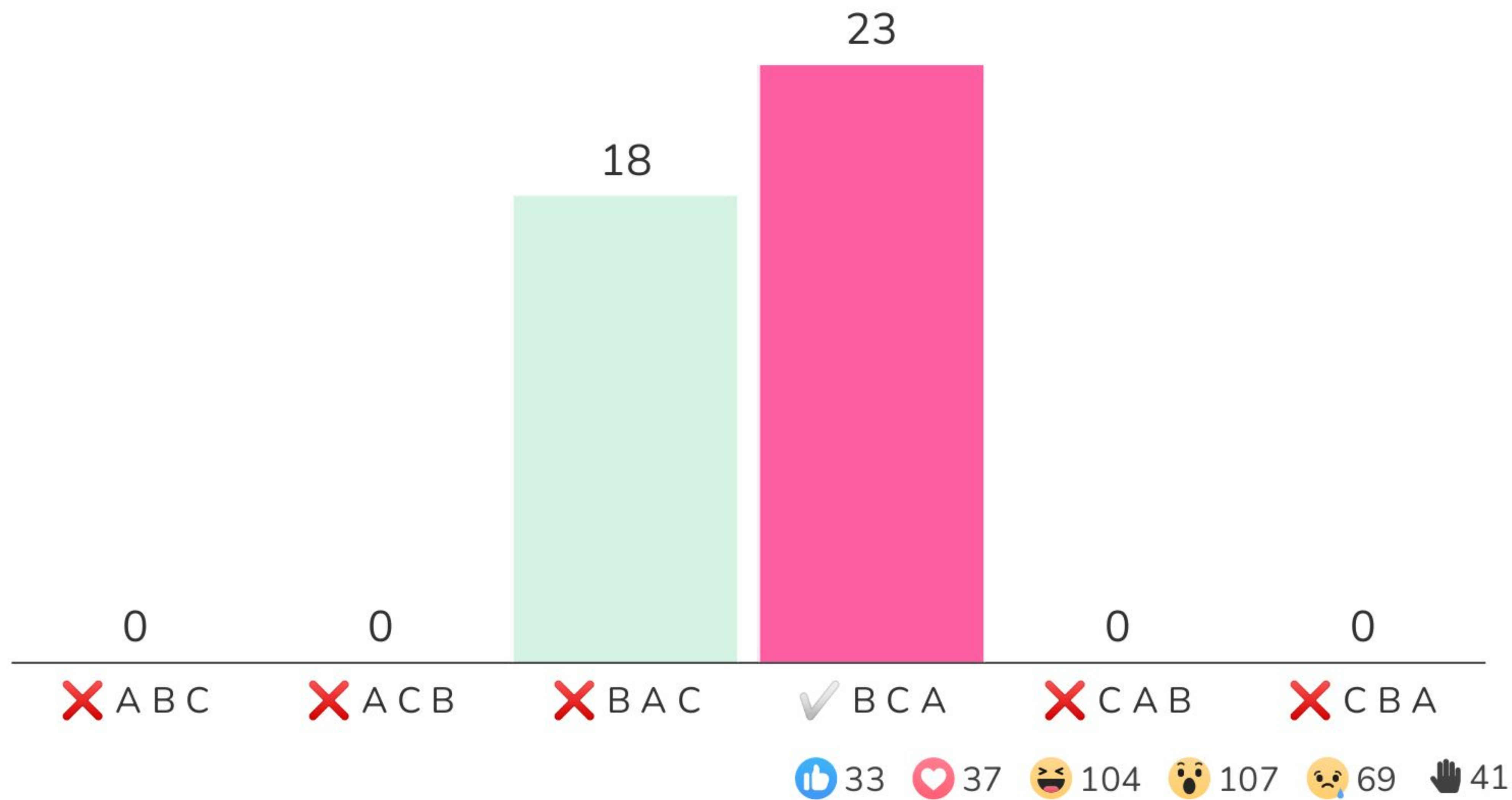
47 players



```
package main|

import "fmt"





func main() {
    defer fmt.Println("A")
    fmt.Println("B")
    defer fmt.Println("C")
}
```





# Leaderboard

47 players

1	 mjl	179p +94
2	 Stephan	163p +92
3	 Thomas	158p +79
4	 Ilya	153p +88
5	 Tim	149p +70



```
package main

import (
    "encoding/json"
    "fmt"
)

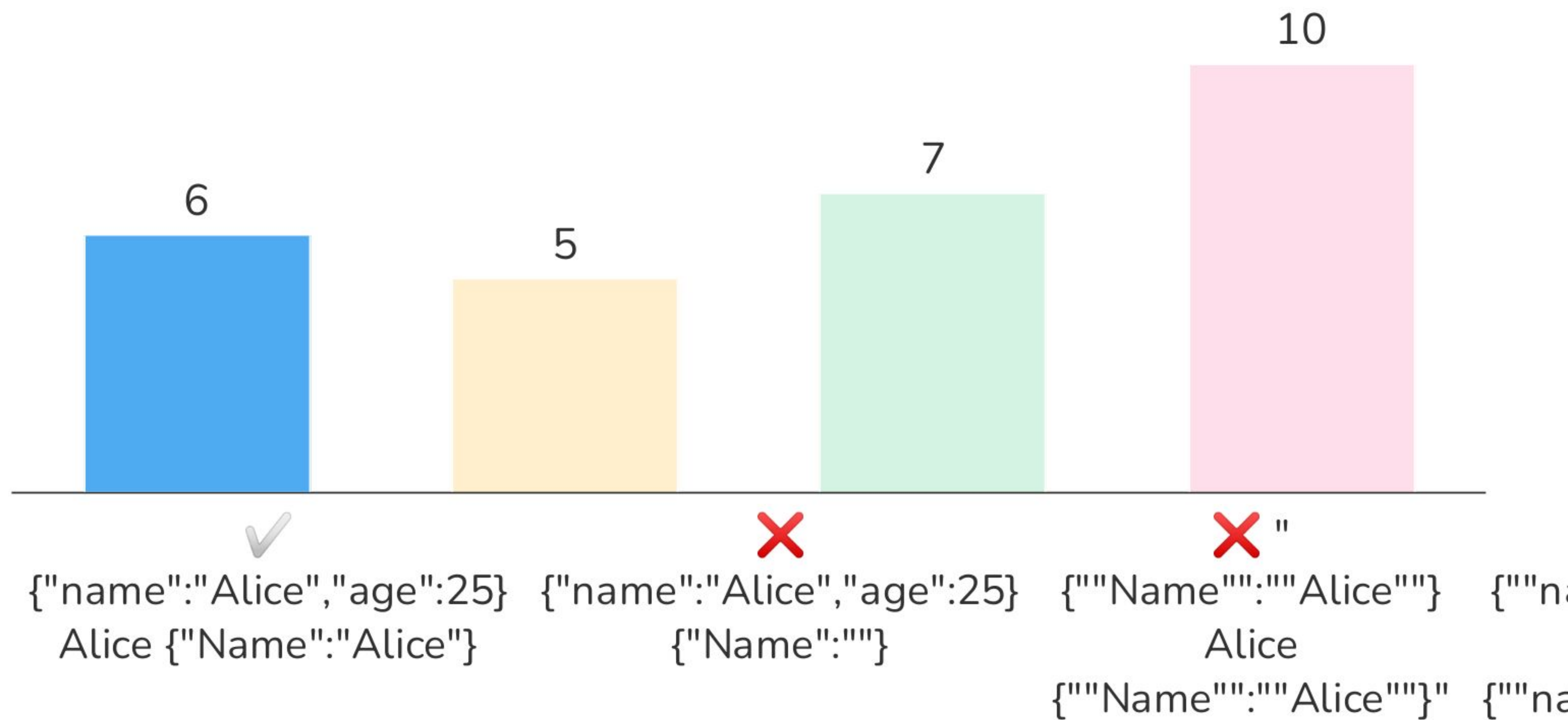
func main() {
    a := &Person{Name: "Alice"}
    data, _ := json.Marshal(a)
    fmt.Println(string(data))

    var b Person
    json.Unmarshal(data, &b)
    fmt.Println(b.Name)

    data, _ = json.Marshal(b)
    fmt.Println(string(data))
}

type Person struct{ Name string }

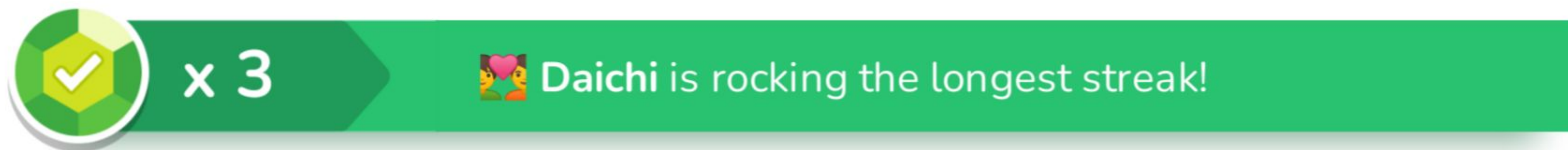
func (p *Person) MarshalJSON() ([]byte, error) {
    return []byte(fmt.Sprintf(`{"name": "%s", "age": 25}`, p.Name)), nil
}
```





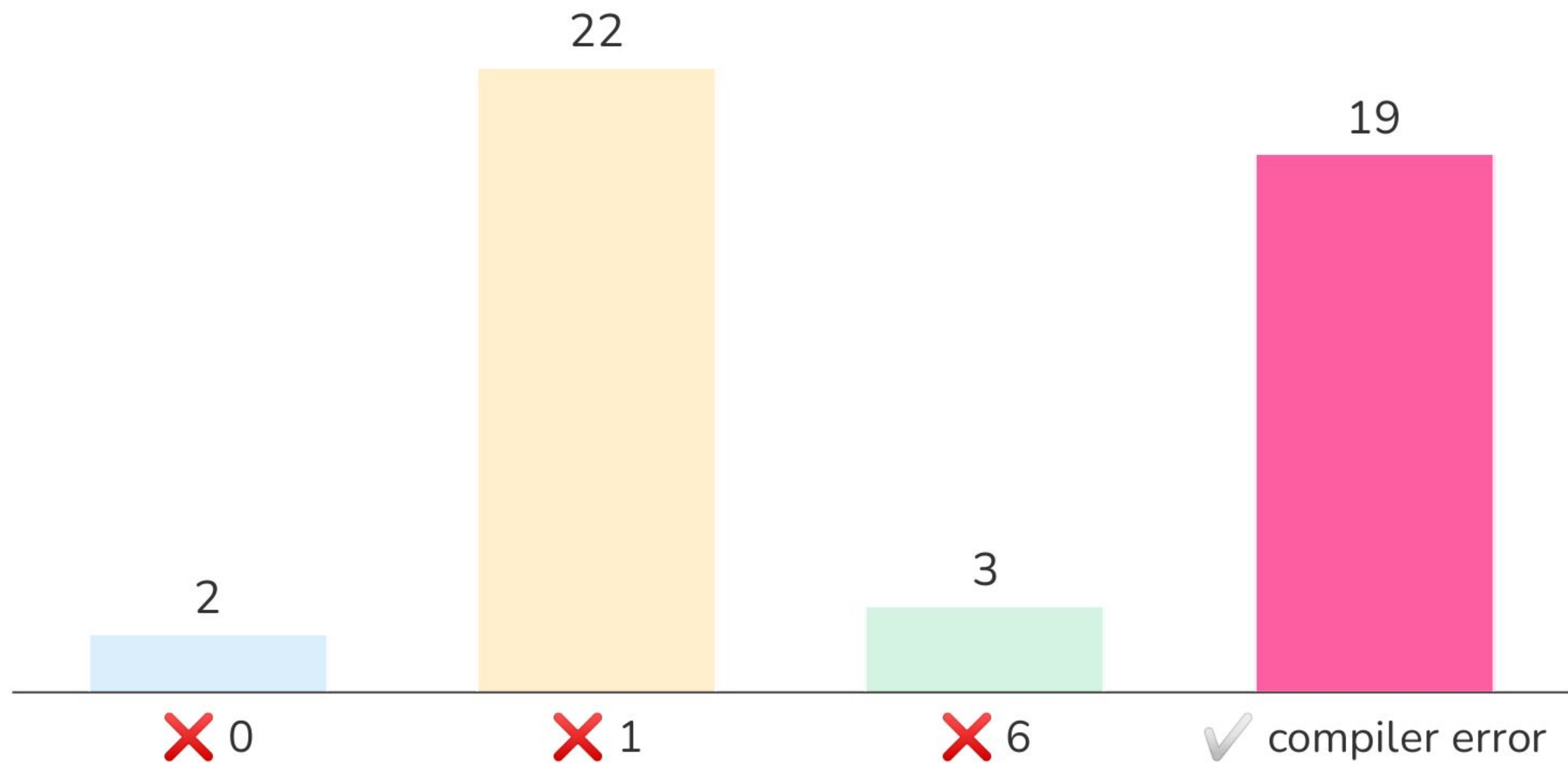
# Leaderboard

47 players



1	 Daichi	202p +57
2	 mjl	179p +0
3	 Stephan	163p +0
4	 Thomas	158p +0
5	 The clown's avatar	150p +0

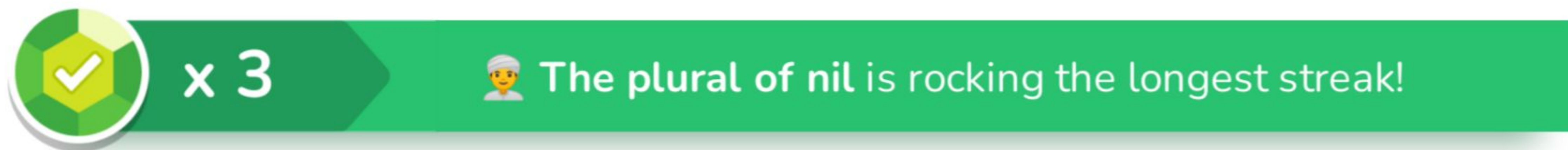
```
m := map[string]int{
    "foo": 1,
}
fmt.Println(cap(m))
```










# Leaderboard

47 players



1	 The plural of nil	241p +85
2	 Mattias	227p +94
3	 Ilya	225p +72
4	 Daichi	202p +0
5	 Tine	188p +0



```

type Response struct {
    Meta struct {
        Title string
    }
}

func main() {
    r := Response{
        Meta: {
            Title: "foo",
        },
    }

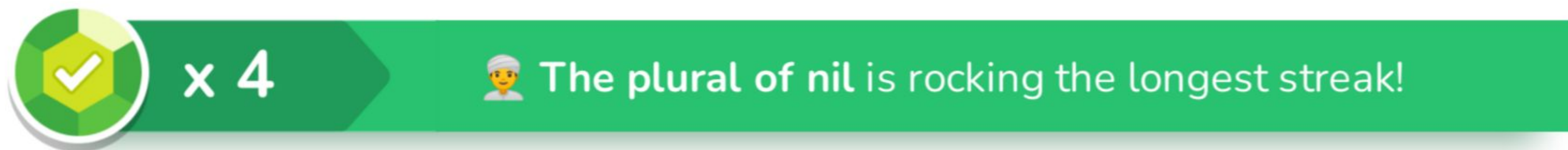
    fmt.Println(r)
}






```



# Leaderboard

47 players

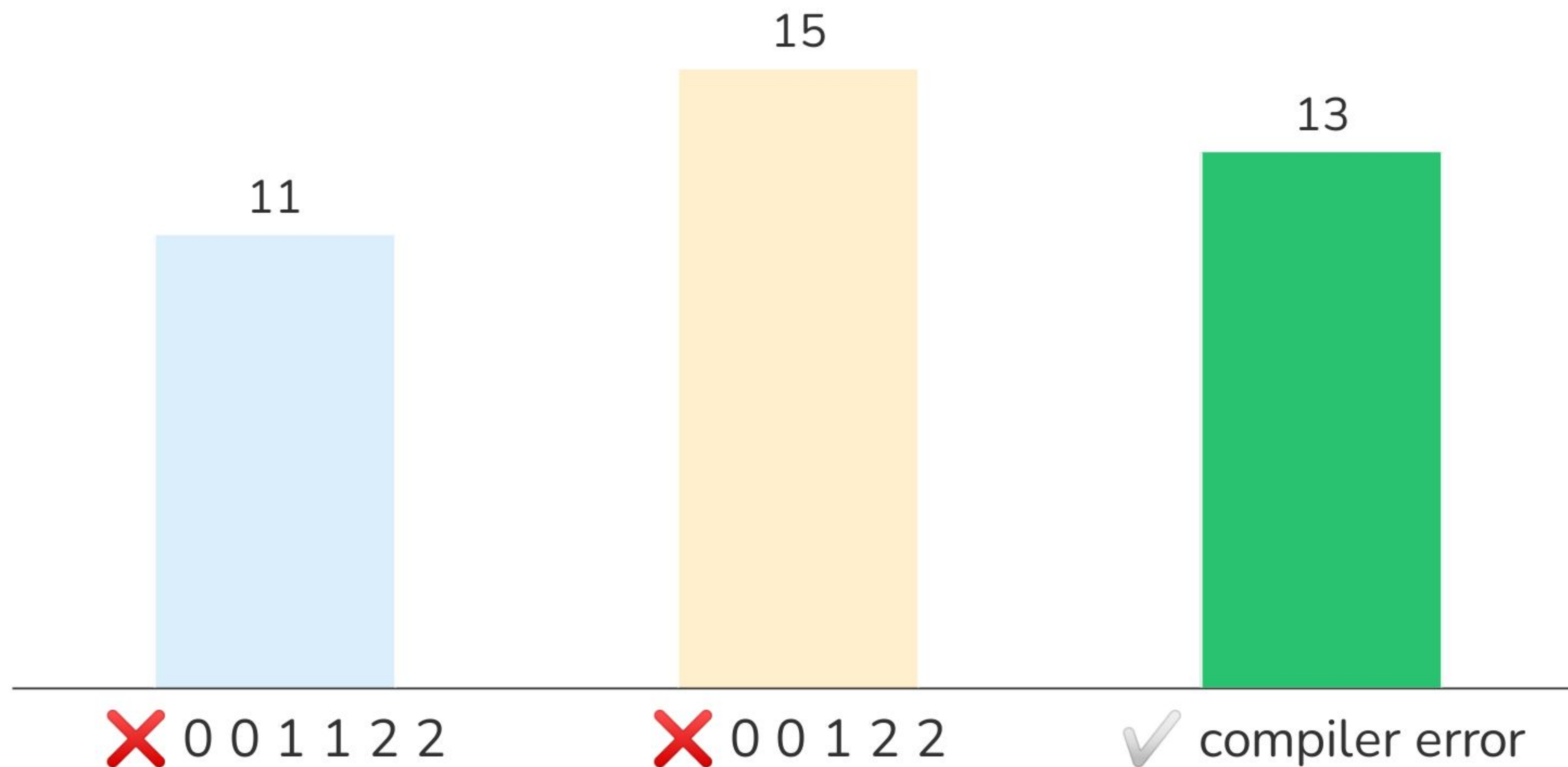


1	 The plural of nil	319p +78
2	 Ilya	287p +62
3	 mjl	273p +94
4	 Daichi	272p +70
5	 Dety	227p +50

```
func main() {
    for i := 0; i < 3; i++ {
        if i == 1 {
            goto anotherblock
        }

        fmt.Println(i)

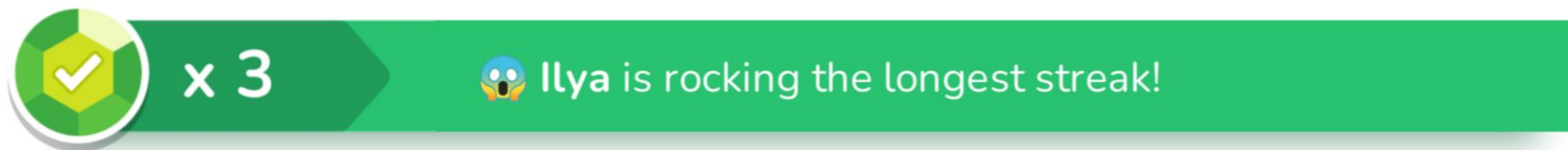
        {
        anotherblock:
            fmt.Println(i)
        }
    }
}
```





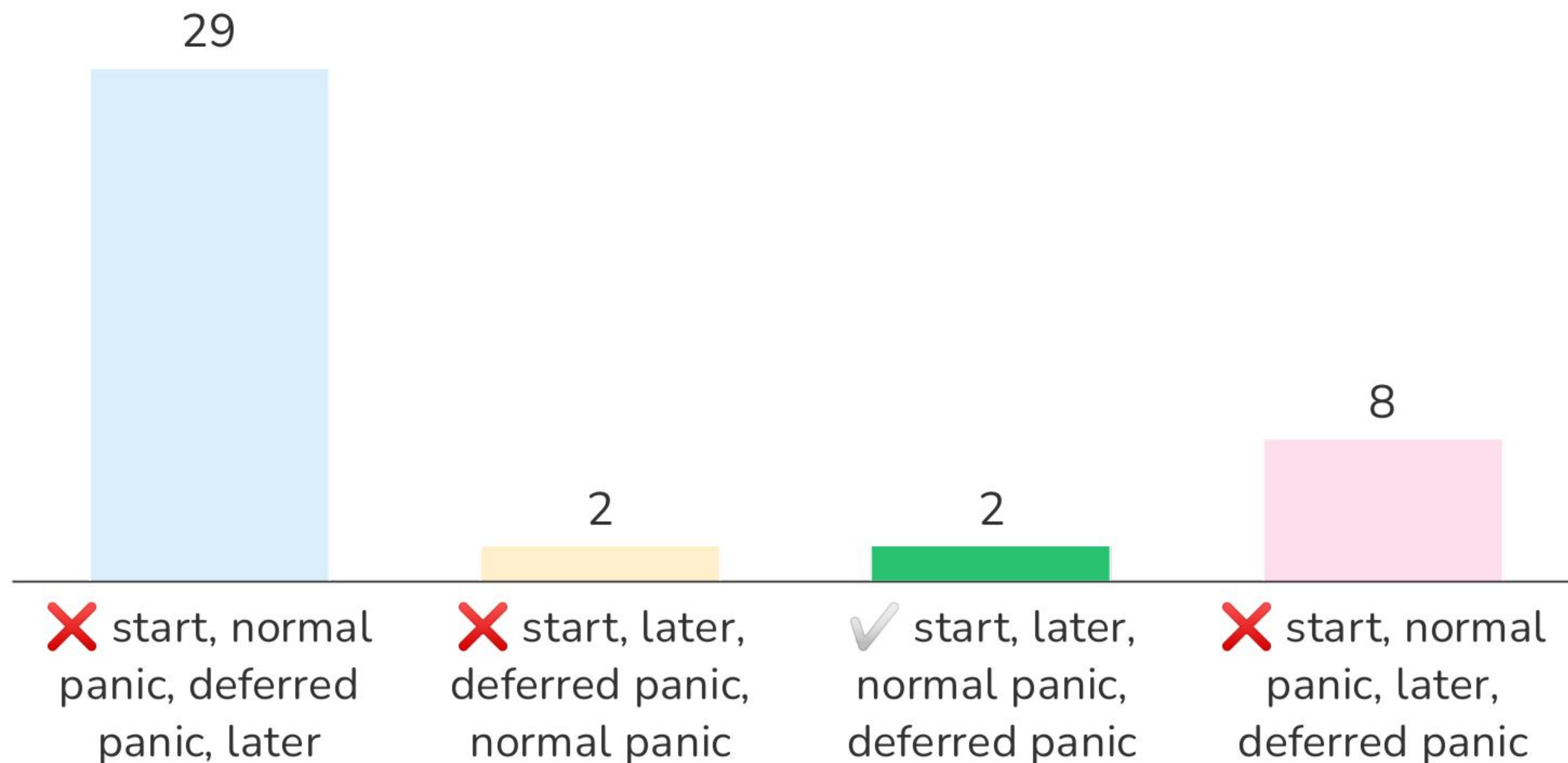
# Leaderboard

47 players



1	 Ilya	341p +54
2	 mjl	340p +67
3	 The plural of nil	319p +0
4	 Petr	293p +56
5	 Dasha	272p +0






```
func main() {
    fmt.Println("start")
    defer fmt.Println("later")
    defer panic("deferred panic")
    panic("normal panic")
}
```








# Leaderboard

47 players

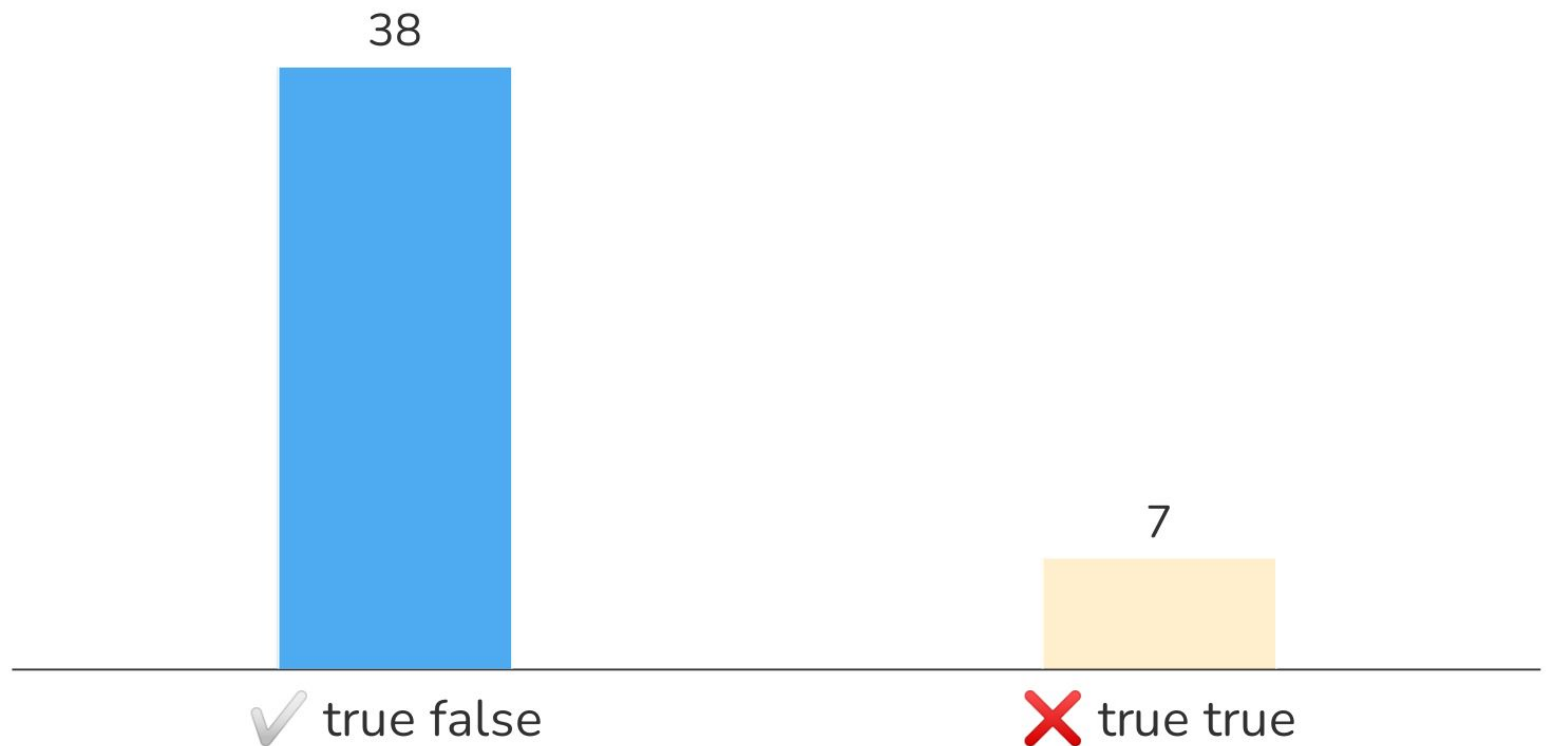
1	 Ilya	341p +0
1	 Daichi	341p +69
3	 mjl	340p +0
4	 The plural of nil	319p +0
5	 Petr	293p +0










```
func main() {  
    var a map[string]int  
  
    fmt.Println(a == nil)  
  
    a = make(map[string]int)  
  
    fmt.Println(a == nil)  
}
```



# Leaderboard

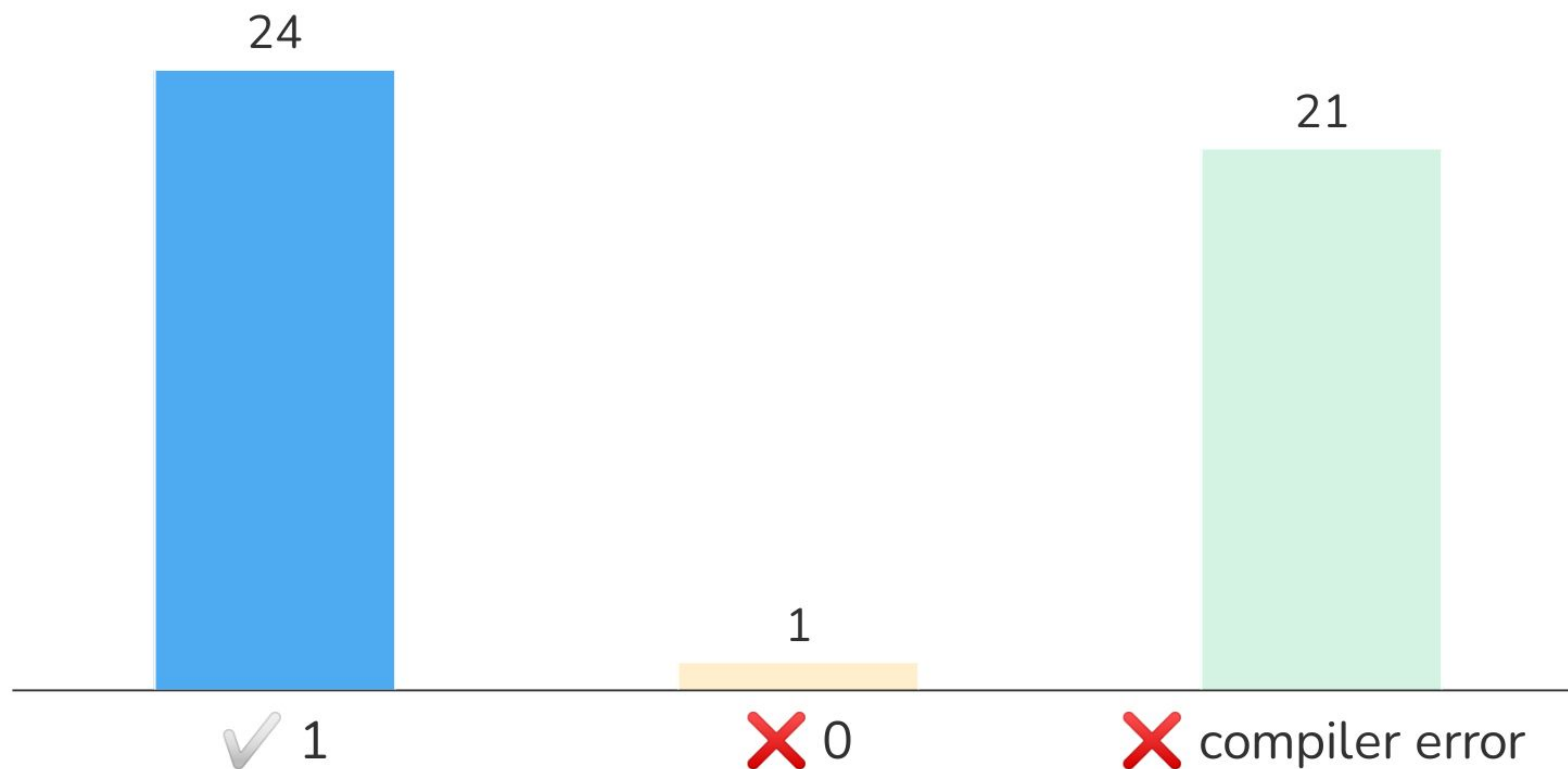
47 players

1	 Ilya	434p +93
2	 mjl	433p +93
3	 The plural of nil	409p +90
4	 Petr	362p +69
5	 Boris	350p +88



```
func main() {
    a := 1






    fmt.Println(*&a)
}
```








# Leaderboard

47 players

1	 Ilya	517p +83
2	 The plural of nil	502p +93
3	 Petr	447p +85
4	 mjl	433p +0
5	 Daichi	429p +88



```
func main() {
    ctx, cancel := context.WithTimeout(context.Background(), time.Millisecond)
    defer cancel()

    time.Sleep(time.Second)

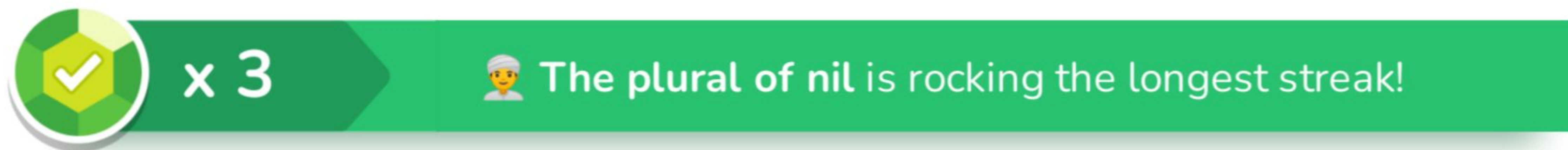
    ctx, cancel = context.WithTimeout(ctx, time.Hour)
    defer cancel()






    fmt.Println(ctx.Err())
}
```



# Leaderboard

47 players



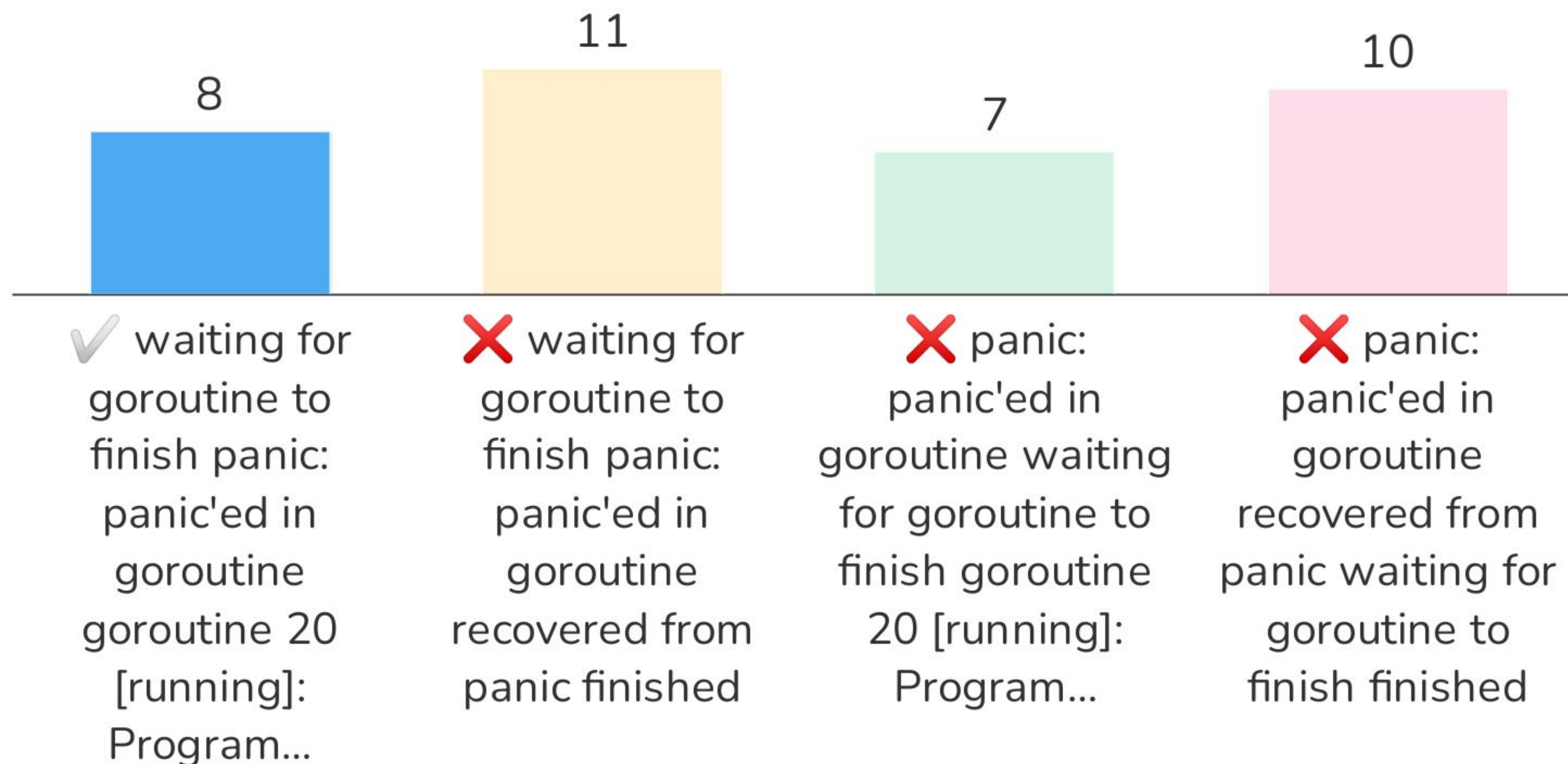
1	 The plural of nil	570p +68
2	 Ilya	517p +0
3	 Herlon	470p +86
4	 Mattias	467p +70
5	 Detu	447p +0



```
func main() {
    defer func() {
        if err := recover(); err != nil {
            log.Println("recovered from panic:", err)
        }
    }()

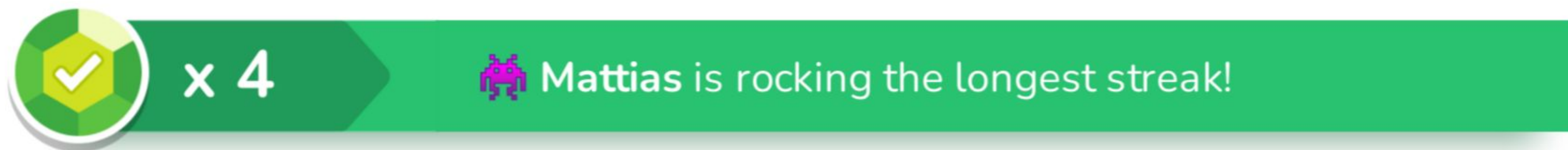
    var wg sync.WaitGroup
    wg.Add(1)
    go func() {
        defer wg.Done()
        time.Sleep(1 * time.Second)
        panic("panic'ed in goroutine")
    }()



    fmt.Println("waiting for goroutine to finish")
    wg.Wait()
    fmt.Println("finished")
}
```



# Leaderboard

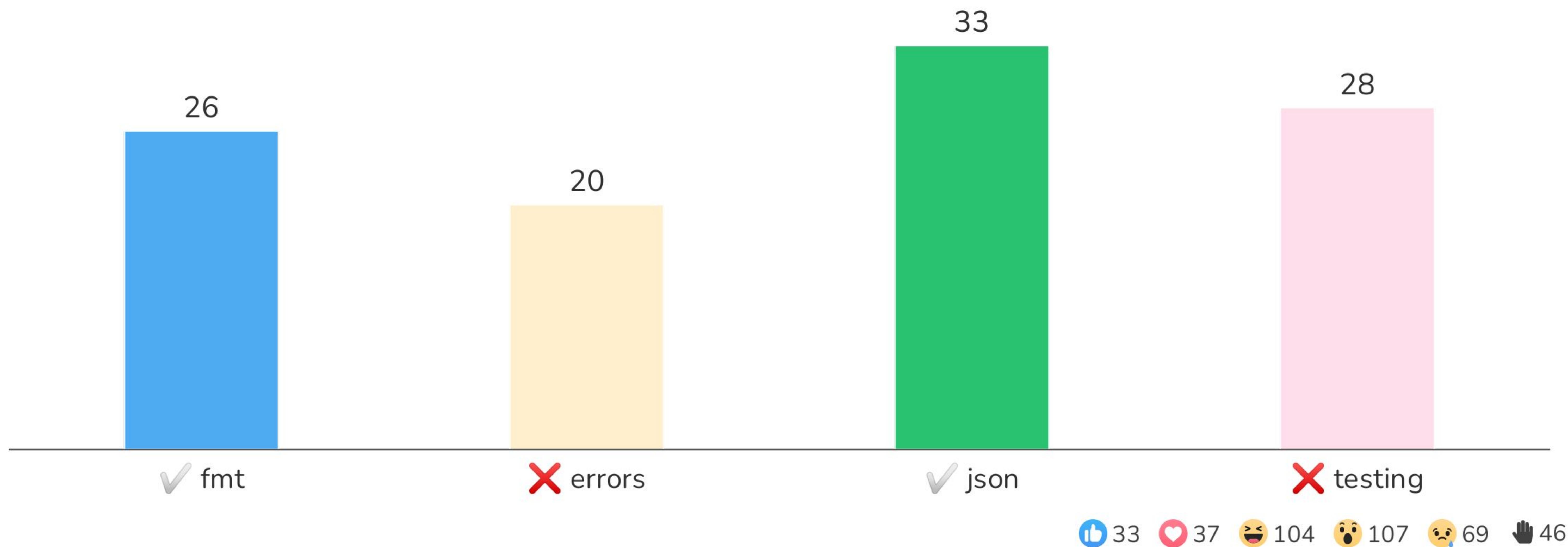
48 players



1	 The plural of nil	570p +0
2	 Mattias	531p +64
3	 Ilya	517p +0
4	 Daichi	490p +61
5	 Harlowe	470p +0



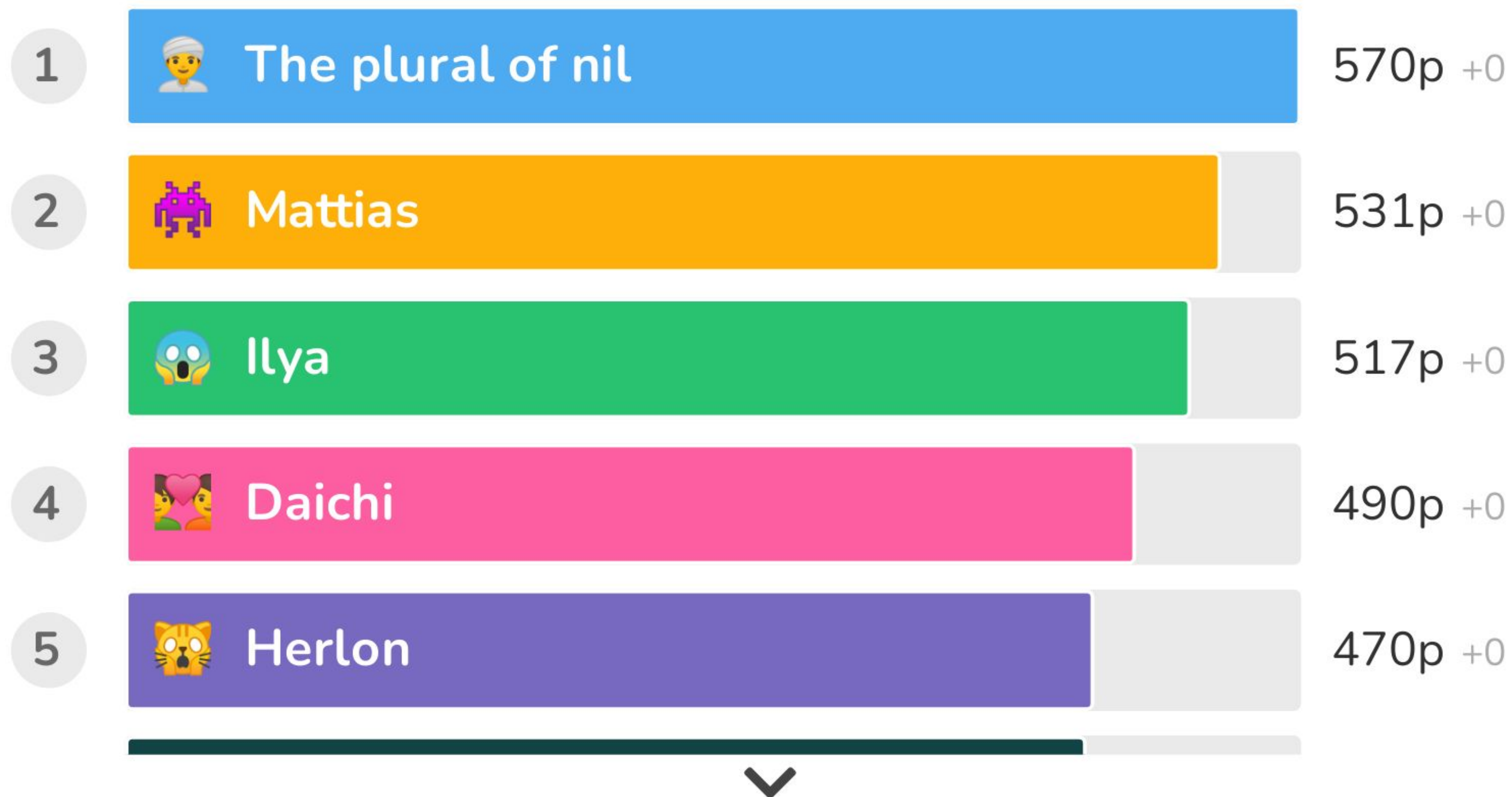
# Which of the following packages import reflect?





# Leaderboard

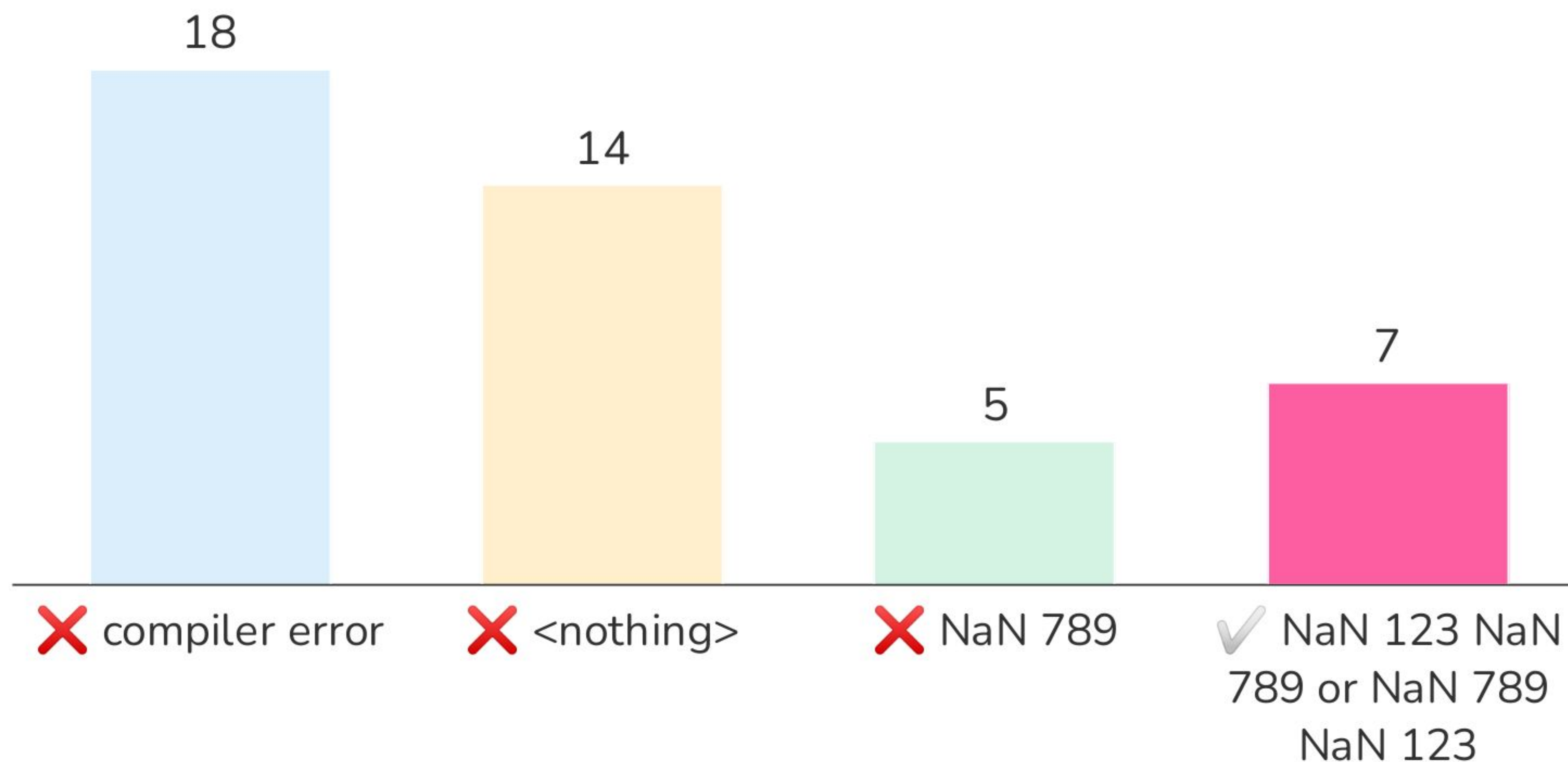
48 players



```
var a = math.NaN()
var m = map[float64]int{}






m[a] = 123
m[a] = 789
delete(m, a)

for k, v := range m {
    fmt.Println(k, v)
}
```



# Leaderboard

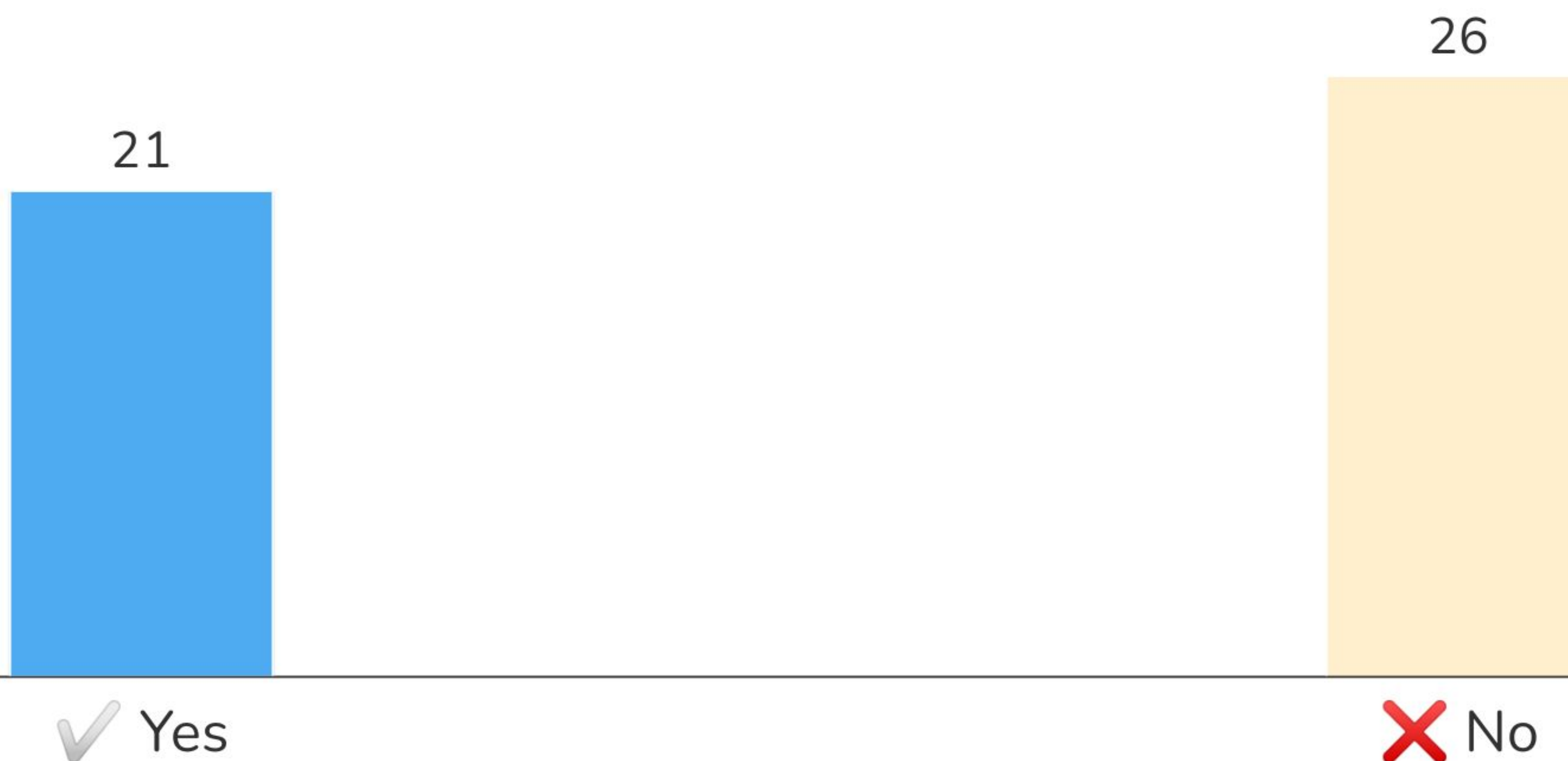
48 players

1	 The plural of nil	642p +72
2	 Mattias	598p +67
3	 Ilya	517p +0
4	 mjl	507p +74
5	 Daichi	490p +0



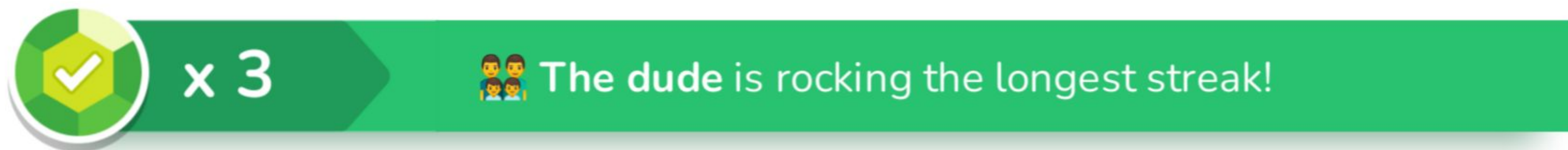







Can you create and read files in the <https://play.golang.org/> sandbox?



# Leaderboard

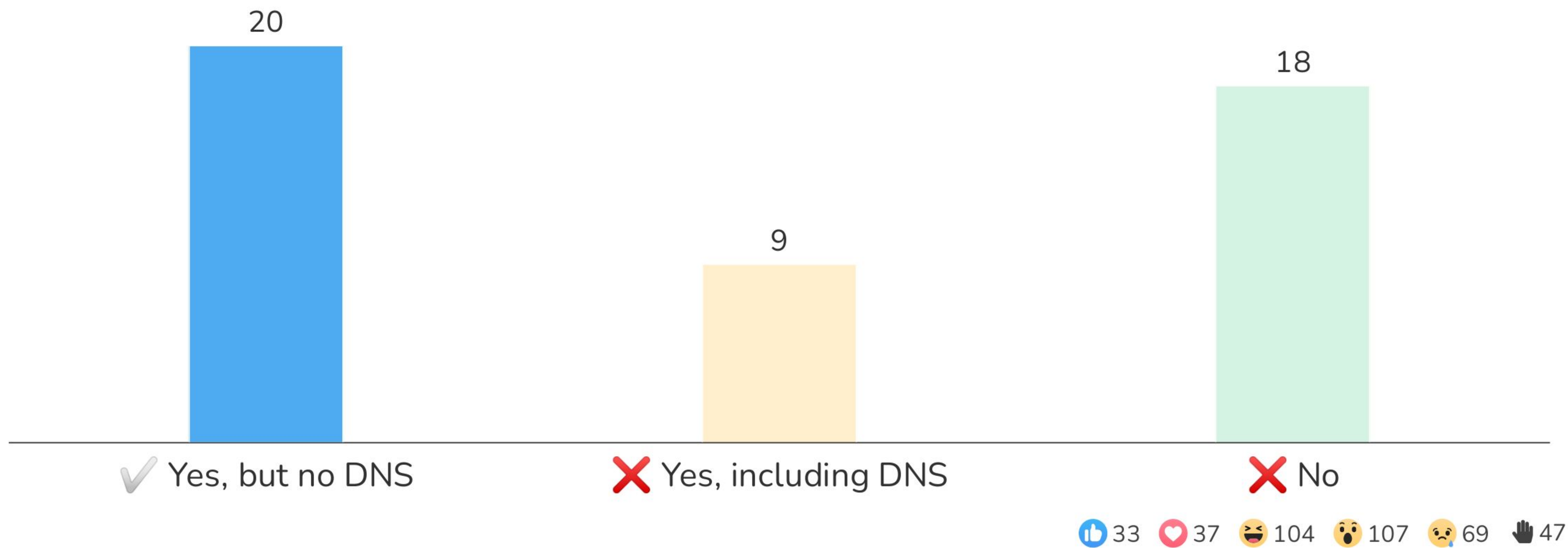
48 players



1	 The plural of nil	739p +97
2	 Mattias	695p +97
3	 Ilya	613p +96
4	 mjl	604p +97
5	 Didi	570p +97



Is TCP allowed in the <https://play.golang.org/> sandbox?





# Leaderboard

48 players



x 4



The plural of nil holds the longest streak record!



## The plural of nil

1



The plural of nil



827p +88

2



Mattias



788p +93

3



Stephan

## 827 points

618p +93

4



Ilya

613p +0

5



nil

604p +0