Questão 02

Alunos: Victor Shin Iti Kanazawa Noda (201905474)

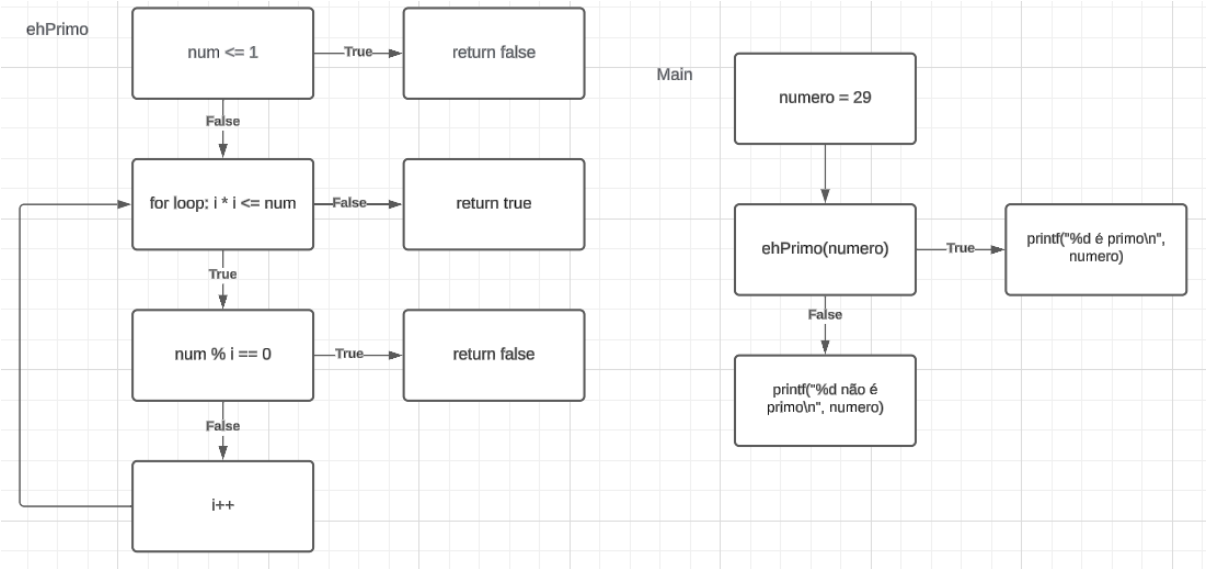
Assunto: Teste de Software e Gerência de Configuração de Software
Valor da questão: até 3,0 pontos.
Entrega: A resposta deverá ser entregue como projeto do GitHub. O link da resposta deverá ser enviado no SIGAA.

Considere trecho de código em C que implementa uma função para verificar se um número é primo:

```
#include <stdio.h>
#include <stdbool.h>

bool ehPrimo(int num) {
    if (num <= 1) {
        return false;
    }

    for (int i = 2; i * i <= num; i++) {
        if (num % i == 0) {
            return false;
        }
    }

    return true;
}

int main() {
    int numero = 29;
    if (ehPrimo(numero)) {
        printf("%d é primo\n", numero);
    } else {
        printf("%d não é primo\n", numero);
    }
    return 0;
}
```

Utilizando o teste estrutural (também conhecido como teste de caixa branca), responda o que se pede a seguir:

a) Qual o grafo de fluxo de controle para este código?



b) Quantos caminhos independentes existem neste código?

V = E - N + 2P
V = 11 - 10 + 2*1
V = 3

c) Liste todos os caminhos independentes identificados.

Caminho 1: num <= 1 é verdadeiro, retorna false.
Caminho 2: num > 1, entra no loop, encontra um divisor (num % i == 0 é verdadeiro), retorna false.
Caminho 3: num > 1, entra no loop, não encontra divisores, completa o loop, retorna true.

d) Para cada caminho independente, descreva um caso de teste que garantiria a cobertura desse caminho.

Caminho 1: num <= 1 é verdadeiro, retorna false

- **Entrada:** num = -1
- **Descrição:** Como num é menor ou igual a 1, a função ehPrimo deve retornar false.
- **Expectativa:** -1 não é primo

Caminho 2: num > 1, entra no loop, encontra um divisor (num % i == 0 é verdadeiro), retorna false

- **Entrada:** num = 4
- **Descrição:** Como num é maior que 1, a função entra no loop. Para i = 2, 4 % 2 == 0 é verdadeiro, então a função deve retornar false.
- **Expectativa:** 4 não é primo

Caminho 3: num > 1, entra no loop, não encontra divisores, completa o loop, retorna true

- **Entrada:** num = 29
- **Descrição:** Como num é maior que 1 e não tem divisores entre 2 e a raiz quadrada de 29, a função deve retornar true.
- **Expectativa:** 29 é primo

e) Quais são as condições lógicas presentes no código?

Condição num <= 1

- **Avaliação:** Verdadeira se num é menor ou igual a 1; caso contrário, falsa.
- **Efeito:** Se verdadeira, a função retorna false.

Condição i * i <= num no loop for

- **Avaliação:** Verdadeira enquanto o quadrado de i for menor ou igual a num.
- **Efeito:** Controla a execução do loop. Se falsa, o loop termina.

Condição num % i == 0 dentro do loop for

- **Avaliação:** Verdadeira se num é divisível por i sem resto; caso contrário, falsa.
- **Efeito:** Se verdadeira, a função retorna false.

f) Descreva um conjunto mínimo de casos de teste que garantam a cobertura de todas as condições lógicas.

Caso de Teste 1: num = -1 (Cobertura para num <= 1 verdadeiro)
Caso de Teste 2: num = 4 (Cobertura para num <= 1 falso, i * i <= num verdadeiro, num % i == 0 verdadeiro)
Caso de Teste 3: num = 5 (Cobertura para num <= 1 falso, i * i <= num verdadeiro, num % i == 0 falso)
Caso de Teste 4: num = 1 (Cobertura para num <= 1 verdadeiro)

Esses 5 casos de teste são suficientes para garantir que todas as ramificações lógicas do código sejam verificadas

g) Descreva os casos de teste usando análise de valor limite considerando que um número primo é aquele que é maior que 1 é divisível apenas por 1 e por ele mesmo.

1. Limite Inferior Menor que o Menor Número Primo

- **Entrada:** num = 0
- **Descrição:** Verifica se o programa identifica que 0 não é um número primo.
- **Expectativa:** 0 não é primo

2. Limite Inferior (Primeiro número não primo maior que 0)

- **Entrada:** num = 1
- **Descrição:** Verifica se o programa identifica que 1 não é um número primo.
- **Expectativa:** 1 não é primo

3. Limite Exatamente no Primeiro Número Primo

- **Entrada:** num = 2
- **Descrição:** Verifica se o programa identifica corretamente que 2 é um número primo.
- **Expectativa:** 2 é primo

4. Limite Superior Imediatamente Após o Primeiro Número Primo

- **Entrada:** num = 3
- **Descrição:** Verifica se o programa identifica corretamente que 3 é um número primo.
- **Expectativa:** 3 é primo

5. Limite Superior Imediatamente Após o Segundo Número Primo (Primeiro Composto Pequeno)

- **Entrada:** num = 4
- **Descrição:** Verifica se o programa identifica que 4 não é um número primo.
- **Expectativa:** 4 não é primo

6. Limite de um Número Primo Maior

- **Entrada:** num = 29
- **Descrição:** Verifica se o programa identifica corretamente que 29 é um número primo.
- **Expectativa:** 29 é primo

7. Limite de um Número Composto Maior

- **Entrada:** num = 30
- **Descrição:** Verifica se o programa identifica que 30 não é um número primo.
- **Expectativa:** 30 não é primo