

Trabalho de Programação

Autor: Adriano Araújo Domingos dos Santos

Código de Pessoa: 1568608

Curso: Ciência da Computação - CAMPUS LOURDES

Turno: Manhã

Orientador: Soraia Lúcia da Silva

Enunciado

O aluno deverá criar um único problema e desenvolvê-lo nas cinco linguagens vistas em sala de aula:

- C
- C++
- C#
- Java
- Python

O programa deverá incluir: comandos de entrada e saída, algum operador relacional e aritmético, pelo menos uma condicional if composta/aninhada e uma estrutura de repetição. Submeta, no Canvas, um arquivo .pdf contendo os seguintes prints:

- A descrição do enunciado
 - O algoritmo em alto nível
 - Os códigos comentados nas linguagens solicitadas
 - Exemplos de execuções
-

Algoritmo em alto nível

1. Função para ler um número positivo do usuário pelo terminal

Repete os seguintes passos até que um número positivo seja entrado

1. Pede para o usuário que escreva um número pelo terminal
2. Lê o número
3. Se o número for negativo, escreve uma mensagem de aviso para o usuário

Retorna um número positivo.

2. Função para verificar se um número é primo

Recebe como parâmetro um número.

Se o número for maior que um, entra em uma estrutura de repetição que testa se o número é divisível por outro número além de 1 e dele mesmo. Com o passo de 1 a cada iteração, até que seja encontrado algum divisor, ou até que o índice da iteração seja maior que a raiz quadrada do número (ao invés de até o número, para evitar testes desnecessários).

Se nenhum divisor for encontrado, retorna Verdadeiro ou Falso caso contrário.

3. Função principal

Chama a função para ler um número positivo.

Passa o número para a função para verificar se o número é primo.

Caso o número seja primo, escreve na tela que ele é primo, caso contrário, escreve que o número não é primo.

Códigos comentados

C

```
// Importa a biblioteca para o uso de funções de input e output
#include <stdio.h>

// Importa a biblioteca para o uso do tipo boolean
#include <stdbool.h>

// Importa a biblioteca para o uso da função de raiz quadrada
#include <math.h>

// Função que retorna lê um número positivo do terminal
int readPositiveInteger() {
    int number;
    bool invalid;

    // Estrutura de repetição para garantir que o número seja positivo
    do {
        printf("\nEnter a number: ");
        scanf("%d", &number);

        invalid = number < 0;

        // Escreve mensagem de aviso quando o número é negativo
        if (invalid) printf("\aWarning: invalid input, the number should be
positive.");
    } while (invalid);

    return number;
}

// Função que recebe um número e retorna Verdadeiro se ele for primo ou Falso caso
contrário
bool numberIsPrime(int number) {
    // Teste executado para números negativos e um que pela definição não são primos
    bool isPrime = number > 1;

    // Variável de controle para evitar testes desnecessários
    int maxTest = sqrt(number);
```

```

    int i = 2;

    // Estrutura de repetição para verificar se o número não tem nenhum divisor fora
    1 e ele mesmo
    while (isPrime && i <= maxTest) {
        if (number % i == 0) isPrime = false;

        i++;
    }

    return isPrime;
}

// Função principal para a execução do algoritmo
int main() {
    int number = readPositiveInteger();

    // Escreve no terminal se o número é primo ou não
    if (numberIsPrime(number)) {
        printf("The number %d is prime!", number);
    } else {
        printf("The number %d is not prime!", number);
    }

    return 0;
}

```

C++

```

// Importa a biblioteca para o uso de funções de input e output
#include <iostream>

// Importa a biblioteca para o uso da função de raiz quadrada
#include <cmath>

// Facilitador de chamadas de funções de input e output
using namespace std;

// Função que retorna lê um número positivo do terminal
int readPositiveInteger() {
    int number;
    bool invalid;

    // Estrutura de repetição para garantir que o número seja positivo
    do {
        cout << "\nEnter a number: ";
        cin >> number;

        invalid = number < 0;

        // Escreve mensagem de aviso quando o número é negativo
    } while (invalid);
}

```

```

        if (invalid) cout << "\aWarning: invalid input, the number should be
positive.";
    } while (invalid);

    return number;
}

// Função que recebe um número e retorna Verdadeiro se ele for primo ou Falso caso
contrário
bool numberIsPrime(int number) {
    // Teste executado para números negativos e um que pela definição não são primos
    bool isPrime = number > 1;

    // Variável de controle para evitar testes desnecessários
    int maxTest = sqrt(number);
    int i = 2;

    // Estrutura de repetição para verificar se o número não tem nenhum divisor fora
1 e ele mesmo
    while (isPrime && i <= maxTest) {
        if (number % i == 0) isPrime = false;

        i++;
    }

    return isPrime;
}

// Função principal para a execução do algoritmo
int main() {
    int number = readPositiveInteger();

    // Escreve no terminal se o número é primo ou não
    if (numberIsPrime(number)) {
        cout << "The number ";
        cout << number;
        cout << " is prime!";
    } else {
        cout << "The number ";
        cout << number;
        cout << " is not prime!";
    }

    return 0;
}

```

C#

```

class Program
{
    // Função que retorna lê um número positivo do terminal
    static int ReadPositiveInteger()

```

```

{
    int number;
    bool invalid;

    // Estrutura de repetição para garantir que o número seja positivo
    do
    {
        Console.WriteLine("\nEnter a number: ");

        number = Convert.ToInt16(Console.ReadLine());

        invalid = number < 0;

        // Escreve mensagem de aviso quando o número é negativo
        if (invalid) Console.WriteLine("\aWarning: invalid input, the number should be
positive.");
    } while (invalid);

    return number;
}

// Função que recebe um número e retorna Verdadeiro se ele for primo ou Falso
caso contrário
static bool NumberIsPrime(int number)
{
    // Teste executado para números negativos e um que pela definição não são
    primos
    bool isPrime = number > 1;

    // Variável de controle para evitar testes desnecessários
    int maxTest = (int)Math.Sqrt(number);
    int i = 2;

    // Estrutura de repetição para verificar se o número não tem nenhum divisor
    fora 1 e ele mesmo
    while (isPrime && i <= maxTest)
    {
        if (number % i == 0) isPrime = false;

        i++;
    }

    return isPrime;
}

// Função principal para a execução do algoritmo
static void Main()
{
    int number = ReadPositiveInteger();

    // Escreve no terminal se o número é primo ou não
    if (NumberIsPrime(number))
    {
        Console.WriteLine("The number %d is prime!", number);
    }
}

```

```

    else
    {
        Console.WriteLine("The number %d is not prime!", number);
    }
}
}

```

Java

```

// Importa a biblioteca para o uso da função de raiz quadrada
import java.util.Scanner;

public class PrimeNumber {
    // Função que retorna lê um número positivo do terminal
    public static int readPositiveInteger() {
        // Cria o objeto scanner para poder coletar dados do terminal
        Scanner scanner = new Scanner(System.in);

        int number; boolean invalid;

        // Estrutura de repetição para garantir que o número seja positivo
        do {
            System.out.println("Enter a number: ");
            number = scanner.nextInt();

            invalid = number < 0;

            // Escreve mensagem de aviso quando o número é negativo
            if (invalid) System.out.println(
                "Warning: invalid input, the number should be positive."
            );
        } while (invalid);

        // Remove o objeto scanner para que não vaze recursos
        scanner.close();

        return number;
    }

    // Função que recebe um número e retorna Verdadeiro se ele for primo ou Falso
    caso contrário
    public static boolean numberIsPrime(int number) {
        // Teste executado para números negativos e um que pela definição não são
        primos
        boolean isPrime = number > 1;

        // Variável de controle para evitar testes desnecessários
        int maxTest = (int) Math.sqrt(number);
        int i = 2;

        // Estrutura de repetição para verificar se o número não tem nenhum divisor
        fora 1 e ele mesmo
    }
}

```

```

while (isPrime && i <= maxTest) {
    if (number % i == 0) isPrime = false;

    i++;
}

return isPrime;
}

// Função principal para a execução do algoritmo
public static void main(String[] args) {
    int number = readPositiveInteger();

    // Escreve no terminal se o número é primo ou não
    if (numberIsPrime(number)) {
        System.out.println("The number "+ number + " is prime!");
    } else {
        System.out.println("The number "+ number + " is not prime!");
    }
}
}

```

Python

```

# Importa a biblioteca para o uso da função de raiz quadrada
import math

# Função que retorna lê um número positivo do terminal
def readPositiveInteger():
    number = -1

    # Estrutura de repetição para garantir que o número seja positivo
    while number < 0:
        try:
            number = int(input("Enter a number: "))

            # Escreve mensagem de aviso quando o número é negativo
            if number >= 0:
                return number
            else:
                print("Warning: invalid input, the number should be positive.")
        # Escreve mensagem de aviso quando o usuário não escreve um número
        except ValueError:
            print("Warning: invalid input, enter a number.")

# Função que recebe um número e retorna Verdadeiro se ele for primo ou Falso caso contrário
def numberIsPrime(number: int):
    # Teste executado para números negativos e um que pela definição não são primos
    isPrime = number > 1

    # Variável de controle para evitar testes desnecessários

```

```
maxTest = int(math.sqrt(number))
i = 2

# Estrutura de repetição para verificar se o número não tem nenhum divisor fora
1 e ele mesmo
while isPrime and i <= maxTest:
    if number % i == 0:
        isPrime = False

    i += 1

return isPrime

# Função principal para a execução do algoritmo
def main():
    number = readPositiveInteger()

    # Escreve no terminal se o número é primo ou não
    if numberIsPrime(number):
        print(f"The number {number} is prime!")
    else:
        print(f"The number {number} is not prime!")

main()
```

Exemplos de execução

```
Enter a number: 7
The number 7 is prime!
```

```
Enter a number: 4
The number 4 is not prime!
```

```
Enter a number: -5
Warning: invalid input, the number should be positive.
Enter a number: 7
The number 7 is prime!
```