



Algoritmos - Structs e Funções

Tarefa - Algoritmos - Structs e Funções

1 Structs e Funções

Struct é um recurso da linguagem de programação C que auxilia o programador na organização do código, mais precisamente na organização de armazenamento de dados. Struct permite criar um novo tipo definido pelo usuário que será um "agrupamento" de variáveis.

Structs são úteis, por exemplo, para permitir que uma função retorne mais de um valor, para criar tipos compostos como um tipo endereço que junta cep, estado, cidade, bairro, rua, número, para criar um tipo de fração na qual a fração é representada por um numerador e denominador, o código abaixo descreve esse último exemplo.

```
1 #include <stdio.h>
2
3 // Declaração de um tipo para representar uma fração
4 struct Fracao {
5     // utilizar um inteiro para representar o numerador
6     int numerador;
7     // utilizar um inteiro para representar o denominador
8     int denominador;
9 };
10
11 // criação de uma função que utiliza duas frações, f1 e f2 e
12 // calcula a soma f1 + f2. A soma f1 + f2 é retornada através
13 // de um tipo fração.
14 struct Fracao somarFracao(struct Fracao f1, struct Fracao f2) {
15     // variável intermediária para armazenar a soma de f1 e f2
16     struct Fracao resultado;
17     // regra de cálculo de soma de frações
18     resultado.numerador = f1.numerador * f2.denominador +
19         f2.numerador * f1.denominador;
20     resultado.denominador = f1.denominador * f2.denominador;
21     // retornar o valor da soma.
22     return resultado;
23 }
24
25 int main() {
26     // criação de frações e atribuição de valores, repare que é
27     // é possível inicializar o valor da fração através de { }
28     struct Fracao f1 = {1, 2};
```

```
29     struct Fracao f2 = {3, 5};
30     // chamada da função somarFracao com os valores de f1 e f2,
31     // o resultado será armazenado em res
32     struct Fracao res = somarFracao(f1, f2);
33     // impressão em tela das frações f1, f2 e res
34     printf("A soma das fracoes %d/%d e %d/%d eh %d/%d\n",
35           f1.numerador, f1.denominador, f2.numerador,
36           f2.denominador, res.numerador, res.denominador);
37     // retorno para o sistema operacional
38     return 0;
39 }
```

Programa 1: Exemplo de programa utilizando structs e funções

Os livros [Paul Deitel, 2022] e [Brian W. Kernighan, 1988] podem ser utilizados para obter mais informações sobre structs e funções.

2 Exercícios

1. Crie funções para somar, subtrair, multiplicar e dividir duas frações. Utilize Structs.
2. Crie funções para somar, subtrair, multiplicar e dividir dois numeros complexos. Utilize Structs.
3. Crie funções para somar, subtrair, multiplicar e calcular a inversa de duas matrizes 2x2. Utilize Structs.
4. Crie funções para somar e subtrair dois vetores de \mathbb{R}^3 , crie funções para calcular produto interno (produto escalar) e o produto cruzado entre dois vetores de \mathbb{R}^3 . Utilize Structs. (O Livro [Stewart, 2019] apresenta um resumo de vetores e cálculos de produto interno e cruzado).

Referências

- [Brian W. Kernighan, 1988] Brian W. Kernighan, D. M. R. (1988). *C Programming Language*. Prentice Hall, 2 edition.
- [Paul Deitel, 2022] Paul Deitel, H. D. (2022). *C How to Program*. Pearson, 9 edition.
- [Stewart, 2019] Stewart, J. (2019). *Calculus*. Cengage Learning, Boston, MA, 8th edition.