

Exercício1 - Aula Prática 3

Tendo como dados de entrada o tipo de ovo (A, B ou C) e número de unidades desejadas, efetuar o atendimento do pedido, informando: número de unidades fornecidas (obedecendo limite máximo), valor do desconto (se for o caso) e valor da compra.

Informações adicionais:

1. **Preços:** A - R\$5,00; B - R\$4,30; C - R\$3,80.
2. **Limites:** A - 50 unidades; B - 30 unidades; C - 20 unidades;

Algoritmo: apenas passos

- Leitura e consistência (tipo válido, nro. de unidades > 0)
- Se válido:
 - verificar limites
 - estabelecer preço
 - calcular desconto (se for o caso)
 - informar resultados

Profa. Cora Pinto Ribeiro

1

Algoritmo venda de ovo

{Ler valores e.....}

Entrada: tipo(char), unidades pedidas(int)

Saída: unidades vendidas, valor venda, valor desconto, valor a pagar

1. início
2. ler tipo, unid_ped
3. se (tipo = 'A') ou (tipo = 'B') ou (tipo = 'C') //conteúdos válidos
..... (próximo slide)

```
3. Se (tipo = 'A') ou (tipo = 'B') ou (tipo = 'C') {conteúdos válidos}
3.1 se (tipo = 'A')
3.1.1 se unid_ped > 50
3.1.1.1 unid_vend = 50 // só vende 50
informa "parcialmente atendido, só 50 unidades"
3.1.1.2 senão
    unid_venda = unid_pedido // vende o nro solicitado
3.1.2 val_venda = unid_venda * 5.0 // calcula valor da venda
3.1.3 se unid_venda > 50 / 2 // verifica se terá desconto
3.1.3.1 desconto = val_venda * 0.05 // calcula 5% do valor da venda
3.1.3.2 senão
    desconto = 0 // não tem desconto
3.1.4 a_pagar = val_venda - desconto
3.1.5 informa vel_venda, desconto, a_pagar

3.2 se (tipo = 'B')
3.2.1 se unid_ped > 30
3.2.1.1 unid_vend = 30 // só vende 30
informa "parcialmente atendido, só 30 unidades"
3.2.1.2 senão
    unid_venda = unid_pedido // vende o nro solicitado
3.2.2 val_venda = unid_venda * 4.3 // calcula valor da venda
3.2.3 se unid_venda > 30 / 2 // verifica se terá desconto
3.2.3.1 desconto = val_venda * 0.05 // calcula 5% do valor da venda
3.2.3.2 senão
    desconto = 0 // não tem desconto
3.2.4 a_pagar = val_venda - desconto
3.2.5 informa vel_venda, desconto, a_pagar
```

```
3. Se (tipo = 'A') ou (tipo = 'B') ou (tipo = 'C') {conteúdos válidos}
3.1 se (tipo = 'A')
3.1.1 se unid_ped > 50
3.1.1.1 unid_vend = 50 // só vende 50
informa "parcialmente atendido, só 50 unidades"
3.1.1.2 senão
    unid_venda = unid_pedido // vende o nro solicitado
3.1.2 val_venda = unid_venda * 5.0 // calcula valor da venda
3.1.3 se unid_venda > 50 / 2 // verifica se terá desconto
3.1.3.1 desconto = val_venda * 0.05 // calcula 5% do valor da venda
3.1.3.2 senão
    desconto = 0 // não tem desconto
3.1.4 a_pagar = val_venda - desconto
3.1.5 informa vel_venda, desconto, a_pagar

// Se o tipo é A, porque está perguntando novamente?????

3.2 se (tipo = 'B')
3.2.1 se unid_ped > 30
3.2.1.1 unid_vend = 30 // só vende 30
informa "parcialmente atendido, só 30 unidades"
3.2.1.2 senão
    unid_venda = unid_pedido // vende o nro solicitado
3.2.2 val_venda = unid_venda * 4.3 // calcula valor da venda
3.2.3 se unid_venda > 30 / 2 // verifica se terá desconto
3.2.3.1 desconto = val_venda * 0.05 // calcula 5% do valor da venda
3.2.3.2 senão
    desconto = 0 // não tem desconto
3.2.4 a_pagar = val_venda - desconto
3.2.5 informa vel_venda, desconto, a_pagar
```

```
3. Se (tipo = 'A') ou (tipo = 'B') ou (tipo = 'C') {conteúdos válidos}
3.1 se (tipo = 'A')
3.1.1 se unid_ped > 50
3.1.1.1 unid_vend = 50 // só vende 50
informa "parcialmente atendido, só 50 unidades"
3.1.1.2 senão
    unid_venda = unid_pedido // vende o nro solicitado
3.1.2 val_venda = unid_venda * 5.0 // calcula valor da venda
3.1.3 se unid_venda > 50 / 2 // verifica se terá desconto
3.1.3.1 desconto = val_venda * 0.05 // calcula 5% do valor da venda
3.1.3.2 senão
    desconto = 0 // não tem desconto
3.1.4 a_pagar = val_venda - desconto
3.1.5 informa vel_venda, desconto, a_pagar

// Se o tipo é A, porque está perguntando novamente?????

3.2 se (tipo = 'B')
3.2.1 se unid_ped > 30
3.2.1.1 unid_vend = 30 // só vende 30
informa "parcialmente atendido, só 30 unidades"
3.2.1.2 senão
    unid_venda = unid_pedido // vende o nro solicitado
3.2.2 val_venda = unid_venda * 4.3 // calcula valor da venda
3.2.3 se unid_venda > 30 / 2 // verifica se terá desconto
3.2.3.1 desconto = val_venda * 0.05 // calcula 5% do valor da venda
3.2.3.2 senão
    desconto = 0 // não tem desconto
3.2.4 a_pagar = val_venda - desconto
3.2.5 informa vel_venda, desconto, a_pagar

// Se o tipo não é A nem B, porque está testando?????

3.3 se (tipo = 'C')
3.3.1 se unid_ped > 20
3.3.1.1 unid_vend = 20 // só vende 20
informa "parcialmente atendido, só 20 unidades"
3.3.1.2 senão
    unid_venda = unid_pedido // vende o nro solicitado
3.3.2 val_venda = unid_venda * 4.3 // calcula valor da venda
3.3.3 se unid_venda > 30 / 2 // verifica se terá desconto
3.3.3.1 desconto = val_venda * 0.05 // calcula 5% do valor da venda
3.3.3.2 senão
```

```
3. Se (tipo = 'A') ou (tipo = 'B') ou (tipo = 'C') {conteúdos válidos}
3.1 se (tipo = 'A')
3.1.1 se unid_ped > 50
3.1.1.1 unid_vend = 50 // só vende 50
informa "parcialmente atendido, só 50 unidades"
3.1.1.2 senão
    unid_venda = unid_pedido // vende o nro solicitado
3.1.2 val_venda = unid_venda * 5.0 // calcula valor da venda
3.1.3 se unid_venda > 50 / 2 // verifica se terá desconto
3.1.3.1 desconto = val_venda * 0.05 // calcula 5% do valor da venda
3.1.3.2 senão
    desconto = 0 // não tem desconto
3.1.4 a_pagar = val_venda - desconto
3.1.5 informa vel_venda, desconto, a_pagar

// Se o tipo é A, porque está perguntando novamente?????

3.2 se (tipo = 'B')
3.2.1 se unid_ped > 30
3.2.1.1 unid_vend = 30 // só vende 30
informa "parcialmente atendido, só 30 unidades"
3.2.1.2 senão
    unid_venda = unid_pedido // vende o nro solicitado
3.2.2 val_venda = unid_venda * 4.3 // calcula valor da venda
3.2.3 se unid_venda > 30 / 2 // verifica se terá desconto
3.2.3.1 desconto = val_venda * 0.05 // calcula 5% do valor da venda
3.2.3.2 senão
    desconto = 0 // não tem desconto
3.2.4 a_pagar = val_venda - desconto
3.2.5 informa vel_venda, desconto, a_pagar

// Se o tipo não é A nem B, porque está testando?????

3.3 se (tipo = 'C')
3.3.1 se unid_ped > 20
3.3.1.1 unid_vend = 20 // só vende 20
informa "parcialmente atendido, só 20 unidades"
3.3.1.2 senão
    unid_venda = unid_pedido // vende o nro solicitado
3.3.2 val_venda = unid_venda * 4.3 // calcula valor da venda
3.3.3 se unid_venda > 30 / 2 // verifica se terá desconto
3.3.3.1 desconto = val_venda * 0.05 // calcula 5% do valor da venda
3.3.3.2 senão
```

Redundância:
trecho mutuamente exclusivo
O que fazer:
sempre usar else

```

3. Se (tipo = 'A') ou (tipo = 'B') ou (tipo = 'C') (conteudos válidos)
3.1 se (tipo = 'A')
3.1.1 se (unidade > 50) // só vende 50
informa "parcialmente atendido, só 50 unidades"
3.1.1.2 senão
unidade_venda = unidade_pedido // vende o nro solicitado
3.1.2 val_venda = unidade_venda * 5.0 // calcula valor da venda
3.1.3 se (unidade_venda > 50) // verifica se terá desconto
3.1.3.1 desconto = val_venda * 0.05 // calcula 5% do valor da venda
3.1.3.2 senão
desconto = 0 // não tem desconto
3.1.4 a_pagar = val_venda - desconto
3.1.5 informa val_venda, desconto, a_pagar

3.2 se (tipo = 'B')
3.2.1 se (unidade > 30) // só vende 30
informa "parcialmente atendido, só 30 unidades"
3.2.1.2 senão
unidade_venda = unidade_pedido // vende o nro solicitado
3.2.2 val_venda = unidade_venda * 4.3 // calcula valor da venda
3.2.3 se (unidade_venda > 30) // verifica se terá desconto
3.2.3.1 desconto = val_venda * 0.05 // calcula 5% do valor da venda
3.2.3.2 senão
desconto = 0 // não tem desconto
3.2.4 a_pagar = val_venda - desconto
3.2.5 informa val_venda, desconto, a_pagar

3.3 se (tipo = 'C')
3.3.1 se (unidade > 20) // só vende 20
informa "parcialmente atendido, só 20 unidades"
3.3.1.2 senão
unidade_venda = unidade_pedido // vende o nro solicitado
3.3.2 val_venda = unidade_venda * 3.8 // calcula valor da venda
3.3.3 se (unidade_venda > 20) // verifica se terá desconto
3.3.3.1 desconto = val_venda * 0.05 // calcula 5% do valor da venda
3.3.3.2 senão
desconto = 0 // não tem desconto
3.3.4 a_pagar = val_venda - desconto
3.3.5 informa val_venda, desconto, a_pagar

```

O que varia:

- preço unitário
- limite

Restante:

exatamente igual!!!

O que fazer:

INTEGRAR!!!

Algoritmo venda de ovo
{Ler valores e....}
Entrada: tipo(char), unidades pedidas(int)
Saída: unidades vendidas, valor venda, valor desconto, valor a pagar

1. Início
2. ler tipo, unit_ped
// conteudos válidos, pode também incluir teste de unit_ped >= 0

3. se (tipo = 'A') ou (tipo = 'B') ou (tipo = 'C') // só entra aqui se consistente!

```

3.1 se (tipo = 'A') // seta valores para ovo tipo A
3.1.1 limite = 50
3.1.1.2 preco_unit = 5.0
3.1.2 senão
3.1.2.1 se (tipo = 'B') // seta valores para ovo tipo B
3.1.2.1.1 limite = 30
3.1.2.1.2 preco_unit = 4.3
3.1.2.2 senão // seta valores para ovo tipo C - é o que sobrou!!! Sem testar!!!
3.1.2.2.1 limite = 20
3.1.2.2.2 preco_unit = 3.8
define limite e preço unitário

// Calcula a venda, baseado no conteúdo em limite e preco_unit definidos acima:
3.2 se (unit_ped > limite)
3.2.1 unidade_venda = limite // só vende limite
informa "parcialmente atendido, só limite unidades serão vendidas"
3.2.1.2 senão
unidade_venda = unit_ped // vende o nro solicitado
3.2.2 val_venda = unidade_venda * preco_unit // calcula valor da venda
3.2.3 se (unidade_venda > limite / 2) // verifica se terá desconto
3.2.3.1 desconto = val_venda * 0.05 // calcula 5% do valor da venda
3.2.3.2 senão
desconto = 0 // não tem desconto
3.2.4 a_pagar = val_venda - desconto
3.2.5 informa val_venda, desconto, a_pagar
determina valores

3.2.6 senão
3.2.1 informa tipo solicitado não existe
4. fim

```

```

/* Processa venda de ovos de Páscoa */
#include<stdio.h>
#include<stdlib.h>
#include<ctype.h>
int main()
{
    int quant_ped, quant_venda, limite;
    char tipo_ovo;
    float venda, a_pagar, desconto, preco_unit;
    system("color f3"); // muda cor da tela / letras
    printf("Digite o tipo de ovo desejado (A-grande, B-médio, C-pequeno): ");
    scanf("%c", &tipo_ovo);
    tipo_ovo = toupper(tipo_ovo); // converte para maiúscula
    if((tipo_ovo != 'A') && (tipo_ovo != 'B') && (tipo_ovo != 'C'))
        printf("\n --- Tipo de ovo solicitado não existe. ---");
    else
    {
        // só chega aqui se tipo do ovo for válido - A, B ou C:
        printf("Digite o número de unidades do pedido: ");
        scanf("%d", &quant_ped);
        // determina preço unitário e limite, conforme o ovo escolhido:
        if(tipo_ovo == 'A')
        {
            preco_unit = 5;
            limite = 50;
        }
        else // se tipo não for A
        {
            // só chega aqui se tipo do ovo for válido - B, C ou C:
            printf("Digite o número de unidades do pedido: ");
            scanf("%d", &quant_ped);
            // determina preço unitário e limite, conforme o ovo escolhido:
            if(tipo_ovo == 'B')
            {
                preco_unit = 4.3;
                limite = 30;
            }
            else // se tipo não for B
            {
                preco_unit = 3.8;
                limite = 20;
            }
        }
        // verifica se quantidade pedida não é maior que o limite:
        if(quant_ped > limite)
        {
            printf("\n *** Seu pedido ultrapassou o limite de unidades permitidas.");
            printf("\n *** Unidades atendidas : %d ovos.", limite);
            quant_venda = limite; // unidades vendidas recebe valor limite
        }
        else
        {
            quant_venda = quant_ped; // unidades vendidas recebe pedido
        }
        venda = preco_unit * quant_venda; // calcula valor da venda
        if (quant_venda > limite / 2) // verifica se unidades vendidas > 50% do limite
            desconto = venda * 0.05;
        else
            desconto = 0;
        a_pagar = venda - desconto;
        printf("\n\nValor da venda : R$ %.2f", venda);
        printf("\nDesconto fornecido: R$ %.2f", desconto);
        printf("\nValor a se pagar: R$ %.2f\n\n", a_pagar);
        system("PAUSE>>NULL");
        return 0;
    }
}

```

```

.....
{ // só chega aqui se tipo do ovo for válido - A, B ou C:
printf("Digite o número de unidades do pedido: ");
scanf("%d", &quant_ped);
// determina preço unitário e limite, conforme o ovo escolhido:
if(tipo == 'A')
{
    preco_unit = 5;
    limite = 50;
}
else // se tipo não for A
if(tipo == 'B')
{
    preco_unit = 4.3;
    limite = 30;
}
else // só pode ser do tipo C:
{
    preco_unit = 3.8;
    limite = 20;
}
// verifica se quantidade pedida não é maior que o limite:
.....
}

```

```

// verifica se quantidade do pedido não é maior que o limite:
if(quant_ped > limite)
{
    printf("\n\n *** Seu pedido ultrapassou o limite de unidades permitidas.");
    printf("\n *** Unidades atendidas : %d ovos.", limite);
    quant_venda = limite; // unidades vendidas recebe valor limite
}
else
{
    quant_venda = quant_ped; // unidades vendidas recebe pedido
}
venda = preco_unit * quant_venda; // calcula valor da venda
if (quant_venda > limite / 2) // verifica se unidades vendidas > 50% do limite
    desconto = venda * 0.05;
else
    desconto = 0;
a_pagar = venda - desconto;
printf("\n\nValor da venda : R$ %.2f", venda);
printf("\nDesconto fornecido: R$ %.2f", desconto);
printf("\nValor a se pagar: R$ %.2f\n\n", a_pagar);
system("PAUSE>>NULL");
return 0;
}

```

C:\Cora\Disciplinas\INF01202\Programas\A...

Digite o tipo de ovo desejado (A-grande, B-médio, C-pequeno): A
 Digite o número de unidades do pedido: 57

*** Seu pedido ultrapassou o limite de unidades permitidas.
 *** Unidades atendidas : 50 ovos.

Valor da venda :	R\$ 250.00
Desconto fornecido:	R\$ 12.50
Valor a se pagar:	R\$ 237.50

Comandos de seleção condicional

• Condicional Simples

```
if (condição)
    comando;
```

• Seleção Dupla

```
if (condição)
    comando;
else comando;
```

→ Seleção múltipla

```
switch (variável)
{
    expressão;
}
```

Seja o problema:

Imprimir uma mensagem de acordo com um código recebido:

- 1 - Mensagem 1
- 2 - Mensagem 2
- 3 - Mensagem 3
- 4 - Mensagem 4

Para qualquer valor diferente de 1, 2, 3 ou 4, deve ser apresentada a mensagem "Mensagem de erro".

Soluções: if e if else

```
/* Imprime uma mensagem, de
com um código lido - solução us
simplis */
#include <stdio.h>
#include <stdlib.h>
int main ()
{
    int cod;
    printf("\nCodigo (1 a 4): ");
    scanf ("%d", &cod);
    if (cod == 1)
        printf ("\nMensagem 1\n");
    if (cod == 2)
        printf ("\nMensagem 2\n");
    if (cod == 3)
        printf ("\nMensagem 3\n");
    if (cod == 4)
        printf ("\nMensagem 4\n");
    if (cod < 1 || cod > 4)
        printf ("\nMensagem de er
system("Pause");
return 0;
} // solução inaceitável!!!

/* Imprime uma mensagem, de acordo com
um código lido - solução usando if - else */
#include <stdio.h>
#include <stdlib.h>
int main ()
{
    int cod;
    printf("\nCodigo (1 a 4): ");
    scanf ("%d", &cod);
    if (cod == 1)
        printf ("\nMensagem 1\n");
    else
        if (cod == 2)
            printf ("\nMensagem 2\n");
        else
            if (cod == 3)
                printf ("\nMensagem 3\n");
            else
                if (cod == 4)
                    printf ("\nMensagem 4\n");
                else // sobrou erro!
                    printf ("\nMensagem de erro\n");
    system("Pause");
    return 0;
}
```

Comando de seleção múltipla

Comando switch - case

- Quando o valor da expressão coincide com o valor de uma constante da lista, os comandos associados àquela constante e todos os demais comandos em sequência no switch/case, até o término de sua execução, são executados.
- Para que apenas os comandos associados a um (ou sequência de) case seja executado, devemos inserir o comando break no ponto de encerramento da execução.

Comando break

- Uso limitado ao switch/case (programação estruturada!).
- Finaliza a execução de uma sequência de cases, fazendo com que a execução retorne ao comando que segue ao switch.
- Sintaxe: break; .

Comando de seleção múltipla

seletor

switch (expressão)

```
{
    case constante:
        instruções;
    case constante:
        instruções;
    ...
    case constante:
        instruções;
    [default:
        instruções;
}
```

- Expressão: int ou char.
- Constantes: literais, do mesmo tipo da expressão.
- Valor da expressão comparado a conteúdos (na sequência em que estão).
- Se expressão = conteúdo (sempre constante literal), comando associado é executado até que break seja encontrado.
- Se nenhum conteúdo for igual, comandos associado a default (opcional!) são executados.

Solução: switch/case - INCORRETA.

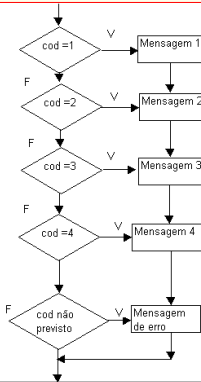
//Prog. incorreto, apresenta mais

```
#include <stdio.h>
#include <stdlib.h>
int main ()
{
    int cod;
    printf("\nCodigo (1 a 4): ");
    scanf ("%d", &cod);
    switch (cod)
    {
        case 1: printf ("\nMensagem 1\n");
        case 2: printf ("\nMensagem 2\n");
        case 3: printf ("\nMensagem 3\n");
        case 4: printf ("\nMensagem 4\n");
        default: printf ("\nMensagem de erro\n");
    }
    system("Pause");
    return 0;
}
```

Lista de constantes

```
ex C:\backup\aida\EAD2007\Encontro13\marswitch\msgcom
Codigo <1 a 4>: 3
Mensagem 3
Mensagem 4
Mensagem de erro
Pressione qualquer tecla para continuar. . .
```

Funcionamento de *switch/case* sem comandos *breaks*



No *switch/case* do programa anterior não foi usado nenhum comando *break* (ver segunda versão do programa a seguir)

Logo, quando um teste resulta verdadeiro, a mensagem correspondente e todas as mensagens seguintes são apresentadas, até o término do *switch/case*.

Solução: encerrar *case* com *break*.

Profa. Co

19

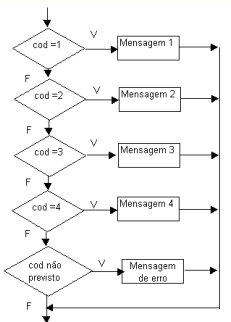
Solução: *switch/case* - CORRETA

```
//Imprime uma mensagem,
#include <stdio.h>
#include <stdlib.h>
int main ( )
{
    int cod ;
    printf("\nCodigo (1 a 4): ");
    scanf ("%d", &cod);
    switch (cod)
    {
        case 1: printf ("\nMensagem 1\n");
                break; // aqui aceita comandos! Não requer {}
        case 2: printf ("\nMensagem 2\n");
                break;
        case 3: printf ("\nMensagem 3\n");
                break;
        case 4: printf ("\nMensagem 4\n");
                break;
        default: printf ("\nMensagem de erro\n");
    } // último comando do switch não requer break: término normal
    system("Pause");
    return 0;
}
```

Profa. Cora Pinto Ribeiro

20

Funcionamento de *switch/case* com *breaks*



No *switch/case* do slide anterior foram usados *breaks* no final de cada *case*.

Logo, quando um teste resulta verdadeiro, a mensagem correspondente, e apenas ela, é apresentada e o *switch/case* é concluído.

Profa. Cora Pinto Ribeiro

21

Seja o problema:

Fazer um programa que calcule o preço de um produto, a partir do tempo levado em sua produção e do tempo gasto no projeto do produto (mínimo 1 mês cada).

Usar a seguinte tabela para calcular o preço:

- 2 meses: R\$ 1.000,00
- 3 meses: R\$ 1.500,00
- 4 meses: R\$ 2.000,00
- mais de 4 meses: R\$ 2.500,00

Profa. Cora Pinto Ribeiro

22

```
//Calcula preço a ser cobrado por um serviço
#include <stdio.h>
#include <stdlib.h>
int main ( )
{
    int tempo_prod, tempo_projeto;
    printf("Tempo de projeto: ");
    scanf ("%d", &tempo_projeto);
    printf("Tempo de produção: ");
    scanf ("%d", &tempo_prod);
    switch (tempo_prod + tempo_projeto)
    {
        case 2 : printf ("\nPreço: R$ 1.000,00\n");
                break;
        case 3 : printf ("\nPreço: R$ 1.500,00\n");
                break;
        case 4 : printf ("\nPreço: R$ 2.000,00\n");
                break;
        default: printf ("\nPreço: R$ 2.500,00\n");
    }
    system("Pause");
    return 0;
}
```

Expressão inteira
com operadores

Profa. Cora Pinto Ribeiro

23

Seja o problema:

Fazer um programa que simule uma calculadora que executa as quatro operações aritméticas (soma, subtração, produto e divisão). O programa deve ler os dois valores (operandos) e a operação que deverá ser efetuada. Após o cálculo, dar a resposta.

Profa. Cora Pinto Ribeiro

24

//executa uma das quatro operações aritméticas sobre dois operandos

```
#include <stdio.h>
#include <stdlib.h>
int main ( )
{
    float operando1, operando2;
    char operador;
    printf("nOperador: ");
    scanf("%c", &operador);
    printf("nOperando 1: ");
    scanf("%f", &operando1);
    printf("nOperando 2: ");
    scanf("%f", &operando2);
    switch (operador)
    {
        case '+': printf("nresultado: %fn", operando1 + operando2);
                break;
        case '-': printf("nresultado: %fn", operando1 - operando2);
                break;
        case '*': printf("nresultado: %fn", operando1 * operando2);
                break;
        case '/': printf("nresultado: %fn", operando1 / operando2);
                break;
        default: printf("nOperador invalido");
    }
    system("PAUSE");
    return 0;
}
```

Solução não
controla o
problema da
divisão por zero.

//executa uma das quatro operações aritméticas sobre dois operandos:

```
#include <stdio.h>
#include <stdlib.h>
int main ( )
{
    float operando1, operando2;
    char operador;
    printf("nOperador: ");
    scanf("%c", &operador);
    printf("nOperando 1: ");
    scanf("%f", &operando1);
    printf("nOperando 2: ");
    scanf("%f", &operando2);
    switch (operador)
    {
        case '+': printf("nresultado: %fn", operando1 + operando2);
                break;
        case '-': printf("nresultado: %fn", operando1 - operando2);
                break;
        case '*': printf("nresultado: %fn", operando1 * operando2);
                break;
        case '/': if (operando2 // verdadeiro se diferente de zero!!!
                    printf("nresultado: %fn", operando1 / operando2);
                    break;
                    else
                    printf("nDivisao por zero\n");
                    break;
        default: printf("nOperador invalido");
    }
    system("PAUSE");
    return 0;
}
```

Solução controla
o problema da
divisão por zero.

Seja o problema:

Processar e informar uma venda de livros em uma livraria.

Para isso, obter código do tipo de livro vendido (A, B, C) e número de unidades.

Preços:

Tipo A - R\$ 10,00

Tipo B - R\$ 20,00

Tipo C - R\$ 30,00

// Calcula preço de uma venda de livro:

```
#include <stdio.h>
#include <stdlib.h>
int main ( )
{
    char codigo;
    float a_pagar;
    int numero_unidades;
    printf("Codigo do livro: ");
    scanf("%c", &codigo);
    printf("Numero de unidades: ");
    scanf("%d", &numero_unidades);
    switch (codigo)
    {
        case 'A':
        case 'a': a_pagar = numero_unidades * 10;
                break;
        case 'B': a_pagar = numero_unidades * 20;
                break;
        case 'C': a_pagar = numero_unidades * 30;
                break;
        default: printf("Codigo errado");
                a_pagar = 0;
    }
    printf("nO valor total a pagar eh: R$ %.2f\n", a_pagar);
    system("Pause");
    return 0;
}
```

Default com mais de
um comando.

Imprime se
pagamento = 0!

// Calcula preço de uma venda de livro:

```
#include <stdio.h>
#include <stdlib.h>
int main ( )
{
    char codigo;
    float a_pagar;
    int numero_unidades;
    printf("Codigo do livro: ");
    scanf("%c", &codigo);
    printf("Numero de unidades: ");
    scanf("%d", &numero_unidades);
    switch (codigo)
    {
        case 'A':
        case 'a': a_pagar = numero_unidades * 10;
                break;
        case 'B': a_pagar = numero_unidades * 20;
                break;
        case 'C': a_pagar = numero_unidades * 30;
                break;
        default: printf("Codigo errado\n");
                a_pagar = 0;
    }
    if (a_pagar > 0) // só imprime se válido
        printf("nO valor total a pagar eh: R$ %.2f\n", a_pagar);
    system("Pause");
    return 0;
}
```

Para integrar
case 'A':
e
case 'a':
?

// Calcula preço de uma venda de livro:

```
#include <stdio.h>
#include <stdlib.h>
#include <ctype.h> // funções toupper e tolower
int main ( )
{
    char codigo;
    float a_pagar;
    int numero_unidades;
    printf("Codigo do livro: ");
    scanf("%c", &codigo);
    codigo = tolower(codigo); // converte caractere lido para minúscula, se possível
    printf("Numero de unidades: ");
    scanf("%d", &numero_unidades);
    switch (codigo)
    {
        case 'a': a_pagar = numero_unidades * 10;
                break;
        case 'b': a_pagar = numero_unidades * 20;
                break;
        case 'c': a_pagar = numero_unidades * 30;
                break;
        default: printf("Codigo errado\n");
                a_pagar = 0;
    }
    if (a_pagar > 0) // só imprime se válido
        printf("nO valor total a pagar eh: R$ %.2f\n", a_pagar);
    system("Pause");
    return 0;
}
```

Seja o problema:

Processar e informar uma venda de livros em uma livreria.
Para isso, obter código do tipo de livro vendido (A, B, C) e número de unidades.

Preços:

Tipo A - R\$ 10,00

Tipo B - R\$ 20,00

Tipo C - R\$ 30,00

Se tipo = B, ofereça 10% de desconto para compras a vista.

```
...
char codigo;
float a_pagar, desconto;
int numero_unidades, resposta;
printf("Codigo do livro: ");
scanf("%c", &codigo);
printf("Numero de unidades: ");
scanf("%d", &numero_unidades);
switch (codigo)
{
    case 'A':
        a_pagar = numero_unidades * 10;
        break;
    case 'B':
        printf("A vista? (1 - sim; 0 - nao): ");
        scanf("%d", &resposta);
        a_pagar = numero_unidades * 20;
        if (resposta) // se a vista, calcula e informa o desconto
        {
            desconto = a_pagar * 0.1;
            a_pagar = a_pagar - desconto;
            printf("O valor do desconto eh: R$ %.2fn", desconto);
        }
        break;
    case 'C':
        a_pagar = numero_unidades * 30;
        break;
    default: printf("Codigo errado\n");
            a_pagar = 0;
}
if (a_pagar > 0) // só imprime se válido
    printf("O valor total a pagar eh: R$ %.2fn", a_pagar);
system("Pause");
return 0;
}
```

Seja o problema:

Fornecido um código caractere, correspondente a um estado civil, devolver por extenso o estado civil correspondente.

Devem ser aceitas na entrada tanto maiúsculas quanto minúsculas.

Codificação dos estados civis considerados:

S - Solteiro

C - Casado

D - Divorciado

V - Viúvo

Qualquer outro caractere de entrada deve ser acusado como estado civil incorreto.

```
//Apresenta o estado civil correspondente ao caractere informado
#include <stdio.h>
#include <stdlib.h>
#include <ctype.h> // funções toupper e tolower
int main ( )
{
    char estado_civil;
    printf("\nEstado Civil (C - Casado, S - Solteiro, D - Divorciado ou V - Viúvo):");
    scanf("%c", &estado_civil);

    switch (estado_civil)
    {
        case 'c':
        case 'C': printf("\nCasado\n");
                 break;
        case 's':
        case 'S': printf("\nSolteiro\n");
                 break;
        case 'd':
        case 'D': printf("\nDivorciado\n");
                 break;
        case 'v':
        case 'V': printf("\nViúvo\n");
                 break;
        default: printf("\nEstado Civil incorreto\n");
    }
    system("PAUSE");
    return 0;
}
```

Valores literais char devem ser colocados entre apóstrofes.

O modo de funcionamento do switch/case permitiu aceitar para cada opção de estado civil tanto maiúscula quanto minúscula.

```
//Apresenta o estado civil correspondente ao caractere informado
#include <stdio.h>
#include <stdlib.h>
#include <ctype.h> // funções toupper e tolower
int main ( )
{
    char estado_civil;
    printf("\nEstado Civil (C - Casado, S - Solteiro, D - Divorciado ou V - Viúvo):");
    scanf("%c", &estado_civil);
    estado_civil = tolower(estado_civil); // se maiúscula, converte para minúscula
    switch (estado_civil)
    {
        case 'c': printf("\nCasado\n");
                 break;
        case 's': printf("\nSolteiro\n");
                 break;
        case 'd': printf("\nDivorciado\n");
                 break;
        case 'v': printf("\nViúvo\n");
                 break;
        default: printf("\nEstado Civil incorreto\n");
    }
    system("PAUSE");
    return 0;
}
```

Reduz teste apenas para minúsculas

Seja o problema:

Conforme uma nota inteira fornecida, entre 0 10, imprimir uma das seguintes mensagens:

0 a 5 - Reprovado

6 a 9 - Aprovado

10 - Aprovado com louvor.

Em C, esse problema pode ser resolvido tanto por uma sucessão de ifs, quanto por um comando switch/case.

Mensagem dependendo de nota inteira: com ifs.

```
//Recebida uma nota, fornecer uma das seguintes mensagens:
//0 a 6: Reprovado; 7 a 9: Aprovado; 10: Aprovado com louvor.
#include <stdio.h>
#include <stdlib.h>
int main ( )
{
    int nota;
    printf("\nNota (0 a 10): ");
    scanf("%d", &nota);
    if (nota < 0 || nota > 10) // elimina notas inválidas
        printf("Nota inválida\n");
    else
        if (nota <= 6)
            printf("\nReprovado\n");
        else
            if (nota <= 9)
                printf("\nAprovado\n");
            else // só sobrou nota = 10
                printf("\nAprovado com louvor\n");
    system("PAUSE");
    return 0;
}
```

Mensagem dependendo de nota inteira: com switch/case.

```
// Recebida uma nota, fornecer uma das seguintes mensagens:
//0 a 6: Reprovado; 7 a 9: Aprovado; 10: Aprovado com louvor.
#include <stdio.h>
#include <stdlib.h>
int main ( )
{
    int nota;
    printf("\nNota (0 a 10): ");
    scanf("%d", &nota);
    if (nota < 0 || nota > 10) // elimina valores inválidos!!!
        printf("Nota inválida\n");
    else
        switch (nota)
        {
            case 0:
            case 1:
            case 2:
            case 3:
            case 4:
            case 5:
            case 6: printf("\nReprovado\n");
                    break;
            case 7:
            case 8:
            case 9: printf("\nAprovado\n");
                    break;
            case 10: printf("\nAprovado com louvor\n"); // ou default
        }
    system("PAUSE");
    return 0;
}
```

Mas se a nota não for inteira?

Nesse caso, em C, apenas *ifs* podem ser usados para testar as condições.

Switch/case não é mais uma opção, já que só opera com expressões *int*, *long* (integer) ou *char*.

Ex: Fazer o trecho de um programa C que, dependendo do conteúdo de “codigo”, atribua o conteúdo especificado na tabela abaixo à variável “resultado”.

Codigo	Resultado
1	Resultado + 5
2	Inalterado
3	Resultado - 2
4	Inalterado
5	Resultado - 2
6	2* (Resultado +1)
7	Inalterado
8	0
9	0
10	0
11	123

```
...
//uso de seleção dupla
if (codigo == 1)
    resultado = resultado + 5;
else
    if (codigo == 3 || codigo == 5)
        resultado = resultado - 2;
    else
        if (codigo == 6)
            resultado = 2 * (resultado + 1);
        else
            if (codigo > 7 && codigo < 11)
                resultado = 0;
            else
                if (codigo == 11)
                    resultado = 123;
                else
                    if (codigo != 4 & codigo != 2 & codigo != 7)
                        printf("código inválido!");
...

```

Codigo	Resultado
1	Resultado + 5
2	Inalterado
3	Resultado - 2
4	Inalterado
5	Resultado - 2
6	2* (Resultado +1)
7	Inalterado
8	0
9	0
10	0
11	123

código inválido?

Para a próxima aula:

Converta o trecho abaixo, implementado através de if-elses encadeados, para uma solução que utilize o switch-case:

```
...
//uso de seleção dupla
if (codigo == 1)
    resultado = resultado + 5;
else
    if (codigo == 3 || codigo == 5)
        resultado = resultado - 2;
    else
        if (codigo == 6)
            resultado = 2 * (resultado + 1);
        else
            if (codigo > 7 && codigo < 11)
                resultado = 0;
            else
                if (codigo == 11)
                    resultado = 123;
                else
                    if (codigo < 1 || codigo > 11)
                        printf("código inválido!");
...

```