

# Tipos de Dados e Tipos Opcionais

Faça um jogo de adivinhar o número. O jogo tem dois jogadores. O **Jogador 1** indica um número a ser adivinhado. Depois disso, esse número some da tela e o **Jogador 2** tem três chances para adivinhar o número escolhido. Para cada tentativa, o jogo deve dizer ao jogador se o número que ele forneceu é maior, menor ou igual ao correto.

Ler entrada do teclado: `readLine()`

```
let idade = readLine()  
print("Você tem quase \(idade + 1) anos")
```



Binary operator '+' cannot be applied to operands of type 'String?' and 'Int'

# Tipos de dados

Números inteiros (Int):

1, 2, 3, 4, 53546, 10000000

Números de ponto flutuante (Float e Double):

0.4, 1.6E3, 1.0, 3.000000243, 0xFp2

Texto (Strings)

"abc", "Fernando", "1", "4546546456", "True"

Booleanos (Bool)

true, false

# Relembrando: Vários tipos de expressões e operadores

**Aritméticos**: produzem números (inteiros, de ponto flutuante, etc.) como resultado:

+   -   \*   /   %   pow

De **comparação e lógicos**: produzem valores do tipo verdadeiro ou falso como resultado:

==   !=   <   <=   >   >=

&&   ||   !

# Conversão de Tipos

```
let idade = readLine()
```

```
print("Você tem quase \ (idade + 1) anos")
```

# Conversão de Tipos

```
let idade = readLine()  
let idadeNumerica = Int(idade) //cria um novo Int  
print("Você tem quase \ (idadeNumerica + 1) anos")
```



# Tipos Opcionais

```
//o usuário digitou o palpite "Zagreb"  
var palpite = readLine()  
if palpite == respostaP1 {  
    print(palpite) //imprime Optional("Zagreb")  
}
```

`readLine()` devolve um `String`?

(isso não é uma pergunta!)

# Mais Tipos Opcionais

```
let a = Int("123")  
print(a) //imprime Optional(123)
```

# Mais Tipos Opcionais

```
let a: Int? = Int("123")  
print(a) //imprime Optional("123")
```



# Tipos opcionais

let a: String?

String

nil





# O que é nil?

```
let a:String = "Garantidamente um String"  
print(a.uppercased())
```

# O que é nil?

```
let a:String? = readLine()  
print(a.uppercased()) // não funciona!
```

# O que é nil?

"abc".uppercased() → "ABC"



# O que é nil?

"abc".uppercased() → "ABC"

"".uppercased() → ""

# O que é nil?

`"abc".uppercased()` → `"ABC"`

`"".uppercased()` → `""`

`nil.uppercased()` →

`MyPlayground.playground/: error: Playground execution aborted: error: Execution was interrupted, reason: EXC_BAD_INSTRUCTION (code=EXC_I386_INVOP, subcode=0x0).`

Int("123")

Int("10\*10")

Float("0xFp0")

Int("0xFp0")

Float("1")

Double("42.42")

# Desempacotamento forçado

```
let nome = readLine()!  
let nomeOpt = nome!  
print(nomeOpt.uppercased())
```

# Ligação de Opcional (*Optional Binding*)

```
let nome = readLine()  
if let nomeOpt = nome {  
    print(nomeOpt.uppercased())  
}
```

# Voltando ao joguinho

```
... // outras coisas ocorreram antes
palpite = readLine()
if let x = palpite {
    if let p = Int(x), p < numeroSecreto {
        print("O número secreto é maior que seu palpite!")
    }
}
```

# Voltando ao joguinho, com emoção


```
... // outras coisas ocorreram antes
palpite = readLine()
let p = palpite as! String
if p < numeroSecreto {
    print("O número secreto é maior que seu palpite!")
}
```

Tópico fortemente relacionado:  
*Optional Chaining*  
(encadeamento de opcionais)



# Encadeamento de Opcionais

```
let a:String? = readLine()  
print(a?.uppercased()) // desempacota
```



Se o usuário digitou o String "abc", devolve "abc".

Se você atingiu EOF (ctrl+D), devolve nil.

# Desempacotamento Implícito

```
let a:String! = readLine()  
print(a.uppercased())
```

# O que é nil?

```
let a:String! = nil  
print(a.uppercased())
```

```
MyPlayground.playground/: error:  
Playground execution aborted: error:  
Execution was interrupted, reason:  
EXC_BAD_INSTRUCTION (code=EXC_I386_INVOP,  
subcode=0x0).
```

Reveja seu jogo de adivinhar o  
número para que trabalhe com  
Ints, ao invés de Strings

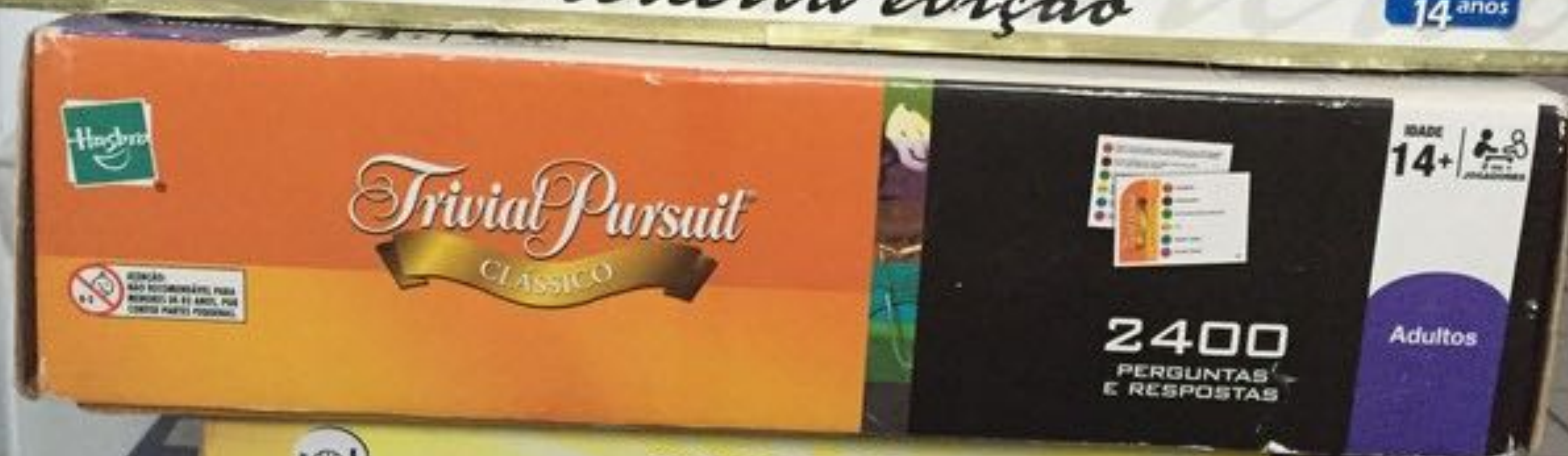
Faça um programa que lê um número entre 100 e 999 a partir do teclado e verifica se ele é divisível por 3.

Faça um programa que lê três números a partir do teclado e imprime os três em ordem crescente.

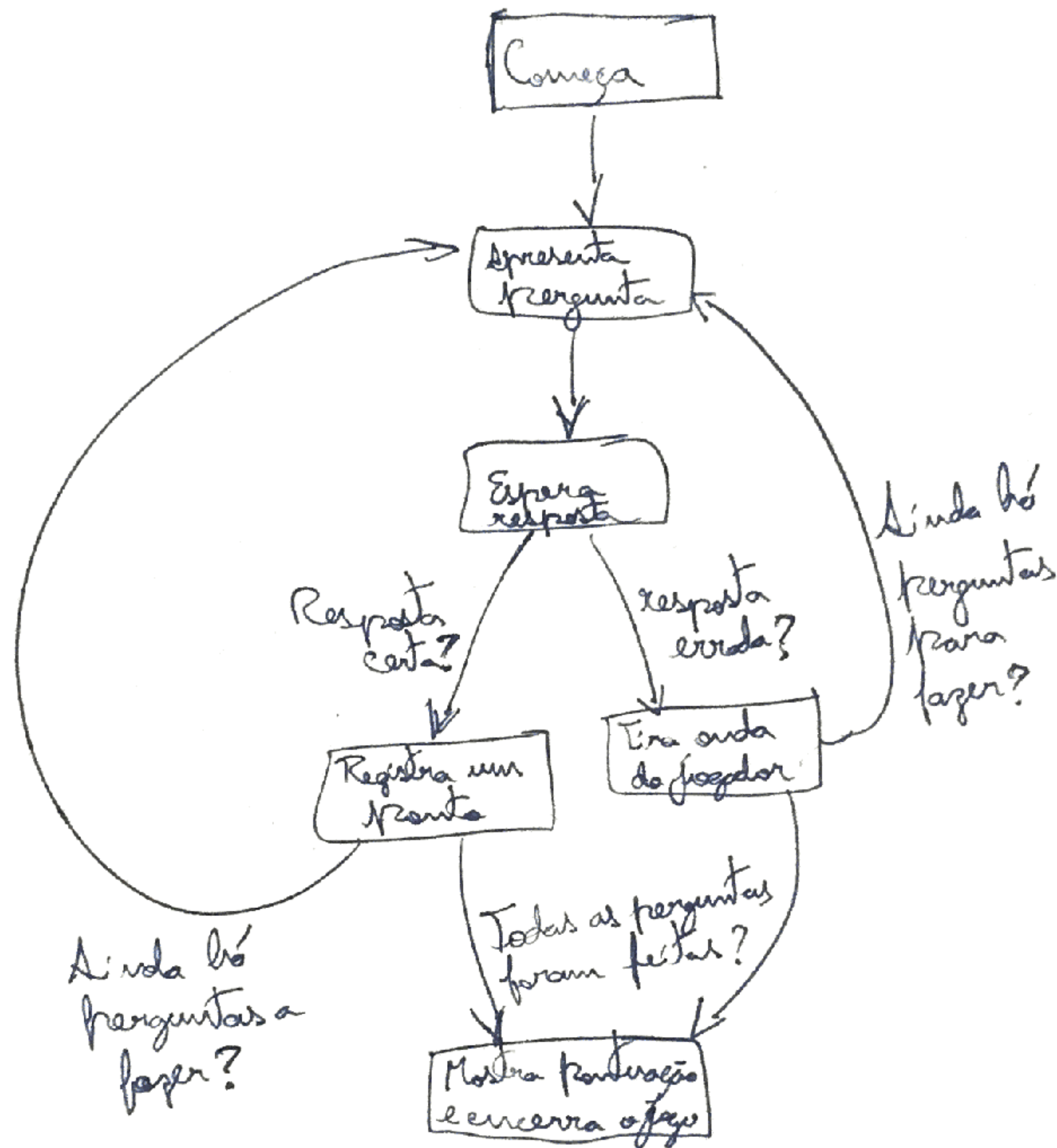
Faça uma calculadora de 4 operações que recebe o operador (+, -, \* ou /) primeiro, os dois operandos depois, realiza a operação desejada e imprime o resultado.

Modifique sua calculadora para que o programa termine caso um operador ou operando inválido seja fornecido. Forneça também uma mensagem de erro para o usuário.









# Implemente o jogo de Trivia em Swift

Implemente um jogo de Forca  
em Swift