

Iteração/*Loops*/Laços

Como você limpava a tela no
programa de adivinhar o número?

Laços (ou *loops*)

Executam um comando enquanto uma **condição** for verdadeira

Similares a comandos `if`, só que executam **várias vezes**

```
...  
print()  
print()  
print()  
print()  
print()  
print()  
print()  
print()
```

```
print("Forneça um palpite:")  
let palpite = readLine()!  
if let palpiteNum = Int(palpite) {  
    ...  
}
```

Forma geral do comando while

```
while condição {  
    comandos1  
}  
comandos2
```

```
...  
var numLinhas = 50  
while numLinhas > 0 {  
    print()  
    numLinhas = numLinhas - 1  
}  
  
print("Forneça um palpite:")  
let p = readLine()  
if let palpiteNum = p, palpite = Int(palpiteNum) {  
    ...  
}
```

Condição

...

var numLinhas = 50

while numLinhas > 0 {

print()

numLinhas = numLinhas - 1

}

Corpo

print("Forneça um palpite:")

let p = readLine()

if let palpiteNum = p, palpite = Int(palpiteNum) {

...

}

Write a program that prints the numbers from 1 to 100. But for multiples of three print “Fizz” instead of the number and for the multiples of five print “Buzz”. For numbers which are multiples of both three and five print “FizzBuzz”

Construa um programa que fica pedindo que o usuário digite números e os soma até que o usuário digite 0. Neste momento o programa deve imprimir o valor total da soma.

Crie um programa que lê um número N do teclado e calcula o seu fatorial.

```
print("Calcular o fatorial de qual número?")
var numero = readLine()

if let ns = numero, let n = Int(ns) {
    var fat = 1
    while n > 0 {
        fat = fat * n
        n = n - 1
    }
    print(fat)
}
```

Faça um programa que, dado um número lido a partir do teclado, calcula a soma de seus dígitos (independentemente do número de dígitos)

Construa um programa que lê um número X a partir do teclado e verifica se ele é primo. Seu programa deve informar ao usuário se X é primo ou não.

```
var nStr = readLine()
if let ns = nStr, let n = Int(ns) {
    var i = 2
    var ehPrimo:Bool = true
    if n > 2 {
        while i < n && ehPrimo == true {
            if n % i == 0 {
                ehPrimo = false
            }
            i = i + 1
        }
    }
    if ehPrimo {
        print("\n) é um número primo")
    } else {
        print("\n) NÃO é um número primo")
    }
}
```

Voltando ao código do programa
de adivinhar o número

Faça com que seu programa só aceite números a ser adivinhados entre 0 e 100. Enquanto os números não estiverem nesse intervalo, seu jogo deve continuar repetidamente pedindo que o número a ser adivinhado seja fornecido.

Faça com que a parte que pede os palpites e checa se estão corretos também rode em um laço, de modo a evitar repetição de código

Depois disso, modifique o programa para que peça 10 palpites, ao invés de 3

Voltando ao jogo de adivinhar o número, modifique-o para que, uma vez que o jogador acerte o número, o laço termine imediatamente, independentemente do número máximo de palpites

```
while numPalpites < limitePalpites {  
    print("Dê um palpite.")  
    var palpiteStr = readLine()  
  
    if let p = palpiteStr, let palpite = Int(p) {  
        if palpite < numeroAAdivinhar {  
            print("Seu palpite é menor...")  
        } else if palpite > numeroAAdivinhar {  
            print("Seu palpite é maior...")  
        } else {  
            print("ACERTOU.")  
            acertou = true  
            break  
        }  
    }  
}  
  
    numPalpites = numPalpites + 1  
}
```

```
...  
while !acertou && numPalpites < limitePalpites {  
    print("Dê um palpite.")  
    var palpiteStr = readLine()  
    ...  
}
```

Construa um programa que lê um número inteiro N a partir do teclado e calcula a raiz cúbica **inteira** desse número, se houver. Caso não haja, ele deve informar isso ao usuário.

A sequência de Fibonacci é dada por

$$fib(0) = 1$$

$$fib(1) = 1$$

$$fib(n) = fib(n-1) + fib(n-2)$$

Faça um programa que, dado um número n , calcula o $fib(n)$.

Construa um programa que lê um número N a partir do teclado e calcula todos os números primos entre 2 e N

A solução usa **laços aninhados**

Faça seu programa imprimir todos os divisores de todos os números de 1 a um N lido a partir do teclado. Você não precisa levar em conta nem o 1 nem o próprio número como divisores. Apresente essa informação para o usuário **após** dizer para ele que o número não é primo.

Dado um número N , fornecido pelo usuário, calcule todas as médias aritméticas dos números entre 1 e j , para todo número j tal que $j \leq N$.

Construa uma variante do jogo de adivinhar o número que funciona da seguinte maneira. O número inteiro a ser adivinhado N pode estar **entre 1 e 200**. Palpites podem ter apenas três valores: 1, 10 e 100 (o jogo não deve aceitar valores diferentes). Além disso, o jogo mantém uma variável S **inicialmente igual a 0**. Quando o segundo jogador fornece um palpite, um dos dois cenários ocorre:

(1) S é menor que o número a ser adivinhado e o novo S é o resultado de **somar** o novo palpite ao S antigo

(2) S é maior que o número a ser adivinhado e o novo S é o resultado de **subtrair** o novo palpite do S antigo

Se, após essas operações, o valor de S for igual ao número a ser adivinhado, o jogo acaba. Caso contrário, prossegue conforme as mesmas regras. O jogo deve contar o número de palpites que foram necessários para acertar e imprimir esse número no final.

Crie um programa que constrói uma tabela de multiplicação. Por exemplo, para um $N = 4$, a tabela de multiplicação correspondente é a seguinte:

	1	2	3	4
1	1	2	3	4
2	2	4	6	8
3	3	6	9	12
4	4	8	12	16

Note que, em uma célula qualquer (x,y) , o valor armazenado é igual a $x * y$. Seu programa deve receber o valor de N como entrada e produzir uma tabela como essa, organizada. Não se preocupe, porém, se o valor de N for grande demais para apresentar a tabela de forma arrumada.

Faça um programa que, dado um raio R ,
desenha apenas usando asteriscos um círculo
cujo raio é R caracteres.