

Algoritmos com seleção: seleção múltipla



INF01202 – Algoritmos e Programação

Turma: H – Modalidade EAD – 2008/1

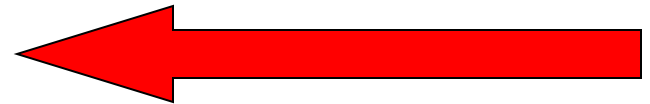
Professoras responsáveis:

Maria Aparecida C. Livi

Maria Aparecida M. Souto

Comandos de seleção

- seleção simples
- seleção dupla
- seleção múltipla





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".



Estratégias de solução usando *if...else...*

- Codificar uma sucessão de condições simples, uma para cada mensagem que deve ser produzida.
- Codificar uma sucessão de *ifs* aninhados, para obter o mesmo resultado, mas de forma mais eficiente.

A seguir os códigos correspondentes a cada uma dessas estratégias.

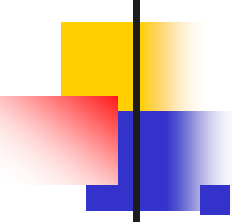
Apresentação de mensagens: primeira solução, usando condição simples

```
//Inserir cabeçalho ...
#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 erro\n");
    system("Pause");
    return 0;
}
```

Apresentação de mensagens: segunda solução, usando ifs aninhados

```
//Inserir cabeçalho ...
#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
                    printf ("\nMensagem de erro\n");
    system("Pause");
    return 0;
}
```

Seleção múltipla



A segunda solução do problema das mensagens, usando *ifs* aninhados, trabalha com seleção múltipla, ou seja, mais de duas condições estão sendo examinadas.

- Para esse tipo problema, além do comando *if...else...*, em vários casos é possível usar-se um outro comando: **SWITCH/CASE**.



Comando Switch/case

Comando que testa o valor de uma expressão contra uma lista de constantes.



Comando switch/case



Pseudolinguagem

```
escolha expressão  
  rótulo 1 : comando ;  
  rótulo 2 : comando ;  
  ...  
  rótulo n : comando ;  
[ por falta : comandos ]  
fim escolha
```

Expressão deve ser int, long ou char

Linguagem C

```
switch (expressão)  
{  
  case constante1 : instruções1 ;  
  case constante2 : instruções2 ;  
  ...  
  case constanten : instruçõesn ;  
  [ default: instruções; ]  
}
```

Uso de *default* é opcional

Constantes devem ser do mesmo tipo de expressão.

Comando Switch/case

Funcionamento:

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.

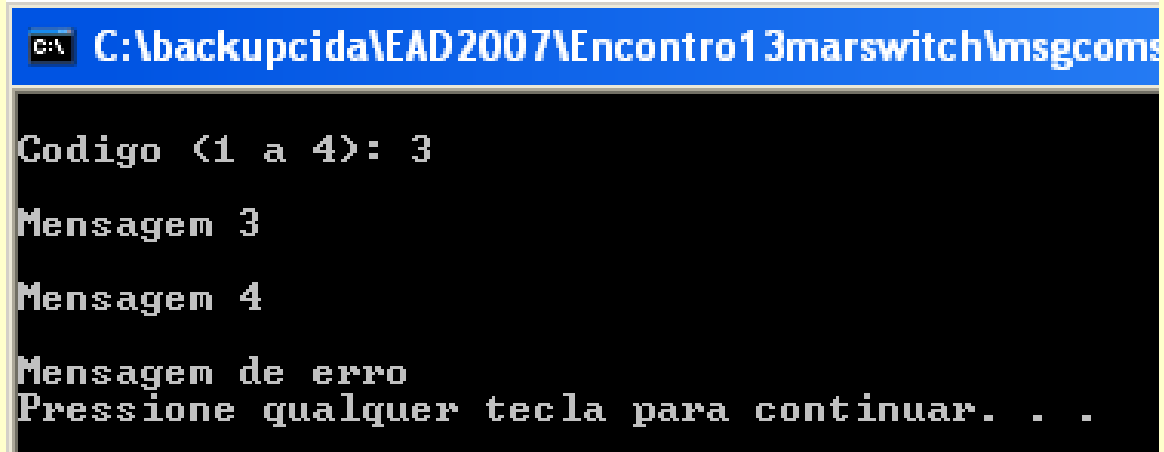
Apresentação de mensagens: terceira solução, com switch/case. INCORRETA.

//Prog. incorreto, apresenta mais de uma mensagem por código

```
#include <stdio.h>
#include <stdlib.h>
int main ( )
```

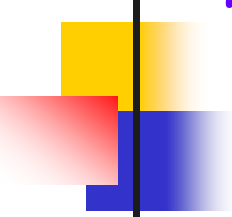
```
{
    int main ( ) Expressão
{
    int cod ;
    printf("\nCodigo (1 a 4): ");
    scanf ("%d", &cod);
    switch (cod)
```

```
{
    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;
}
```



```
C:\backupcida\EAD2007\Encontro13marswitch\msgcoms
Codigo (1 a 4): 3
Mensagem 3
Mensagem 4
Mensagem de erro
Pressione qualquer tecla para continuar. . .
```

Por que o código da terceira solução está incorreto?

- 
- Porque para cada valor de código deveria ser emitida apenas uma mensagem e, conforme o teste apresentado, pelo menos para um valor de código são apresentadas TRÊS mensagens.
 - Razão: **MODO DE FUNCIONAMENTO DO 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 imediatamente seguintes no Switch/case, até o término de sua execução, são executados.”
 - Assim, no caso em questão, após a mensagem 3 ser produzida, o comando **switch/case** continuou ativo e as demais ações em sequência foram executadas.

Correção da terceira solução: uso do comando **break**

Uma vez que as ações correspondentes a uma ou mais constantes estiverem encerradas, deve ser colocado um comando **break** para finalizar essa seqüência de cases. Ao ser ativado o **break**, ele interromperá a execução do **Switch/case** e a execução continuará no próximo comando do código do programa.

Apresentação de mensagens: quarta solução, com switch/case. CORRETA.

//Imprime uma mensagem, de acordo com um código lido

```
#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;
```

```
        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");
```

```
    }
```

```
    system("Pause");
```

```
    return 0;
```

```
C:\backupcida\EAD2007\Encontro13marswitch\msgcomsw
```

```
Codigo (1 a 4): 2
```

```
Mensagem 2
```

```
Pressione qualquer tecla para continuar. . .
```

Observar que nos cases
não é necessário
colocar-se **{ }** caso se
queira dois ou mais comandos.

Default não precisa de **break**
porque depois dele
o comando **switch/case**
vai terminar de
qualquer forma.



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>
```

```
int main ( )
```

```
{
```

```
char estado_civil;
```

```
printf("\nEstado Civil (C - Casado, S - Solteiro, D - Divorciado ou V - Viuvo):");
```

```
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 ("\nViuvo\n");  
break;
```

```
default: printf ("\nEstado Civil incorreto\n");
```

```
}
```

```
system("PAUSE");
```

```
return 0;
```

A expressão é tipo char, logo as constantes também devem ser.
Valores literais char devem ser colocados entre apóstofos.

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



Seja o seguinte problema:

Conforme uma nota inteira fornecida, entre 0 e 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**.

Apresentação de mensagem dependendo de nota inteira: com *ifs*.

```
//Cabecalho ...
//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 <=6)
        printf("\nReprovado\n");
    if (nota >=7 && nota <= 9)
        printf("\nAprovado\n");
    if (nota == 10)
        printf("\nAprovado com louvor\n");
    system("PAUSE");
    return 0;
}
```

Apresentação de mensagem dependendo de nota inteira: com *switch/case*.

```
//cabecalho ...  
// 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);  
    switch (nota)  
    {  
        case 0:  
        case 1:  
        case 2:  
        case 3:  
        case 4:  
        case 5:  
        case 6: printf("\nReprovado\n");  
        case 7:  
        case 8:  
        case 9: printf("\nAprovado\n");  
        case 10: printf("\nAprovado com louvor\n");  
    }  
    system("PAUSE");  
    return 0;  
}
```

Default no
switch/case
é opcional.

Seu uso é
uma escolha
do usuário.



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* ou *char*.

Apresentação de mensagem dependendo de nota não inteira: com ifs.

```
//Cabecalho ...  
//Recebida uma nota não inteira, 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 ( )  
{  
    float nota;  
    printf("\nNota (0 a 10): ");  
    scanf("%f", &nota);  
    if (nota >=0 && nota <=6)  
        printf("\nReprovado\n");  
    if (nota >6 && nota <10)  
        printf("\nAprovado\n");  
    if (nota == 10)  
        printf("\nAprovado com louvor\n");  
    system("PAUSE");  
    return 0;  
}
```

Ex: Fazer um programa que calcule o preço de um produto, a partir do tempo levado em sua produção. Neste tempo deve ser incluído o tempo gasto no projeto do produto. Usar a seguinte a 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

//Calcula preco a ser cobrado por um servico

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
int main ( )
```

```
{
```

```
    int tempo_prod, tempo_projeto;
```

```
    printf("Tempo de projeto: ");
```

```
    scanf("%d", &tempo_projeto);
```

```
    printf("Tempo de producao: ");
```

```
    scanf("%d", &tempo_prod);
```

```
    switch (tempo_prod + tempo_projeto)
```

```
    {
```

```
        case 2 : printf ("\nPreco: R$ 1.000,00\n");  
                break;
```

```
        case 3 : printf ("\nPreco: R$ 1.500,00\n");  
                break;
```

```
        case 4 : printf ("\nPreco: R$ 2.000,00\n");  
                break;
```

```
        default: printf ("\nPreco: R$ 2.500,00\n");
```

```
    }
```

```
    system("Pause");
```

```
    return 0;
```

```
}
```

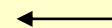
**Expressão inteira
com operadores**

Ex: Fazer um programa que simule uma calculadora que execute 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.

//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: %f\n", operando1 + operando2);
                  break;
        case '-': printf("\nResultado: %f\n", operando1 - operando2);
                  break;
        case '*': printf("\nResultado: %f\n", operando1 * operando2);
                  break;
        case '/': printf("\nResultado: %f\n", operando1 / operando2);
                  break;
        default: printf("\nOperador invalido");
                  break;
    }
    system("PAUSE");
    return 0;
}
```

**Solução não
controla o
problema da
divisão por zero.**

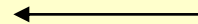


```

//Cabecalho ...
//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: %f\n", operando1 + operando2);
                  break;
        case '-': printf("\nResultado: %f\n", operando1 - operando2);
                  break;
        case '*': printf("\nResultado: %f\n", operando1 * operando2);
                  break;
        case '/': if (operando2)
                  {
                      printf("\nResultado: %f\n", operando1 / operando2);
                      break;
                  }
                  else
                  {
                      printf ("\nDivisao por zero\n");
                      break;
                  }
        default: printf ("\nOperador invalido");
                  break;
    }
    system("PAUSE");
    return 0;
}

```

**Solução controla
o problema da
divisão por zero.**



Ex: Processar uma venda de livros em uma livraria.
Fornecer 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
Calcular e informar preço a pagar.

```
#include <stdio.h>
#include <stdlib.h>
int main ( )
{
    char codigo;
    int numero_unidades, a_pagar;
    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':
        case 'b': a_pagar = numero_unidades * 20;
                break;

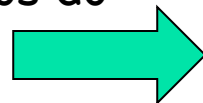
        case 'C':
        case 'c': a_pagar = numero_unidades * 30;
                break;

        default: printf("Codigo errado!");
                a_pagar = 0;
    }
    printf("\nO valor total a pagar eh: R$ %d\n", a_pagar);
    system("Pause");
    return 0;
}
```

Default com mais de um comando.

No mesmo problema, dar 10% desconto para pagamento a vista de livros do tipo B.

Algoritmos com seleção múltipla



Ex: Processar uma venda de livros em uma livraria.

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

Preços: Tipo A: R\$ 10,00

Tipo B: R\$ 20,00 (desconto de 10% para pagamento à vista deste tipo)

Tipo C: R\$ 30,00

Calcular e informar preço a pagar.

```
#include <stdio.h>
#include <stdlib.h>
int main ( )
{
    char  codigo;
    int   numero_unidades, resposta;
    float a_pagar;
    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':
        case 'b': printf("A vista? (1 - sim; 0 - nao): ");
                scanf("%d", &resposta);
                if (resposta)
                    a_pagar = numero_unidades * 20 * 0.9;
                else
                    a_pagar = numero_unidades * 20;
                break;

        case 'C':
        case 'c': a_pagar = numero_unidades * 30;
                break;

        default: printf("Codigo errado!");
                a_pagar = 0;
    }
    printf("\nO valor total a pagar eh: R$ %f\n", a_pagar);
    system("Pause");
    return 0;
}
```

Variável **a_pagar** que
era **int**, foi convertida
para **float**.



Aninhamento de switch/cases:

Seja o problema:

Realizar uma de três tarefas, conforme solicitação do usuário:

- 1 - emissão de mensagens de aprovação ou reprovação dependendo do valor de uma nota inteira;
- 2 - cálculo do preço a ser pago por um serviço;
- 3 - indicação por extenso de um estado civil informado.

Na solução apresentada a seguir, foram usados comandos *switch/case* para resolver, tanto a ativação das opções do usuário, quanto a solução de cada tarefa solicitada.

Tem-se assim um aninhamento de *switch/cases* no caso do cálculo do preço de um serviço.

Observar que nesses casos os valores das expressões são avaliados de forma independente nos *switch/cases*.

Realizar uma de três tarefas, conforme solicitação do usuário:

- 1 - emissão de mensagens de aprovação ou reprovação dependendo do valor de uma nota inteira;**
- 2 - cálculo do preço a ser pago por um serviço;**
- 3 - indicação por extenso de um estado civil informado.**

```
#include <stdio.h>
#include <stdlib.h>
int main ( )
{
    int opcao; // define o que sera feito
    int tempo_prod, tempo_projeto; // variaveis para calculo preco servico
    int nota; // variavel para opcao relativa a nota
    char estado_civil; // variavel para opcao de estado civil
    printf("\nOpcao (1 - mensagem nota; 2 - preco servico; 3 - estado civil): ");
    scanf("%d", &opcao);
    switch (opcao)
    {
        case 1: printf("\nNota (0 a 10): ");
                scanf("%d", &nota);
                switch (nota)
                {
                    case 0:
                    case 1:
                    (...)
                } // fim do switch/case da nota
                break;
        case 2: printf("Tempo de projeto: ");
                scanf("%d", &tempo_projeto);
                printf("Tempo de producao: ");
                scanf("%d", &tempo_prod);
                switch (tempo_prod + tempo_projeto)
                {
                    case 2 : printf ("\nPreco: R$ 1.000,00\n");
                            break;
                    (...)
                } // fim do switch/case da preco servico
                break;
        case 3: printf("\nEstado Civil (C - Casado, S - Solteiro, D - Divorciado ou V - Viuvo):");
                scanf(" %c", &estado_civil);
                switch (estado_civil)
                {
                    case 'c':
                    case 'C': printf ("\nCasado\n");
                            break;
                    (...)
                } // fim do switch/case do estado civil
                break;
    } // fim do switch/case opcao
    system("PAUSE");
    return 0;
}
```

Aninhamento de switch/cases

Relembrando



Switch/case só opera com expressões **int**, **long** ou **char** e só trabalha com a igualdade.

Constantes nos **cases** devem ser do mesmo tipo da expressão.

Uma vez que as ações correspondentes a uma ou mais constantes estiverem encerradas, deve ser colocado um comando **break** para finalizar essa sequência de **cases** e impedir a execução de outros **cases** não desejados.

Nos **cases** não é necessário colocar-se **{ }** caso se queira dois ou mais comandos.

Default não precisa de **break** porque depois dele o comando **switch/case** vai terminar de qualquer forma.

Default no **switch/case** é opcional. Seu uso é uma escolha do usuário.

Switch/cases podem ser aninhados. Nesses casos cada expressão é avaliada de modo independente.

Vantagens e Desvantagens do switch/case:

■ Desvantagens

- Serve como alternativa apenas para casos em que sejam testadas apenas igualdades envolvendo expressões `int`, `long` ou `char`.
- Não pode ser usada com expressões `float` ou `double`. Para estes casos, `if...else...` é mais abrangente!!

■ Vantagens

- O código produzido é mais claro.
- A lógica fica mais evidente.

