Estruturas de Controle

Seleção ou Desvio Condicional

a) Seleção Simples:

Permite a escolha de um grupo de ações (bloco) a ser executado quando determinadas condições, representadas por expressões lógicas ou relacionais são ou não satisfeitas.
Sintaxe usada:

```
if (condição) {
        comandos
}
```

b) Seleção Composta:

Acontece quando em uma dada situação existem duas alternativas que dependem de uma mesma condição. Se a condição for verdadeira, um caminho será seguido, se for falsa, outro caminho será tomado.
Sintaxe usada:

```
if (condição) {

    comandos

}else {

    comandos

}
```

c) Estrutura de Decisão Aninhada ou Encadeada:

Existem casos em que é necessário estabelecer verificação de condições sucessivas, em que uma determinada ação poderá ser executada se um conjunto anterior de instruções ou condições for satisfeito. Sendo a ação executada, ela poderá ainda estabelecer novas condições. Isso significa utilizar uma condição dentro de outra condição. Esse tipo de estrutura poderá possuir diversos níveis de condição, sendo chamadas de aninhadas ou encadeamentos.
Sintaxe usada:

```
if (condição1) {
        comando1;
}else if (condição2) {
        comando2;
}else if (condição3) {
        comando 3;
}else {
        comando 4;
}
```

O uso de estruturas de seleção encadeada envolve normalmente problemas com diversas condições. Veja a necessidade de seu uso a partir do seguinte problema:
Escreva um algoritmo para imprimir o conceito final de um aluno a partir de uma média. A tabela de conversão a ser utilizada é a seguinte:

Conceito “A” se media >= 9	Conceito “C” se media >= 6
Conceito “B” se media >= 7	Conceito “P” se media < 6

Aplicando a técnica de interpretação de enunciado:
SAÍDA:
Quais os resultados que o algoritmo deve fornecer?
Saída: Conceito final

Próximo passo **ENTRADA:**
Quais os insumos necessários para se obter a saída?
Precisa conhecer a média do aluno. Dá para supor que a media será fornecida automaticamente pelo programa, mas vamos fazer que seja fornecida pelo usuário diretamente.
ENTRADA de DADOS:

media

Passos para obter os insumos necessários para saída de dados:
Conceito “A” se media >= 9;
Conceito “B” se media >= 7;
Conceito “C” se media >= 6;
Conceito “P” se media < 6;

Estrutura de seleção:

```
if (media >= 9.0) {
        printf(“Conceito A”);
}

if (media >= 7.0) {
        printf(“Conceito B”);
}

if (media >= 6.0) {
        printf(“Conceito C”);
}

if (media < 6.0) {
        printf(“Conceito P”);
}
```

Simulando a execução do programa teremos o seguinte:
Declaração de Variáveis:

```
float media = 9.5;
```

Instruções de Entrada de dados:

```
printf(“Digite a media: ”);
scanf(" %f ",&media);
```

Na sequencia temos as estrutura de seleção com os testes de condição.

```
if (media >= 9.0) {
        printf(“Conceito A”);
}

if (media >= 7.0) {
        printf(“Conceito B”);
}

if (media >= 6.0) {
        printf(“Conceito C”);
}

if (media < 6.0) {
        printf(“Conceito P”) ;
}
```

Simulando a execução do programa quais conceitos obteriam como saída, se o usuário digitar valor 9.5 para media?

A resposta teria que ser Conceito A, no entanto a solução imprime três mensagens:

Conceito A
Conceito B } resultados de saída.
Conceito C

Assim executa os três primeiros blocos de instruções devidas serem verdadeiro. Isso ocorre devido ao uso de quatro estruturas de seleção simples ao invés de seleção composta com else. Outra possibilidade seria fazer uso de seleção composta com operadores lógicos verificando os intervalos. Por exemplo:

```
float media = 9.5;
        if (media >= 9.0) {
                printf("Conceito A");
        }

        if (media >= 7 && media < 9) {
                printf("Conceito B");
        }

        if (media >= 6 && media < 7) {
                printf("Conceito C");
        }

        if (media <  6) {
                printf("Conceito P");
        }
```

Com essa estrutura somente o conceito “A” será exibido o qual está correto. No entanto essa estrutura ainda não é conveniente. Ela é eficaz nas ainda não é ideal. Nesse caso como temos vários Ifs, precisamos da estrutura de seleção aninhada ou encadeada.

ESTRUTURAS DE SELEÇÃO ENCADEADAS são sequencias de estruturas If...else if...else... usados para testes de múltiplos casos encadeados.

```
float media = 9.5 ;
```

```
if (media >= 9.0){
        printf("Conceito A");
}else if (media >= 7.0) {
        printf("Conceito B");
}else if (media >= 6.0) {
        printf("Conceito C");
}else{
        printf("Conceito P");
}
```

Exercícios sobre estrutura de decisão aninhada ou encadeada:

1) Elaborar um algoritmo que verifique qual dos dois valores digitado pelo usuário é maior, ou se os mesmo são iguais. Observem o exercício resolvido na figura 1

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


int main(){

    setlocale(LC_ALL,"portuguese");
    int n1, n2;
    printf("Informe um valor: ");
    scanf("%d",&n1);
    printf("Informe o segundo valor: ");
    scanf("%d",&n2);
    if(n1>n2){
        printf("%d é maior que %d",n1,n2);
    }else if(n2>n1){
        printf("%d é maior que %d",n2,n1);
    }else{
        printf("Os valores são iguais ");
    }

    return 0;
}
```

Figura 1 - Exercício 1 em linguagem de programação C

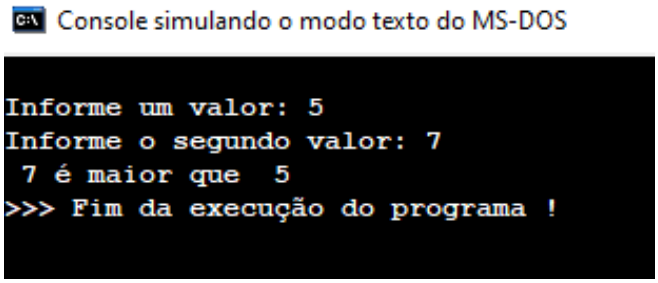

Figura 2 - Exercício 1, console, entrada e saída de dados

Após a execução (F9) do algoritmo no VISUALG, conforme ilustrado na figura 1, com a entrada dos dois valores nas variáveis n1, n2, faz uma verificação na primeira condição (n1>n2), que se lê, o valor de n1 é maior que o valor da variável n2. Na figura 2, observem que o valor que a variável n1 recebe é o valor 5 e a variável n2 valor 7. Na execução do algoritmo, a leitura das linhas de códigos se dá de cima para baixo, dá esquerda para direita, linha após linha, de forma sequencial. E se tratando de estrutura de seleção, a partir disso, faz com que execute ou não um determinado bloco de código. Na linha 12 da

figura 1, a primeira condição resulta em falso, por que o valor 5 não é maior que o valor 7. Isso faz que não entre no primeiro "se" e pule para a próxima condição (n2>n1) na linha 15 da figura 1, retornando verdadeiro, por que 7 é maior que 5 e entra nesse bloco e executa o comando printf("%d é maior que %d",n2,n1). Após executar o comando a leitura do código pula para o final do if.

2) Elaborar um algoritmo que lê um número e identifica se ele é positivo, negativo ou nulo.

Obs.: ***A seguir, estrutura de decisão utilizando operadores lógicos:***

3) Elaborar um algoritmo que lê três números e escrever o maior. Observem o exercício resolvido na figura 3.

```
1  #include <stdio.h>
2  #include <locale.h>
3
4
5  int main(){
6
7      setlocale(LC_ALL,"portuguese");
8      int n1, n2, n3, acumulador = 0 ;
9      printf("Informe três números inteiros: ");
10     scanf("%d %d %d",&n1,&n2,&n3);
11
12     if(n1>=n2 && n1>=n3){
13         acumulador = n1;
14     }else if(n2>n1 && n2>=n3){
15         acumulador = n2;
16     }else{
17         acumulador = n3;
18     }
19         printf("O maior número é %d",acumulador);
20     return 0;
21 }
```

Figura 3 – Exercício 3 em linguagem de programação C

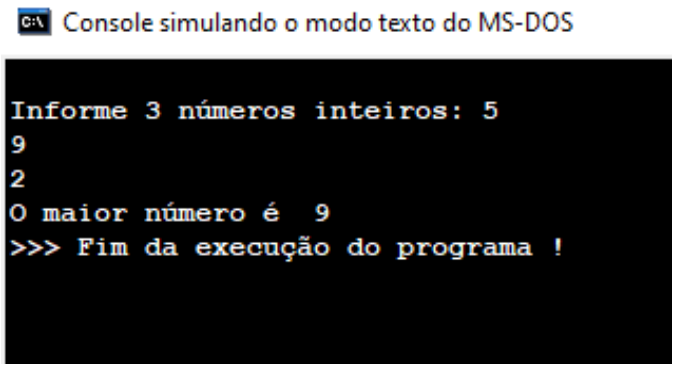

Figura 4 - Exercício 3, console, entrada e saída de dados

Conforme figura 3, na linha 10 onde tem a primeira verificação referente aos valores das variáveis ((n1>=n2) && (n1>=n3)) , usando como referencia para essa explicação a figura 4, observem que a variável **n1** recebe o valor 5, **n2** o valor 9 e **n3** o valor 2, vai retornar o valor falso, ou seja, não entra nesse bloco. Por que retorna falso? Como as expressões estão ligadas pelo operador lógico " **&&** " e substituindo na representação ((n1>=n2) && (n1>=n3)) pelos valores de cada variável pelo respectivo algarismo que receberam na entrada dos dados:

$$\underbrace{((5>=9)}_{\text{falso}} \;\&\&\; \underbrace{(5>=2))}_{\text{verdadeiro}}$$

falso e verdadeiro = falso

Quando se usa operador lógico E (&&), essa conjunção só vai ser verdadeira se todas as expressões forem verdadeiras. Basta uma ser falsa para retornar falso. O operador lógico E, é somatório, todas as expressões matemáticas tem que serem verdadeiras para o resultado final ser verdadeiro.
Por exemplo, se trocarmos a primeira condição (5>=9) pela proposição A e a segundo (5>=2)) pela proposição B.
Vou usar essas proposições (A e B) em uma tabela-verdade para representar graficamente a saída de dados, ou resultado:
Como substitui as 2 expressões pelas proposições A e B, para fazer a tabela-verdade, vai ser $2^2 = 2 \times 2 = 4$, quatro linha na tabela. O que isso significa? Significa base 2 os (0 e 1) em base binária, ou V e F, são os valores que trabalhamos, não existe um terceiro valor.
Ou representamos por 0 e 1 ou podemos representar esses valores com V ou F (Verdadeiro ou Falso).
E o valor no expoente 2? Representa o número de proposições que envolvem na condição.

A	B	A e B
V	V	V
V	F	F
F	V	F
F	F	F

Observem que as duas condições tem que ser verdadeiras para termos um resultado verdadeiro. Conforme a primeira condição analisada do exercício 3, ((5>=9) && (5>=2)) que resulta em falso, podemos observar na tabela-verdade acima que está representado, na linha 3 em laranja. Essas quatro linha da tabela-verdade representa os possíveis resultados a partir dos valores de entrada das variáveis (n1,n2).

Se trocarmos o operador lógico E (&&) por OU (||) como fica com base no exemplo acima?

$$\underbrace{((5>=9)}_{\text{falso}} \;||\; \underbrace{(5>=2))}_{\text{verdadeiro}} = \text{verdadeiro}$$

A	B	A ou B
V	V	V
V	F	V
F	V	V
F	F	F

Observem que agora basta uma das proposições ser verdadeira para o resultado ser verdadeiro. Operador OU (||), o próprio nome sugere, ou um ou outro já basta para resultar em verdadeiro.

ATENÇÃO: Aconselho ler na apostila algoritmo.pdf, ou fazerem uma pesquisa na internet sobre tabela-verdade para um melhor entendimento de condições envolvendo operadores lógicos.

4) Construir um algoritmo que leia duas notas e calcule a média final. Após o cálculo, escreva o resultado final de acordo com os dados:

condição	resultado
media >= 7	aprovado
media >= 5 e media < 7	exame
media < 5	reprovado

5) Algoritmo que lê três notas de um aluno e imprimir a media com os dados de acordo com os dados abaixo:

Condição	resultado
media >= 8	Ótimo
media >= 7 e media < 8	Bom
media >= 5 e media < 7	Regular
media < 5	Péssimo

6) Elaborar um algoritmo para uma calculadora básica. Ler dois valores e perguntar qual das operações básicas matemática irá ser realizada.

7) Algoritmo que verifique se um número digitado pelo usuário está na faixa dos 20 e 90.