

# Algoritmos I



**DESVIO CONDICIONAL**

# Desvio Condicional



- Quando se torna necessário desviar o fluxo dos programas é utilizado o desvio condicional. Nestes casos parte do programa poderá ser executada, parte não, dependendo de um determinado teste lógico ou comando.
- A estrutura de desvio condicional deve ser utilizada quando se quer que uma condição seja analisada e:
  - caso esta condição seja verdadeira, o(s) comando(s) logo abaixo do teste lógico será(ão) executado(s); e
  - caso esta condição seja falsa, outro(s) comando(s) será(ão) executado(s).

# Desvio Condicional



- Os desvios condicionais podem ser de três tipos:
  - Simples;
  - Compostos; e
  - Encadeados.

# Desvio Condicional Simples



- Considere o seguinte enunciado:
  - Elabore um algoritmo que solicite ao usuário duas notas e informe ao final sua média aritmética.
- A resolução seria em algoritmo sequencial, como segue abaixo:

```
1  #include <locale.h>
2  #include <stdio.h>
3
4  int main () {
5      setlocale(LC_ALL, "Portuguese");
6
7      float nota1, nota2, media;
8
9      printf("Entre com as duas notas: ");
10     scanf("%f%f", &nota1, &nota2);
11
12     media = (nota1 + nota2)/2.0;
13
14     printf("\nA média é %.2f", media);
15
16     return 0;
17 }
```

# Desvio Condicional Simples



- Mas, e caso fosse necessário imprimir se o aluno foi aprovado na disciplina?
- Para poder realizar esta operação é necessário **tomar uma decisão**. Isto é feito executando um (ou mais) teste(s) lógico(s) e caso o resultado seja verdadeiro é executado um determinado bloco de instruções. Se for falso esse bloco de instruções não é executado.

**C/C++**

```
if (teste(s) lógico(s)) {  
    ... Bloco de Instruções ...  
}
```

# Desvio Condicional Simples



- Segue algoritmo desenvolvido com desvio condicional simples:

```
1  #include <locale.h>
2  #include <stdio.h>
3
4  int main () {
5      setlocale(LC_ALL, "Portuguese");
6
7      float nota1, nota2, media;
8
9      printf("Entre com as duas notas: ");
10     scanf("%f%f", &nota1, &nota2);
11
12     media = (nota1 + nota2)/2.0;
13
14     if (media >= 6) {
15         printf("\nAprovado com média %.2f.", media);
16     }
17     return 0;
18 }
```

Se a média do aluno for maior ou igual a 6 o programa imprimirá a mensagem. Caso seja falso, irá para o return 0 e encerrará o programa.

# Desvio Condicional Composto



- Mas ficou faltando algo no algoritmo anterior. E se fosse necessário imprimir que o aluno foi reprovado?
- Nesse caso seria necessário utilizar um Desvio Condicional Composto.
  - Se a condição do if for verdadeira será executado o bloco de instruções entre o if e o fechamento de chaves do if.
  - Caso sejam falsa, será executado o bloco de instruções entre as chaves do else.
- Sintaxe:

**C/C++**

```
if (teste(s) lógico(s)) {  
    ... Bloco de Instruções ...  
} else {  
    ... Bloco de Instruções ...  
}
```

# Desvio Condicional Composto



- Segue algoritmo desenvolvido com desvio condicional composto:

```
1  #include <locale.h>
2  #include <stdio.h>
3
4  int main () {
5      setlocale(LC_ALL, "Portuguese");
6
7      float nota1, nota2, media;
8
9      printf("Entre com as duas notas: ");
10     scanf("%f%f", &nota1, &nota2);
11
12     media = (nota1 + nota2)/2.0;
13
14     if (media >= 6) {
15         printf("\nAprovado com média %f.", media);
16     } else {
17         printf("\nReprovado com média %.2f.", media);
18     }
19     return 0;
20 }
```

Se a média do aluno for maior ou igual a 6 o programa imprimirá a mensagem. Caso seja falso, imprimirá a mensagem do else.



# Desvio Condicional Composto



- Outro exemplo. Solicite ao usuário um ano e informe se o mesmo é um ano bissexto ou não.

```
1  #include <stdio.h>
2  #include <locale.h>
3
4  int main(){
5      setlocale(LC_ALL, "Portuguese");
6
7      int ano;
8
9      printf("Qual o ano a ser verificado? ");
10     scanf("%d", &ano);
11
12     if (ano % 4 == 0 && (ano % 400 == 0 || ano % 100 != 0)){
13         printf("\nO ano %d é bissexto.", ano);
14     } else {
15         printf("\nO ano %d não é bissexto.", ano);
16     }
17
18     return 0;
19 }
```

Quando é necessário mais que um teste lógico, os mesmos devem estar interligados com operadores lógicos.

# Desvio Condicional Encadeado



- Como visto anteriormente o desvio condicional permite definir conjuntos de instruções que serão executados caso uma condição seja satisfeita ou não.
- Desvio Condicional também se trata de uma instrução, logo é possível incluir desvios condicionais um dentro do outro.
- E quando isso é feito? Quando se possui três ou mais instruções a serem testadas.
- Ao se encadear um desvio deve-se tomar cuidado, pois uma chave fechada em um local errado pode mudar toda a lógica do programa.
- Existem várias formas de encadeamento, dependendo do que é solicitado. Alguns exemplos:

# Desvio Condicional Encadeado



## C/C++

```
if (teste(s) lógico(s)) {  
    ... Bloco de Instruções ...  
} else if (teste(s) lógico(s)) {  
    ... Bloco de Instruções ...  
} else {  
    ... Bloco de Instruções ...  
}
```

## C/C++

```
if (teste(s) lógico(s)) {  
    ... Bloco de Instruções ...  
    if (teste(s) lógico(s)) {  
        ... Bloco de Instruções ...  
    }  
} else if (teste(s) lógico(s)) {  
    ... Bloco de Instruções ...  
}
```

# Desvio Condicional Encadeado



- Veja um exemplo. Solicite ao usuário um valor inteiro e informe se o mesmo é positivo, negativo ou zero.

```
1  #include <locale.h>
2  #include <stdio.h>
3
4  int main () {
5      setlocale(LC_ALL, "Portuguese");
6
7      int numero;
8
9      printf("Qual o número a ser verificado? ");
10     scanf("%d", &numero);
11
12     if (numero > 0 ) {
13         printf("\nO número %d é positivo.", numero);
14     } else if (numero < 0) {
15         printf("\nO número %d é negativo.", numero);
16     } else {
17         printf("\nO número é Zero.");
18     }
19
20     return 0;
21 }
```

Se a primeira condição for verdadeira, executa o código entre chaves e não testa as outras duas. Caso seja falsa irá testar a segunda condição, e assim sucessivamente.

# Desvio Condicional Encadeado

- Faça um algoritmo que solicite ao usuário a média e a frequência final de uma disciplina e informe se o aluno foi aprovado ou reprovado, e em caso de reprovação se foi por nota ou média.

```
1  #include <stdio.h>
2  #include <locale.h>
3
4  #define TAM 30
5
6  int main(){
7      setlocale(LC_ALL, "Portuguese");
8
9      float mediafinal, frequencia;
10     char nome[TAM];
11
12     printf("Nome do aluno: ");
13     gets(nome);
14     printf("Media Final: ");
15     scanf("%f", &mediafinal);
16     printf("Frequência Final: ");
17     scanf("%f", &frequencia);
18
19     if (mediafinal >= 6 && frequencia >= 75){
20         printf("%s, você foi aprovado.", nome);
21     } else if (mediafinal < 6 && frequencia < 75) {
22         printf("%s, você foi Reprovado por Média e Frequência.", nome);
23     } else if (mediafinal < 6){
24         printf("%s, você foi Reprovado por Média.", nome);
25     } else {
26         printf("%s, você foi Reprovado por Frequência.", nome);
27     }
28
29     return 0;
30 }
```

# Identação



- A identação é um termo aplicado ao código fonte, onde colocamos os elementos que são dependentes do outro mais a direita.
- Em algumas linguagens é obrigatório, em outras não. Mas torna o código mais legível, principalmente em grande quantidade de linhas de código.

# Comando SWITCH



- Assim como o if ... else, este é um comando de decisão.
- A diferença fundamental entre switch e if é que a estrutura switch não aceita expressões (condições), somente constantes.

## **Dica:**

- Utilize o comando switch para menus, pois ele fica mais claro que o comando if ... else.

# Comando SWITCH



## Sintaxe:

```
switch (expressao_de_controle) {  
    case <constante_1> :  
        <comando(s)>;  
        break;  
    case <constante_2> :  
        <comando(s)>;  
        break;  
    ...  
    case <constante_n>  
        <comando(s)>;  
        break;  
    default:  
        <comando(s)>;  
}
```

## Armadilha:

- Ao esquecer um break no comando switch, o compilador não emitirá mensagem de erro, mas não será efetuado o que você desejava.



# Comando SWITCH



- O switch testa a variável e executa a declaração cujo case corresponda ao valor atual da variável.
- O default é opcional e será executado apenas se a variável que está sendo testada não for igual a nenhuma das constantes.
- O comando break faz com que o switch seja interrompido assim que uma das declarações seja executada.

# Comando SWITCH

- Faça um algoritmo que solicite ao usuário dois valores, e qual operação básica ele deseja que seja realizada. Exiba o resultado da operação.

```
1  #include <locale.h>
2  #include <stdio.h>
3
4  int main() {
5      setlocale(LC_ALL, "Portuguese");
6      char operacao;
7      float num1, num2;
8      printf("Entre com o primeiro valor: ");
9      scanf("%f", &num1);
10     printf("Entre com o segundo valor: ");
11     scanf("%f", &num2);
12     fflush(stdin);
13     printf("Qual operação (+, -, /, *)? ");
14     scanf("%c", &operacao);
15     switch (operacao) {
16         case '+':
17             printf("\n%.2f + %.2f = %.2f", num1, num2, num1+num2);
18             break;
19         case '-':
20             printf("\n%.2f - %.2f = %.2f", num1, num2, num1-num2);
21             break;
22         case '*':
23             printf("\n%.2f * %.2f = %.2f", num1, num2, num1*num2);
24             break;
25         case '/':
26             printf("\n%.2f / %.2f = %.2f", num1, num2, num1/num2);
27             break;
28         default:
29             printf("\nOperação Inválida.");
30     }
31     return 0;
32 }
```

# Comando SWITCH



- Faça um algoritmo que solicite ao usuário uma letra e informe se a mesma é uma vogal ou uma consoante.

```
1  #include <locale.h>
2  #include <stdio.h>
3  #include <ctype.h>
4
5  int main() {
6      setlocale(LC_ALL, "Portuguese");
7
8      char letra;
9
10     printf("Qual a letra? ");
11     scanf("%c", &letra);
12     letra = toupper(letra); //Transforma o caracter em maiúsculo.
13
14     switch (letra) {
15         case 'A': case 'E': case 'I': case 'O': case 'U':
16             printf("\n%c é uma Vogal.", letra);
17             break;
18         case 'B': case 'C': case 'D': case 'F': case 'G':
19         case 'H': case 'J': case 'K': case 'L': case 'M':
20         case 'N': case 'P': case 'Q': case 'R': case 'S':
21         case 'T': case 'V': case 'X': case 'W': case 'Y': case 'Z':
22             printf("\n%c é uma Consoante.", letra);
23             break;
24         default:
25             printf("\n%c Não é uma Letra.", letra);
26     }
27     return 0;
28 }
```

# Comando SWITCH



```
1  #include <locale.h>
2  #include <stdio.h>
3
4  int main() {
5      setlocale(LC_ALL, "Portuguese");
6
7      int mes;
8
9      printf("Qual mês deseja saber a quantidade de dias? ");
10     scanf("%d", &mes);
11
12     switch(mes) {
13         case 1: case 3: case 5: case 7: case 8: case 10: case 12:
14             printf("\nEste mês possui 31 dias.");
15             break;
16         case 4: case 6: case 9: case 11:
17             printf("\nEste mês possui 30 dias.");
18             break;
19         case 2:
20             printf("\nFevereiro possui 29 dias em ano bissexto, e 28 em ano normal.");
21             break;
22         default:
23             printf("\nEste número não corresponde a um mês válido.");
24     }
25
26     return 0;
27 }
```