

UNIVERSIDADE SÃO FRANCISCO

Ana Flávia Alves Pereira – RA: 001202004440

Gabriel de Assis Gomes – RA: 001202010614

Luan Sallowicz Padovan – RA: 001202000219

Pedro de Camargo Zago – RA: 001202002305

Teoria dos Grafos e Matemática Discreta

Dígito verificador

Bragança Paulista

2023

1. Introdução

Os dígitos verificadores desempenham um papel crucial na verificação da integridade e precisão dos dados em várias áreas, como processamento de informações, transmissão de dados e segurança da informação. Esses dígitos adicionais são utilizados para detectar erros ou alterações acidentais ou maliciosas nos dados durante sua transmissão, armazenamento ou processamento.

Este trabalho tem como objetivo explorar os conceitos e a importância dos dígitos verificadores e apresentar um código desenvolvido pelo grupo para demonstrar sua utilidade e eficácia na detecção de erros. Com base em um algoritmo específico, o código calculará e verificará os dígitos verificadores em diferentes conjuntos de dados.

2. Código

O código em questão é escrito em linguagem C e tem como objetivo calcular o dígito verificador de um código numérico de nove dígitos.

Imagem 1 – Código desenvolvido para implementar o dígito verificador

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

int fibonacci(int n) {
    if (n <= 1) {
        return n;
    }
    return fibonacci(n - 1) + fibonacci(n - 2);
}

int calcularSomaDigitos(int numero) {
    int soma = 0;

    while (numero > 0 || soma > 9) {
        if (numero == 0) {
            numero = soma;
            soma = 0;
        }
        soma += numero % 10;
        numero /= 10;
    }
    return soma;
}

int calcularDigitoVerificador(char * codigo) {
    int n = 9;
    int soma = 0;
    int codigoLength = strlen(codigo);

    for (int i = codigoLength - 1; i >= 0; i--) {
        int peso = fibonacci(n);
        soma += (codigo[i] - '0') * peso;
        n--;
    }

    return calcularSomaDigitos(soma);
}

int main() {
    char codigo[10];
    printf("Digite o código de nove dígitos: ");
    scanf("%9s", codigo);

    while (strlen(codigo) != 9) {
        printf("O código deve ter exatamente nove dígitos. Digite novamente: ");
        while (getchar() != '\n');
        scanf("%9s", codigo);
    }

    int digitoVerificador = calcularDigitoVerificador(codigo);
    printf("O dígito verificador do código %s é %d.\n", codigo, digitoVerificador);

    return 0;
}
```

2.1. Explicação código

Imagem 2 – Função fibonacci

```
int fibonacci(int n) {  
    if (n <= 1) {  
        return n;  
    }  
    return fibonacci(n - 1) + fibonacci(n - 2);  
}
```

fibonacci(int n): Essa função calcula o n-ésimo termo da sequência de Fibonacci. Ela recebe um número inteiro **n** como parâmetro e retorna o valor correspondente na sequência de Fibonacci. A função utiliza uma abordagem recursiva para calcular o valor. Se **n** for menor ou igual a 1, ela retorna o próprio **n**. Caso contrário, ela chama a função **fibonacci** com os parâmetros **n - 1** e **n - 2** e retorna a soma dos dois resultados.

Imagem 3 – Função calcularSomaDigitos

```
int calcularSomaDigitos(int numero) {  
    int soma = 0;  
  
    while (numero != 0) {  
        soma += numero % 10;  
        numero /= 10;  
    }  
  
    return soma;  
}
```

calcularSomaDigitos(int numero): Essa função calcula a soma dos dígitos de um número inteiro. Ela recebe um número **numero** como parâmetro e inicializa uma variável **soma** com zero. Em seguida, ela entra em um loop while que continua enquanto **numero** for diferente de zero. A cada iteração do loop, a função pega o último dígito de **numero** usando a operação **numero % 10** e o adiciona à variável **soma**. Depois, o valor de **numero** é atualizado para a sua parte inteira, dividindo-o por 10. Dessa forma, a cada iteração, o dígito mais à direita é removido. Ao final do loop, a função retorna o valor da variável **soma**.

Imagem 3 – Função calcularDigitoVerificador

```

int calcularDigitoVerificador(char * codigo) {
    int n = 15;
    int soma = 0;
    int codigoLength = strlen(codigo);

    for (int i = codigoLength - 1; i >= 0; i--) {
        int peso = fibonacci(n);
        soma += (codigo[i] - '0') * peso;
        n--;
    }

    return calcularSomaDigitos(soma) % 5;
}

```

calcularDigitoVerificador(char * codigo): Essa função calcula o dígito verificador do código de nove dígitos. Ela recebe uma string **codigo** como parâmetro. A função declara e inicializa algumas variáveis: **n** é inicializado com o valor 15, **soma** é inicializado com zero e **codigoLength** é calculado usando a função **strlen** para determinar o tamanho da string **codigo**.

Imagem 4 – Função main

```

int main() {
    char codigo[10];
    printf("Digite o código de nove dígitos: ");
    scanf("%9s", codigo);

    while (strlen(codigo) != 9) {
        printf("O código deve ter exatamente nove dígitos. Digite novamente: ");
        while (getchar() != '\n');
        scanf("%9s", codigo);
    }

    int digitoVerificador = calcularDigitoVerificador(codigo);
    printf("O dígito verificador do código %s é %d.\n", codigo, digitoVerificador);

    return 0;
}

```

main(): Essa é a função principal do programa. Ela lê um código de nove dígitos do usuário, calcula o dígito verificador desse código utilizando a função **calcularDigitoVerificador** e exibe o resultado na tela. Primeiro, é declarada uma variável **codigo** como uma string de tamanho 10 para armazenar o código digitado pelo usuário. Em seguida, o programa exibe uma mensagem solicitando ao usuário que digite o código de nove dígitos. O código é lido usando a função **scanf** com o formato "%9s" para garantir que apenas no máximo nove caracteres sejam lidos. Se o código digitado não tiver exatamente nove dígitos, o programa exibe uma mensagem de erro e solicita que o usuário digite novamente. O loop continua até que um código válido seja digitado. Quando um código válido é fornecido, o programa chama a função **calcularDigitoVerificador** passando

o código como parâmetro e armazena o resultado na variável **digitoVerificador**. Por fim, o programa exibe o código digitado e o dígito verificador calculado na tela.

3. Regras de Validação de dígito verificador

Tamanho da credencial – 9 Dígitos + 1 DV

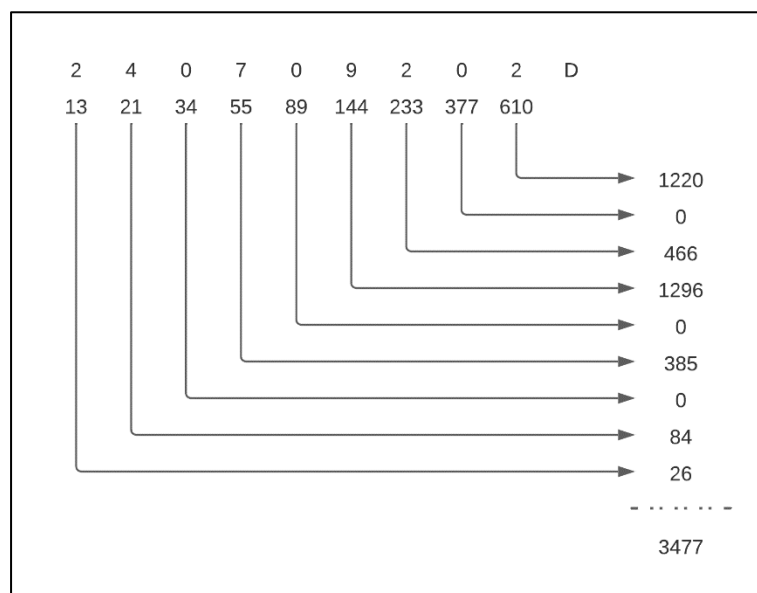
Exemplo:

Credencial: 240709202-1

Para obter o DV da credencial, multiplica-se cada dígito da credencial obtendo-se o peso correspondente a um valor na sequência de Fibonacci passando um valor decrescente de **n** (inicializado como 9) e multiplica esse peso pelo valor numérico do dígito.

Ex:

Credencial:



Os valores são somados $1220 + 0 + 466 + 1296 + 0 + 385 + 0 + 84 + 26 = 3477$.

O DV da credencial será, o resto da divisão por 5 da soma dos dígitos resultantes do cálculo com a sequência de Fibonacci.

Soma dos dígitos de 3477: $3 + 4 + 7 + 7 = 21$.

Mod (5): $21 \div 5 = 1$

O DV será 1, ficando 240709202-1 o resultado da credencial com o dígito.