Programação de Computadores I

DÉCIMA PARTE

CAPÍTULO IV

TIPOS DE DADOS COMPOSTOS

PARTE II - REGISTROS E UNIÕES

Programação de Computadores I

prof. Marco Villaça

REGISTROS
Motivação

- Por exemplo, no controle do estoque de uma loja é necessário informar:
  - ✓ Nome do produto
  - ✓ Quantidade em estoque
  - ✓ Preço
- Como há tipos diferentes, não é possível guardar a informação em um vetor (que só armazena elementos do mesmo tipo).
- Opção: criar um tipo próprio, que armazene essas informações.

REGISTROS
Definição

- Um registro é um agregado possivelmente heterogêneo de elementos de dados.
- Cada elemento individual é identificado por seu nome e acessados por meio de deslocamentos a partir do início da estrutura.
- Exemplo em C

```
struct aluno{
      char nome[20];
      int num_matricula;
} reg_aluno;
…
scanf(“%s”, reg_aluno.nome); // acesso
```

REGISTROS
Definição

```
struct aluno{
      char nome[20];
      int num_matricula;
} reg_aluno;
```

REGISTROS EM C

- Em uma estrutura, cada elemento ou campo (field) tem um nome e tipo
- Definição de uma estrutura

```
struct nome-estrutura {
  tipo-campo nome-campo1;
  tipo-campo nome-campo2;
} nome-variavel;
```

ESTRUTURAS

- Declaração de uma variável

```
struct nome-estrutura nome-variável;
```

- Para acessar os campos da estrutura, usa-se o operador . (ponto)

```
struct estoque {
   char nome[50];
   int quantidade;
   float preco;
};
struct estoque peca_computador[100];
peca_computador[0].quantidade = 20;
```

ESTRUTURAS

```
struct estoque {
    char nome[50];
   int quantidade;
    float preco;
};
struct estoque peca_computador[100];
peca_computador[0].quantidade = 20;
```

ESTRUTURAS

- Atribuição e cópia
  - ✓ Usando o operador igual (=)

```
struct ponto {
    int x;
    int y;
};
...
struct ponto a = {2, 3};
struct ponto b = {5, 8};
b = a; // agora o ponto b também tem coordenadas (2,3)
```

EXEMPLO 1

- Calcular a distância entre 2 pontos

CAPÍTULO IV

TIPOS DE DADOS COMPOSTOS

PARTE II - REGISTROS E UNIÕES CONTINUAÇÃO

Programação de Computadores I

prof. Marco Villaça

Exemplo 2

Criar uma estrutura para armazenar informação sobre alunos de uma disciplina

Nome do aluno
Número de matricula
Três notas do aluno
Media das notas

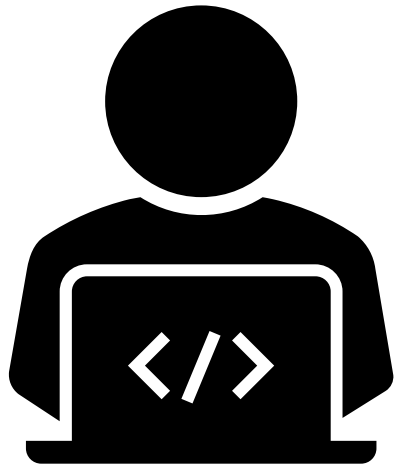
Você usara um vetor (matriz) de estruturas, em que o número de elementos e o número alunos

Por enquanto, fazer a entrada de dados pelo teclado (veremos, mais a frente, como importa-los de um arquivo)

- A média deve ser calculada
- O programa deve informar os alunos aprovados (media final > = 6).

Exercício 1

Modificar o Exemplo 2 de forma que sejam apresentados os alunos por ordem de melhor média.



Exercício 2

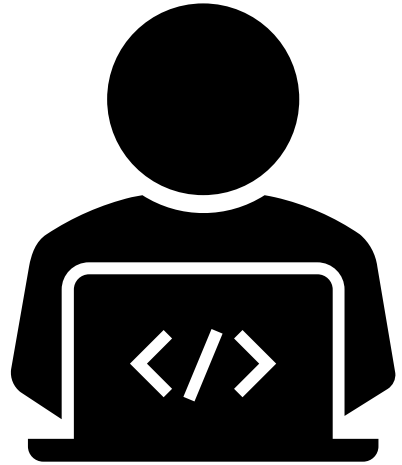
Criar uma estrutura para armazenar informação sobre alunos de uma academia de ginástica:

Nome do aluno, sexo, idade, peso, altura.

Você usara um vetor (matriz) de estruturas, em que o número de elementos é o número alunos

Entrar com os dados pelo teclado

O programa deve imprimir os alunos com imc > 30, informando ao lado do aluno “Obesidade”, e com imc < 18,5, informando ao lado do nome do aluno “Baixo peso”.



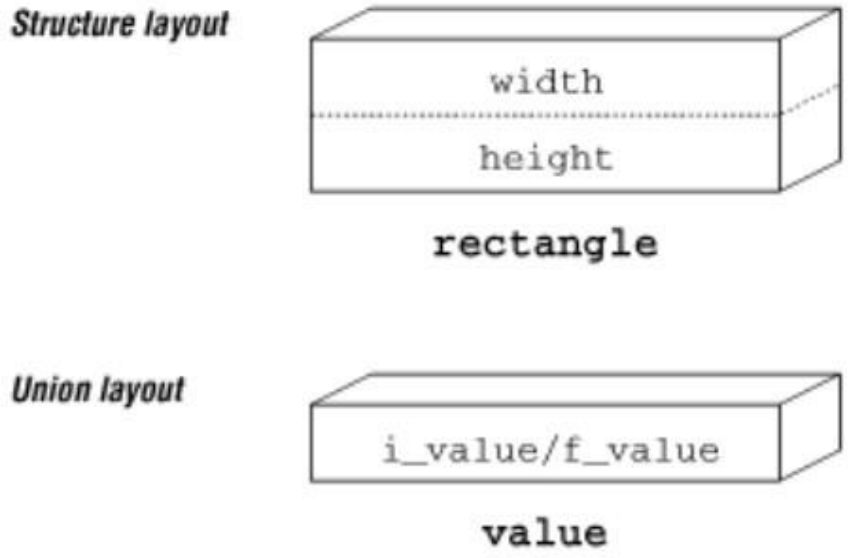
UNIÕES

Registro → campos de tipos diferentes; campos independentes

União → é um tipo especial de dado no qual as variáveis declaradas residem num mesmo endereço de memória.

Exemplo em C

```
struct rectangle {
        int width;
        int height;
    };
union value {
        long int i_value;
        float f_value;
    };
```



UNIÕES

Uniões são particularmente úteis em sistemas embarcados ou em situações onde o acesso direto a memória é necessário. Um exemplo trivial:

```
union registrador{
      struct{
          unsigned char byte1;
          unsigned char byte2;
          unsigned char byte3;
          unsigned char byte4;
       } bytes;
      unsigned int dword;
   };
```

UNIÕES

```
union registrador{
      struct{
          unsigned char byte1;
          unsigned char byte2;
          unsigned char byte3;
          unsigned char byte4;
       } bytes;
      unsigned int dword;
   };
```

- Nesse caso, a variavel reg pode ser acessada como segue:

```
union registrador reg;
reg.dword = 0x12345678;
printf("%x", reg.bytes.byte3); // Escreve 34
```

UNIÕES

- Outro exemplo semelhante:

```
union etipos { unsigned int var32;
               unsigned short int var16;
               unsigned char var8; } tipos;
```

- Se `tipos.var32 = 0x12345678`, resulta:
  - ✓ `tipos.var16 = 0x5678`
  - ✓ `tipos.var8 = 0x78`

UNIÕES ROTULADAS

- Através de um rótulo pode se saber qual membro da união está ativo.

```
struct exemplo {
    int rotulo;
    union {
          unsigned int var32;
          unsigned short int var16;
          unsigned char var8; } tipo;
     }var;
     ...
     switch(var.rotulo) {
       case 1; printf(“%d”, var.tipo.var32);break;
       case 2; printf(“%d”, var.tipo.var16);break;
       case 3; printf(“%d”, var.tipo.var8);break; }
```

Estruturas e uniões C99 Inicializadores designados

- No c99, inicializadores designados também podem ser utilizados em estruturas e uniões:

```
#include <stdio.h>
struct ponto{
	int x;
	int y; };
int main(){
	struct ponto a[5] =
	{
		[0] = {.x = 1, .y = 2},
		[4] = {.x = 10, .y = 20}
	};
 …
 }
```

Bibliografia e crédito das figuras

OUALLINE, S. Practical C Programming. 3a ed. O'Reilly, 1997.

SEBESTA, R. Conceitos de Linguagens de Programação. 5a ed. Porto Alegre: Bookman, 2003.

http://help.scilab.org/docs/6.1.0/pt_BR/index.html