Algoritmos e Programação II

III Lista de Exercícios: Estruturas

Prof. Evandro C. R. Rosa
UNIVALI

Nome Completo: _____ Código de Pessoa: _____

1. Responda sucintamente:

   (a) O que é um tipo de dado primitivo?

   (b) A declaração de estrutura cria automaticamente uma variável de estrutura?

   (c) Tanto arrays quanto estruturas podem armazenar múltiplos valores. Qual é a diferença entre um array e uma estrutura?

   (d) Veja o código abaixo:

```
struct Cidade {
      std::string nome_cidade;
      std::string nome_estado;
      double populacao;
      double altitude;
};

Cidade c = {
      "São José",
      "Santa Catarina",
      270299
};
```

   i. Qual valor está armazenado em `c.nome_cidade`?

   ii. Qual valor está armazenado em `c.nome_estado`?

   iii. Qual valor está armazenado em `c.populacao`?

   iv. Qual valor está armazenado em `c.altitude`?

   (e) Marque com verdadeiro (V) ou falso (F):

   ( ) É necessário um ponto e vírgula após a chave de fechamento de uma declaração de estrutura ou união.

   ( ) Uma declaração de estrutura cria uma variável.

   ( ) O conteúdo de uma variável de estrutura pode ser exibido passando a variável de estrutura para o `cout`.

   ( ) Em uma lista de inicialização de variáveis de uma estrutura, não é necessário fornecer inicializadores para todos os membros.

   ( ) Você pode pular membros em uma lista de inicialização de uma estrutura.

   ( ) A seguinte expressão se refere ao elemento 5 no array: `info_carro.modelo[5]`

( ) Uma variável membro de uma estrutura pode ser passada como argumento para uma função.

( ) Uma função pode retornar uma estrutura.

2. Os códigos abaixo possuem um ou mais erros. Indique quais são:

(a)
```
struct {
      int x;
      float y;
};
```

(b)
```
struct Valores {
      std::string nome;
      int idade;
}
```

(c)
```
struct DoisValores {
      int a, b;
};

int main () {
      DoisValores.a = 10;
      DoisValores.b = 20;
      return 0;
}
```

(d)
```
struct TresValores {
      int a, b, c;
};

int main() {
      TresValores valores = {1, 2, 3};
      std::cout << valores << std::endl;
      return 0;
}
```

(e)
```
struct Nomes {
      std::string primeiro;
      std::string ultimo;
};
int main () {
      Nomes cliente = "John", "Neumann";
      std::cout << cliente.primeiro << std::endl;
      std::cout << cliente.ultimo << std::endl;
      return 0;
}
```

(f)
```
struct QuatroValores {
      int a, b, c, d;
};
int main () {
      QuatroValores numeros = {1, 2, , 4};
      return 0;
}
```

(g)
```c
struct DoisValores {
      int a;
      int b;
};

int main() {
      DoisValores vetor[10];
      vetor.a[0] = 1;
      return 0;
}
```

3. Escreva um programa que simula o funcionamento de uma máquina de refrigerantes. O programa deve utilizar uma estrutura para armazenar os seguintes dados sobre cada refrigerante:

   - O nome do refrigerante (ex.: "Cola", "Laranja").
   - O preço de uma lata de refrigerante.
   - O número de latas de refrigerante disponíveis na máquina.

   O programa deve funcionar da seguinte forma:

   (a) Permitir que o usuário cadastre até cinco tipos diferentes de refrigerantes, informando o nome, o preço e a quantidade disponível de cada um.

   (b) Após a inicialização da máquina, o programa deve entrar em um loop onde:
   - A máquina exibe uma lista dos refrigerantes disponíveis.
   - O cliente seleciona um refrigerante e informa o valor em dinheiro inserido na máquina.
   - A máquina calcula e exibe o troco, subtrai uma unidade da quantidade de latas disponíveis e, se o refrigerante estiver esgotado, exibe uma mensagem de aviso.
   - O processo repete até que o cliente decida encerrar o programa.

   (c) Ao final do programa, exiba o total arrecadado pela máquina.

   Validação de Entrada: O programa deve garantir que:

   - O preço e a quantidade de latas de refrigerante não sejam valores negativos.
   - O valor inserido pelo cliente não seja negativo ou maior que R$10,00.

4. Escreva um programa que utiliza uma estrutura para armazenar as seguintes informações sobre uma conta de cliente:

   - Nome completo do cliente.
   - CPF do cliente.
   - Endereço inclui Cidade, Estado e CEP.
   - Telefone de contato do cliente.
   - O saldo atual na conta do cliente.

   O programa deve utilizar um array contendo, no mínimo, 10 estruturas de contas de clientes e oferecer uma interface de usuário baseada em menus que permita ao usuário:

   (a) Adicionar uma nova conta de cliente.

(b) Remover uma conta existente.

(c) Listar todas as contas cadastradas.

(d) Imprimir todos os dados de um cliente específico.

(e) Atualizar os dados de uma conta de cliente.

Validação de Entrada: O programa deve garantir que:

- Todos os campos sejam preenchidos ao adicionar uma nova conta.
- Não sejam inseridos saldos negativos para as contas.