

Estruturas de dados heterogêneas (registros)

Registros em linguagem C (também conhecidos como structs) são estruturas de dados que permitem ao programador agrupar diferentes variáveis em uma única unidade lógica. Cada variável dentro do registro é chamada de membro e pode ter um tipo diferente.

Por exemplo, podemos criar um registro para armazenar informações de uma pessoa:

```
#include <stdio.h>

struct Pessoa{
    char nome[50];
    int idade;
    float altura;
};
```

Neste caso, a estrutura tem três membros: um array de caracteres para o nome da pessoa, um inteiro para a idade e um float para a altura.

Para utilizar essa estrutura para o armazenamento de um único registro, podemos criar uma variável dessa forma:

```
int main(){

    struct Pessoa p1;

    .
    .
    .
    return 0;
}
```

E então acessar e modificar os membros do registro usando o operador ponto (.):

```
int main(){

    struct Pessoa p1;

    strcpy(p1.nome, "João");
    p1.idade = 25;
    p1.altura = 1.75;

    .
    .
    .
    return 0;
}
```

Para utilizar essa estrutura para o armazenamento de vários registros, podemos criar uma variável do tipo vetor dessa forma:

```
int main(){

    struct Pessoa pessoa[5];

    .
    .
    .
    return 0;
}
```

E então acessar e modificar os membros usando o índice do vetor e operador ponto (.):

```
int main(){

    struct Pessoa pessoa[5];

    strcpy(pessoa[0].nome, "João");
    pessoa[0].idade = 25;
    pessoa[0].altura = 1.75;

    .
    .
    .
    return 0;
}
```

Exemplo

Escreva um programa em linguagem C que armazene o nome, a profissão e a renda de 3 pessoas em uma estrutura de registros.

```
#include <stdio.h>

/* Definição do registro */
struct Pessoa{
    char nome[50];
    char profissao[50];
    float renda;
};

int main(){

    /* Declaração do registro */
    struct Pessoa pessoa[3];

    /* Entrada dos dados */
    for (int i = 0; i < 3; i++) {
```

```

printf("Digite o nome da pessoa %d: ", i + 1);
scanf("%s", pessoa[i].nome);

printf("Digite a profissao da pessoa %d: ", i + 1);
scanf("%s", pessoa[i].profissao);

printf("Digite a renda da pessoa %d: ", i + 1);
scanf("%f", &pessoa[i].renda);
}

/* Saída dos dados */
printf("\nDados das pessoas:\n");
for (int i = 0; i < 3; i++) {
    printf("Pessoa %d:\n", i + 1);
    printf("Nome: %s\n", pessoa[i].nome);
    printf("Profissao: %s\n", pessoa[i].profissao);
    printf("Renda: %.2f\n\n", pessoa[i].renda);
}

return 0;
}

```

Atividades

1. Crie uma estrutura representando um aluno de uma disciplina. Essa estrutura deve conter o nome do aluno, o nome da disciplina e as notas do aluno em duas provas. Em seguida, leia os dados de três alunos e os armazena nessa estrutura. Por fim, exiba o nome do aluno, o nome da disciplina e a média dos alunos que foram aprovados com média igual ou superior a 6,0.
2. Suponha que você está fazendo uma pesquisa de preços para a compra de um tablet. Para auxiliá-lo, escreva um programa que permita o cadastro de cinco registros, cada um contendo o nome de uma loja e o preço do tablet naquela loja. O programa deve calcular a média dos preços e mostrar uma relação contendo o nome lojas que tenham o tablet com preço abaixo da média.
3. Crie uma estrutura representando um atleta. Essa estrutura deve conter o nome do atleta, seu esporte, idade e altura. Em seguida, leia os dados de 5 atletas. Calcule e exiba os nomes do atleta mais alto e do atleta mais velho.
4. Crie uma estrutura com o código e a quantidade em estoque de 5 produtos. Em seguida, leia um pedido composto por um código de produto e quantidade. Se houver quantidade suficiente para atender ao pedido integralmente, atualize a quantidade em estoque do produto e apresente a mensagem "Estoque atualizado", caso contrário, apresente a mensagem "Quantidade insuficiente em estoque".
5. Escreva um programa que simule contas bancárias com as seguintes especificações:
 - Ao iniciar o programa deverá ser criado uma estrutura para as contas bancárias contendo o número da conta, o nome do titular da conta e o saldo, definido inicialmente como 0.
 - Devem ser cadastrados 3 clientes diretamente no programa.
 - Crie um menu para o programa com as seguintes opções:

```
[1] Listagem de clientes  
[2] Depósito  
[3] Saque  
[4] Fim
```

Para cada depósito ou saque realizado, deverá ser mostrado o nome do titular e o saldo atualizado. O programa não deve permitir saques com valores superiores ao saldo atual do cliente.