

```

//
// main.swift
// NanoChallenge-4
//
// Created by Pedro Henrique Dias Hemmel de Oliveira Souza on
24/05/22.
//

import Foundation

//Criando funções do código

//Função que tem como objetivo pular uma linha no código, parametro
passado para determinar o tamanho do espaço
func quebraLinha(qtdVezes: Int) {
    for _ in 1...qtdVezes {
        print()
    }
}

//Função que tem como objetivo ensinar alguns conceitos de laços de
repetições
func ensinaLacoRepeticao() {
    usleep(1000000)
    quebraLinha(qtdVezes: 1)
    print("Laço de Repetição é uma instrução de fluxo de controle
que permite que o código seja executado repetidamente com base em
uma determinada condição atribuída ao loop.")

    usleep(3000000)
    quebraLinha(qtdVezes: 1)
    print("Mas...")

    usleep(1000000)
    quebraLinha(qtdVezes: 1)
    print("Para que devemos usar laços de repetição?")

    usleep(2000000)
    quebraLinha(qtdVezes: 4)
    print("Vamos supor que você virou cantor e ainda está no começo,
3 pessoas vão para o seu show, você quer saber qual é a idade média
do público que te assiste. O que normalmente as pessoas rapidamente
fazem quando se deparam com esse problema, é solicitar a idade de
cada um, então vamos lá.")

    usleep(5000000)
    quebraLinha(qtdVezes: 1)
    //Laço de repetição que pergunta 3 solicitações para o usuário
    for i in 1...3 {
        usleep(1000000)
        quebraLinha(qtdVezes: 1)
        print("Digite (1) para solicitar a idade da " + String(i) +
"° pessoa.")
        //Tratativa para caso o usuário digite algo diferente de 1

```

```

        while(readLine() != "1") {
            quebraLinha(qtdVezes: 1)
            print("Digite um número válido para continuar. Digite
(1) para solicitar a idade da " + String(i) + "° pessoa.")
        }
    }

    usleep(3000000)
    quebraLinha(qtdVezes: 2)
    print("Que legal! conseguimos pegar a idade de cada pessoa e
fazer a média de idade do seu público que é de 23 anos!")

    usleep(3000000)
    quebraLinha(qtdVezes: 4)
    print("Temos novidades, você está ficando famoso, e agora a
quantidade de pessoas que vai ao seu show é de 100 pessoas!")

    usleep(2000000)
    quebraLinha(qtdVezes: 1)
    print("Você ainda quer saber a idade média das pessoas que te
assistem, mas é perceptível que perguntar para cada uma das 100
pessoas seria muito demorado e cansativo. É nesse momento que o laço
de repetição entra em ação, você decide criar um programa que
utiliza loop, e enquanto não tiver perguntado para todo mundo que te
assiste, o programa continuará perguntando.")

    usleep(5000000)
    quebraLinha(qtdVezes: 1)
    print("Digite (1) implementar o programa com laço de
repetição.")

    //Tratativa para caso o usuário digite algo diferente de 1
    while(readLine() != "1") {
        quebraLinha(qtdVezes: 1)
        print("Digite um número válido para continuar. Digite (1)
implementar o programa com laço de repetição.")
    }

    quebraLinha(qtdVezes: 2)
    //Laço de repetição que demonstra como seria o programa de forma
autônoma perguntando as idades para as pessoas
    for i in 1...10 {
        if(i < 4) {
            print("Acabou de ser perguntado a idade da " + String(i)
+ "° pessoa.")
            quebraLinha(qtdVezes: 1)
            usleep(1000000)
        } else if(i >= 4 && i <= 9) {
            print(".")
            quebraLinha(qtdVezes: 1)
            usleep(1000000)
        } else {
            print("Acabou de ser perguntado a idade da 100°
pessoa.")
        }
    }

```

```

        quebraLinha(qtdVezes: 1)
        usleep(1000000)
    }
}

usleep(3000000)
quebraLinha(qtdVezes: 2)
print("Que legal! conseguimos pegar a idade de cada pessoa e
fazer a média de idade do seu público que é de 29 anos!")

usleep(3000000)
quebraLinha(qtdVezes: 1)
print("Viu como é importante e essencial utilizarmos laço de
repetição? Espero você no próximo tópico!")

usleep(3000000)
}

//Função que ensina conceitos de estruturas condicionais
func ensinaCondicionais() {

    usleep(1000000)
    quebraLinha(qtdVezes: 1)
    print("A Estrutura Condicional possibilita a escolha de um grupo
de ações e estruturas a serem executadas quando determinadas
condições são ou não satisfeitas. A Estrutura Condicional pode ser
Simples ou Composta.")

    usleep(3000000)
    quebraLinha(qtdVezes: 2)
    print("A Estrutura Condicional Simples executa um comando ou
vários comandos se a condição for verdadeira. Se a condição for
falsa, a estrutura é finalizada sem executar os comandos. O comando
que define a estrutura é representado pela palavra IF.")

    usleep(3000000)
    quebraLinha(qtdVezes: 4)
    print("Pseudocódigo: ")

    usleep(1000000)
    quebraLinha(qtdVezes: 2)
    print("INICIO DO ALGORITMO")

    usleep(2000000)
    quebraLinha(qtdVezes: 1)
    print("SE <condição> ENTAO")

    usleep(2000000)
    quebraLinha(qtdVezes: 1)
    print("<comando>")

    usleep(2000000)
    quebraLinha(qtdVezes: 1)
    print("FIM SE")
}

```

```

    usleep(2000000)
    quebraLinha(qtdVezes: 1)
    print("FIM DO ALGORITMO")

    usleep(1000000)
    quebraLinha(qtdVezes: 2)
    print("O comando só será executado se a condição for
verdadeira.")

    usleep(1000000)
    quebraLinha(qtdVezes: 2)
    print("A Estrutura Condicional Composta segue o mesmo princípio
da Estrutura Condicional Simples, com a diferença de que quando a
condição não é satisfeita, será executado o outro comando. O comando
que define a estrutura é representado pelas palavras SE e SE NÃO.")

    usleep(3000000)
    quebraLinha(qtdVezes: 4)
    print("Pseudocódigo: ")

    usleep(1000000)
    quebraLinha(qtdVezes: 2)
    print("INICIO DO ALGORITMO")

    usleep(2000000)
    quebraLinha(qtdVezes: 1)
    print("SE <condição> ENTAO")

    usleep(2000000)
    quebraLinha(qtdVezes: 1)
    print("<comando A>")

    usleep(2000000)
    quebraLinha(qtdVezes: 1)
    print("SENAO")

    usleep(2000000)
    quebraLinha(qtdVezes: 1)
    print("<comando B>")

    usleep(2000000)
    quebraLinha(qtdVezes: 1)
    print("FIM SE")

    usleep(2000000)
    quebraLinha(qtdVezes: 1)
    print("FIM DO ALGORITMO")

    usleep(1000000)
    quebraLinha(qtdVezes: 2)
    print("Se a condição for verdadeira, então o comando A será
executado. Caso ela seja falsa, apenas o comando B será executado.")

```

```

        usleep(3000000)
        quebraLinha(qtdVezes: 4)
        print("Mas no final, quando vamos ter que usar uma estrutura
condicional?")

        usleep(1000000)
        quebraLinha(qtdVezes: 1)
        print("Usaremos estruturas condicionais em praticamente todos
programas que forem feitos hoje em dia. Mas para entendermos melhor,
vamos aplicar isso na prática.")

        usleep(3000000)
        quebraLinha(qtdVezes: 2)
        print("Vamos supor que agora você trabalha na empresa
NanoChallisnton na parte financeira, a empresa é pequena, foi criada
a 2 meses e você precisa ao final de cada mês, você precisa atribuir
o salário de cada grupo da empresa dependendo do seu cargo, por
enquanto temos somente 3 cargos: ")

        usleep(1000000)
        quebraLinha(qtdVezes: 1)
        print("- Auxiliar Administrativo - Salário a ser atribuido:
R$1350,00;")

        usleep(1000000)
        quebraLinha(qtdVezes: 1)
        print("- Logística - Salário a ser atribuido: R$3400,00;")

        usleep(1000000)
        quebraLinha(qtdVezes: 1)
        print("- Recursos Humanos - Salário a ser atribuido:
R$2500,00;")

        usleep(3000000)
        quebraLinha(qtdVezes: 4)
        print("O fim do mês chegou e está na hora de você atribuir o
salário de cada grupo dependendo do seu respectivo cargo.")

        //Array criada para auxiliar na logica de printar os nomes na
hora que precisar
        let cargos : [String] = ["Auxiliar Administrativo", "Logística",
"Recursos Humanos"]

        //Laço de repetição que pergunta qual o salário de cada item
presente no array
        for i in 0...(cargos.count - 1) {
            usleep(1000000)
            quebraLinha(qtdVezes: 2)
            print("Digite o salário do grupo com o cargo de " +
cargos[i])
            quebraLinha(qtdVezes: 1)
            print("Cargo do grupo: " + cargos[i] + " | Salário: R$" +
readLine()! + ",00 | Referênte ao mês: 07")
        }

```

```

        usleep(1000000)
        quebraLinha(qtdVezes: 2)
        print("Bacana!..")

        usleep(1000000)
        quebraLinha(qtdVezes: 1)
        print(".")

        usleep(1000000)
        quebraLinha(qtdVezes: 1)
        print(".")

        usleep(1000000)
        quebraLinha(qtdVezes: 1)
        print(".")

        usleep(3000000)
        quebraLinha(qtdVezes: 1)
        print("Será que é bacana mesmo atribuir manualmente cada salário
dependendo do cargo?")

        usleep(3000000)
        quebraLinha(qtdVezes: 2)
        print("Agora pensa comigo... Se tivesse uma forma de estruturar
um programa para que ele possa atribuir esses salários dependendo do
cargo autonomamente. ")

        usleep(1000000)
        quebraLinha(qtdVezes: 1)
        print("Parece uma boa ideia, não acha? ")

        usleep(1000000)
        quebraLinha(qtdVezes: 1)
        print("Então agora vamos atribuir o valor de cada um dos
cargos.")

        //Variaveis usadas para armazenar os salários
        var auxAdm : String?
        var logist : String?
        var rh : String?

        usleep(1000000)
        //Laço de repetição que solicita a entrada de dados do salario
para o usuário
        for i in 0...(cargos.count - 1) {
            quebraLinha(qtdVezes: 1)
            print("Digite o salário do cargo de " + cargos[i])
            if(i == 0) {
                auxAdm = readLine()
            } else if(i == 1) {
                logist = readLine()
            } else {
                rh = readLine()
            }
        }
    }
}

```

```

    }
}

//Array criada para armazenar as variaveis dos salários,
organizando melhor o código e deixando ele melhor visivelmente
var salarioCargos : [String] = [auxAdm!, logist!, rh!]

usleep(1000000)
quebraLinha(qtdVezes: 1)
print("A partir de agora, cada mes que se passar, para cada
grupo, ele vai verificar qual cargo é com a estrutura condicional,
dependendo do cargo, o valor será atribuido autonomamente.")

usleep(1000000)
quebraLinha(qtdVezes: 1)
print("Por exemplo, se o cargo for igual a Auxiliar
Administrativo, ja será atribuido o valor de Auxiliar Adminstrativo
para esse grupo ou funcionário.")

usleep(3000000)
quebraLinha(qtdVezes: 4)
//Laço criado para representar cada mês de exemplo que será
atribuido os salários
for i in 8...10 {
    print("Estamos no mês " + String(i) + " chegou a hora de
atribuir novamente o salário para cada cargo, que bom que agora
temos nossa estrutura condicional!")
    //Laço criado para printar cada cargo e cada salário em um
mês em específico
    for j in 0...(cargos.count-1) {
        usleep(2000000)
        quebraLinha(qtdVezes: 1)
        print("Cargo do grupo: " + cargos[j] + " | Salário: R$"
+ salarioCargos[j] + ",00 | Referênte ao mês: " + String(i))
    }
    usleep(3000000)
    quebraLinha(qtdVezes: 2)
}

quebraLinha(qtdVezes: 4)
print("Viu como é bom utilizar estruturas condicionais? Espero
você no próximo tópico!")

usleep(3000000)

}

//Função que tem como objetivo ensinar o usuário o que é uma
variável
func ensinaVariáveis() {

    //Foi utilizado a função usleep para organizar melhor como o
texto vai aparecer para o usuário

```

```
    quebraLinha(qtdVezes: 1)
    usleep(1000000)
    print("Variável é um jeito de guardar informação em um programa
de computador. Tente imaginar que uma variável é um container, o
nome da variável é o nome estampado no container usado para
identificá-lo, você pode guardar informações dentro dessa variável
assim como pode guardar dentro do container, e quando você quiser
saber o que tem dentro do container, é só achar o nome identificando
o container e abrir ele, a mesma coisa acontece com uma variável.")
    quebraLinha(qtdVezes: 4)
```

```
    usleep(3000000)
    print("Como criar uma variável?")
    quebraLinha(qtdVezes: 1)
```

```
    usleep(1000000)
    print("Primeiramente temos que entender o método de criar uma
variável em Swift, para criar variáveis em outras linguagens não
muda o conceito mas a sintaxe sim. Em Swift temos os dois meios de
criar variáveis, podemos criar a variável(tradicional) e a
constante, o nome já diz, mas quando criamos uma
variável(tradicional), estamos criando um espaço que poderá ser
modificado futuramente, já com a constante é diferente, a constante
é um espaço criado que não poderá ser modificado futuramente.")
    quebraLinha(qtdVezes: 1)
```

```
    usleep(3000000)
    print("Em praticamente todas linguagens de programação, temos os
tipos de variáveis, que são usados para saber qual tipo de valor vai
ser armazenado nela, Por exemplo: ")
```

```
    usleep(1000000)
    quebraLinha(qtdVezes: 1)
    print("- Int: Para numeros inteiros positivos ou negativos;")
    print("- String: Para armazenar texto dentro da variável;")
    print("- Float: para armazenar números decimais;")
    quebraLinha(qtdVezes: 1)
    print("Entre varios outros tipos de variáveis usadas hoje em
dia.")
```

```
    usleep(1000000)
    quebraLinha(qtdVezes: 4)
    print("Exemplo de como criar uma variável em Swift: ")
    quebraLinha(qtdVezes: 1)
```

```
    usleep(3000000)
    print("let novoNumero : Int = 3")
    quebraLinha(qtdVezes: 1)
```

```
    usleep(3000000)
    print("Acabamos de criar uma constante do tipo Inteiro com o
valor 3, podemos usar esse valor para varios problemas, depende do
que você quer fazer.")
```



```

        quebraLinha(qtdVezes: 4)
        print("Agora tente você mesmo criar uma variável. Crie uma
constante com o nome novaVariavel do tipo inteiro, essa variável
vair ser igual a 15: ")
        quebraLinha(qtdVezes: 1)

        //Laço de repetição criado para o usuário tentar criar uma
variável em swift, caso não acerte, ele tem uma nova chance
        while(readLine() != "let novaVariavel : Int = 15") {

            usleep(1000000)
            quebraLinha(qtdVezes: 1)
            print("O que você digitou está errado, verifique o exemplo
acima e tente novamente. Crie uma constante com o nome novaVariavel
do tipo inteiro, essa variável vair ser igual a 15: ")
            quebraLinha(qtdVezes: 1)
        }

        usleep(1000000)
        quebraLinha(qtdVezes: 1)
        print("Parabéns! Você conseguiu criar uma variável em swift.")
        quebraLinha(qtdVezes: 4)

        usleep(1000000)
        print("Mas agora pode ter ficado uma dúvida em você... Por qual
motivo devemos usar uma variável? Qual benefício ele me trará?")

        usleep(3000000)
        quebraLinha(qtdVezes: 4)
        print("Vamos tirar essa dúvida na prática!")

        usleep(1000000)
        quebraLinha(qtdVezes: 1)
        print("Vamos supor que agora você trabalha na empresa
NanoChallisnton na parte financeira, a empresa é pequena, foi criada
a 2 meses e você precisa ao final de cada mês, marcar o mês que
vamos entrar no recibo de cada funcionário, por enquanto a empresa
tem 3 funcionários. Vamos lá?")

        usleep(3000000)
        quebraLinha(qtdVezes: 1)
        print("Vamos entrar em maio, digite o número do mês de maio (05)
em cada recibo: ")
        quebraLinha(qtdVezes: 1)

        //Laço de repetição criado para fazer o usuário escrever 3 vezes
o mês de maio para cada funcionário da empresa
        for i in 1...3 {
            usleep(1000000)
            print("Digite o número do mês de maio para o Funcionário " +
String(i))
            print("Nome: Funcionário " + String(i) + " | Valor:
R$1000.00 | Referênte ao mês: " + readLine()!)
            quebraLinha(qtdVezes: 1)

```

```

    }

    usleep(1000000)
    quebraLinha(qtdVezes: 1)
    print("Bacana! Você atribuiu o mês no recibo de cada funcionário
da NanoChallinston. Até o próximo mês...")

    usleep(3000000)
    quebraLinha(qtdVezes: 4)
    print("Estamos no mês de junho (6) e felizmente a empresa
cresceu bruscamente o número de funcionários. Agora temos 33
funcionários na empresa!");

    usleep(1000000)
    quebraLinha(qtdVezes: 4)
    print("Digite o número do mês de junho (06) em cada recibo: ")
    quebraLinha(qtdVezes: 1)
    for i in 1...33 {
        print("Digite o número do mês de junho para o Funcionário "
+ String(i))
        print("Nome: Funcionário " + String(i) + " | Valor:
R$1000.00 | Referente ao mês: " + readLine()!)
        quebraLinha(qtdVezes: 2)
        usleep(1000000)
        print("Deseja continuar ou ver um jeito melhor de atribuir o
mês no recibo de cada funcionário? Digite (1) para ver um jeito
melhor ou digite (2) para continuar: ")
        if(readLine()! != "2") {
            break;
        }
    }

    usleep(1000000)
    quebraLinha(qtdVezes: 1)

    print("Bem cansativo atribuir o mês para cada funcionário 33
vezes né? Já parou para pensar que, se você criar uma variável
identificada com o número mês e atribuir para todo funcionário da
empresa, você só precisa alterar uma vez?")

    usleep(3000000)
    quebraLinha(qtdVezes: 4)
    print("Vamos supor que acabou de ser implementado uma variável
que contém o número do mês para cada funcionário da empresa. Já
estamos no mês de julho (07).")

    usleep(3000000)
    quebraLinha(qtdVezes: 1)
    print("Digite o número do mês de julho para atualizar a
variável: ")

    while(readLine()! != "07") {
        quebraLinha(qtdVezes: 1)
        print("Você digitou o número do mês de julho da forma

```

```

errada, digite o número do mês de julho: Dica(Julho: 07)")
    }

    usleep(1000000)
    quebraLinha(qtdVezes: 1)
    //Laço de repetição que mostra como seria a exibição e execução
    caso fosse utilizado variáveis
    for i in 1...10 {

        usleep(500000)
        if(i >= 7) {
            print(".")
            quebraLinha(qtdVezes: 1)
        } else {
            print("Nome: Funcionário " + String(i) + " | Valor:
R$1000.00 | Referente ao mês: 07")
            quebraLinha(qtdVezes: 1)
        }
    }

    usleep(3000000)
    quebraLinha(qtdVezes: 4)
    print("Conseguiu perceber que é desnecessário atribuir o mês
para cada funcionário sendo que você pode apenas alterar a variável
mês que já está atribuída a todos?")

    usleep(2000000)
    quebraLinha(qtdVezes: 1)
    print("Acredito que você já tenha sua resposta, até a próxima
aula!")

    usleep(3000000)
}

//Criando a variável que vai ser usada de base para a condição que o
programa vai seguir
var controller : String = "0"

//storytelling
print("Bem vindo a aula de algoritmos em Swift!")
quebraLinha(qtdVezes: 1)
print("Primeiramente escolha qual tema você deseja abordar: ")
quebraLinha(qtdVezes: 1)

while(Int(controller) != 1) {

    //Opções que o usuário pode escolher para chamar a função
    específica
    quebraLinha(qtdVezes: 1)
    print("Digite:");
    quebraLinha(qtdVezes: 1)
    print("(1) para sair do programa;")
    quebraLinha(qtdVezes: 1)
    print("(2) para Variáveis;")
}

```

```

        quebraLinha(qtdVezes: 1)
        print("(3) para Condicionais;")
        quebraLinha(qtdVezes: 1)
        print("(4) para Laços de repetição;")
        quebraLinha(qtdVezes: 1)
        //Lendo o que o usuário digitar
        controller = readLine()!
        quebraLinha(qtdVezes: 1)

        //Criando uma estrutura condicional que vai verificar qual tema o
usuário quer aprender
        switch(Int(controller)) {
        case 1:
            break;
        case 2:
            ensinaVariáveis()
            break;
        case 3:
            ensinaCondicionais()
            break;
        case 4:
            ensinaLacoRepeticao()
            break;
        default:
            print("0 número digitado é inválido.")
            quebraLinha(qtdVezes: 1)
            break;
        }
    }
}

```