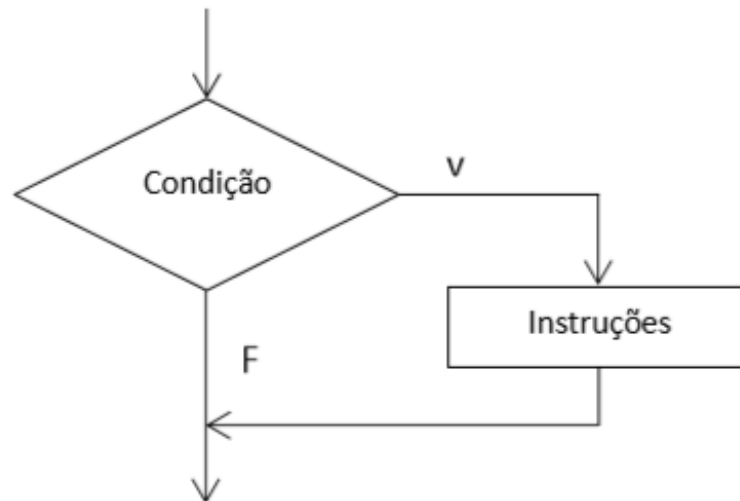


ESTRUTURA DE SELEÇÃO OU ESTRUTURA DE DECISÃO

a) Estrutura de Seleção Simples

Estrutura de decisão é um desvio condicional que é usado para decidir se um conjunto de instruções deve ou não ser realizado. Dependendo de um resultado de um teste de condição executa ou não um conjunto de comando ou instruções.

Fluxograma de "estrutura de decisão simples"



Como é que se formula um teste de condição?

Uma condição é formulada com uso de operadores relacionais.

Operador	Ação
>	Maior do que
>=	Maior ou igual a
<	Menor do que
<=	Menor ou igual a
==	Igual a
!=	Diferente de

Os operadores relacionais retornam verdadeiro(1) ou falso (0).

Para verificar o funcionamento dos operadores relacionais, execute o programa abaixo em linguagem de programação em C:

```
/* Este programa ilustra o funcionamento dos operadores relacionais. */
```

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

```

    setlocale(LC_ALL,"portuguese");
    int i, j;
    printf("\nEntre com dois números inteiros: ");
    scanf("%d%d", &i, &j);
    printf("\n %d == %d é %d \n", i, j, i==j);
    printf("\n %d != %d é %d \n", i, j, i!=j);
    printf("\n %d <= %d é %d \n", i, j, i<=j);
    printf("\n %d >= %d é %d \n", i, j, i>=j);
    printf("\n %d < %d é %d \n", i, j, i<j);
    printf("\n %d > %d é %d \n", i, j, i>j);
    return(0);
}

```

Na figura 1 a seguir, mostra o exemplo acima feito no DEV-C++, que é a ferramenta utilizada em aula para compilar o código, ou seja, para verificar se está correto.

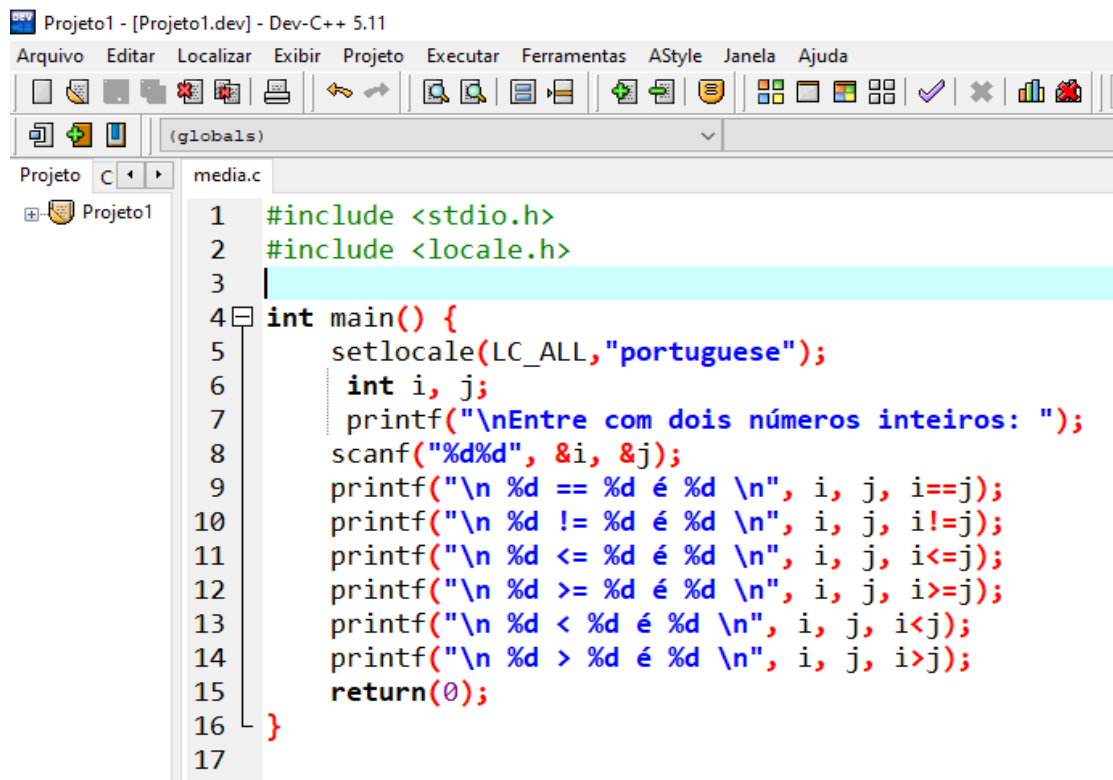


Figura 1 - DEV-C++

Como representar de forma textual uma estrutura de seleção simples ou desvio condicional simples na linguagem de programação em C?

O comando **if** representa uma tomada de decisão do tipo "se (condição) então execute esse ou esses comandos".

A sua forma geral é: **if** (condição) declaração;

A condição do comando **if** é uma expressão que será avaliada. Se o resultado for zero a declaração não será executada. Se o resultado for qualquer coisa diferente de zero a declaração será executada. A declaração pode ser um bloco de código

ou apenas um comando. É interessante notar que, no caso da declaração ser um comando(uma única linha de comando a ser executado), não é necessário o uso de chaves { e }.

Abaixo um exemplo:

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

int main() {
    setlocale(LC_ALL,"portuguese");
    int num;
    printf ("Digite um número: ");
    scanf ("%d",&num);

    if (num>10)
        printf ("\nO número e maior que 10");

    if (num==10) { //inicio o if
        printf ("\n\nVocê acertou!\n");
        printf ("O número e igual a 10.");
    } //fim do if

    if (num<10)

        printf ("\n\nO número e menor que 10");

}
```

Na figura 2, mostra o exemplo acima feito no DEV-C++

```

1  #include <stdio.h>
2  #include <locale.h>
3
4  int main() {
5      setlocale(LC_ALL, "portuguese");
6      int num;
7      printf ("Digite um número: ");
8      scanf ("%d",&num);
9
10     if (num>10)
11         printf ("\nO número e maior que 10");
12
13     if (num==10) { //inicio o if
14         printf ("\n\nVocê acertou!\n");
15         printf ("O número e igual a 10.");
16
17     } //fim do if
18
19     if (num<10)
20
21         printf ("\n\nO número e menor que 10");
22
23 }
24

```

Figura 2 - DEV-C++

No programa acima a expressão `num>10` é avaliada e retorna um valor diferente de zero, se verdadeira, e zero, se falsa. No exemplo, se `num` for maior que 10, será impressa a frase: "O número e maior que 10". Repare que, se o número for igual a 10, estamos executando dois comandos. Para que isto fosse possível, tivemos que agrupa-los em um bloco que se inicia logo após a comparação e termina após o segundo `printf`. Repare quando queremos testar igualdades usamos o operador `==` e não `=`. Isto porque o operador `=` representa apenas uma atribuição à variável.

Pode parecer estranho à primeira vista, mas se escrevêssemos da seguinte maneira:

```
if (num=10) ...    /* Isto está errado */
```

Outros exemplos:

Escreva um programa para calcular a média de um aluno a partir de duas notas. O programa deve exibir a média e também uma mensagem informando se o aluno foi aprovado. Sabe-se que o aluno é aprovado se sua média é igual ou superior a 6.

Saída:

Quais os resultados que o algoritmo deve fornecer?

média e mensagem (aprovado).

Entrada:

Quais os insumos necessários para obter a saída?
notas (nota1 e nota2)

Processamento:

Como transformar os insumos na saída?
 $media = (nota1 + nota2) / 2$

Como verificar se foi aprovado?

Assim o programa completo para exibir a média do aluno e se ele foi aprovado é inicialmente declara as variáveis:
media, nota1, nota2: real

Veja na prática conforme figura 3, como fica na linguagem de programação em C o cálculo da média do aluno.

```
[*] media.c
1  #include <stdio.h>
2  #include <locale.h>
3
4  int main()
5  {
6      setlocale(LC_ALL, "portuguese");
7
8      float nota1, nota2, media;
9      printf("Digite primeira nota: ");
10     scanf("%f", &nota1);
11     printf("Digite segunda nota: ");
12     scanf("%f", &nota2);
13     media = (nota1 + nota2) / 2;
14
15     if(media >= 6.0) {
16         printf("Aprovado ");
17     }
18     printf(" média = %.2f ", media); // 2.2 significa imprimir com 2 casas decimais
19
20     return 0;
21 }
22
```

Figura 3 - Programa do cálculo da média na estrutura simples if.

Quando executar o programa (F11) e dar entrada das 2 notas conforme exemplo mostra a mensagem Aprovado, se média for maior ou igual a 6.

```
Digite primeira nota: 5,5
Digite segunda nota: 8,5
Aprovado média = 7,00
-----
Process exited after 12.12 seconds with return value 0
Pressione qualquer tecla para continuar. . .
```

Figura 4 - Tela de saída de dados

Caso as duas notas de entradas no processamento não se obter valor da média maior ou igual a 6, aparecerá somente a média como saída de dados. Veja exemplo na figura 5 dessas entradas que ocasionou média inferior a 6.

```
Digite primeira nota: 7,0
Digite segunda nota: 3,5
média = 5,25
-----
Process exited after 12.06 seconds with return value 0
Pressione qualquer tecla para continuar. . .
```

Figura 5 - Tela de saída de dados.

A expressão

```
if (media >= 6.0){
    printf(" Aprovado ")
}
```

Será avaliada a condição **media >= 6.0** sendo a condição verdadeira, o comando `printf(" Aprovado ")` será executado.

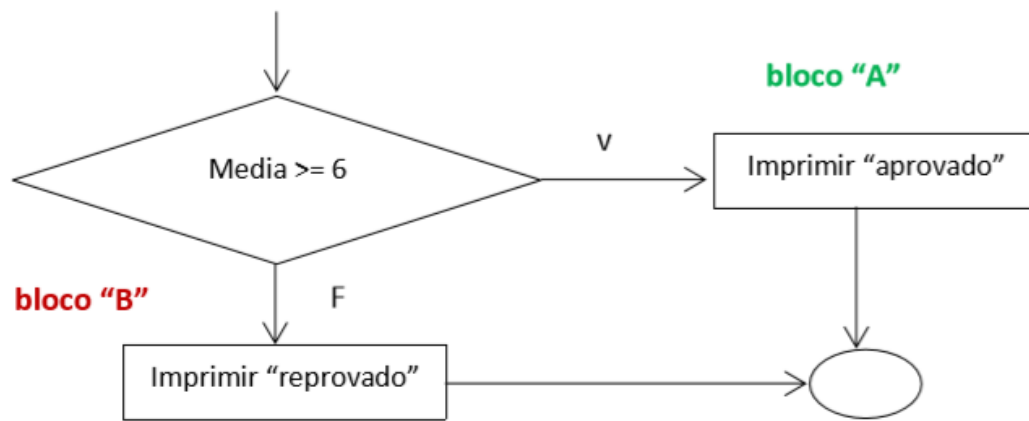
Caso for falsa, não executa a instrução (não entra na condição do if), segue a leitura das linhas de códigos ao final do if.

b) Estrutura de Seleção Composta

Podemos nos deparar com situações em que haja mais de um bloco para ser executado ou desviado de acordo com o resultado do teste de condição, neste caso precisamos fazer uso de estrutura de seleção composta.

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.

Fluxograma "estrutura de decisão ou seleção composta"



Desvio Condicional Composto, uma condição é avaliada e se o resultado for verdadeiro a instrução A1... An de comandos serão executados. Se o resultado for falso a instrução B1... Bn serão executados.

Veja, como representar uma estrutura de seleção composta em representação textual em português estruturado, portugol ou pseudocódigo:

```

if (condição) {
    instrução A1
    instrução A2
    instrução A3
    ...
    comando An
} else{
    instrução B1
    instrução B2
    instrução B3
    ...
    Instrução Bn
}
  
```

Exemplo:

Escreva um algoritmo para calcular a média de um aluno a partir de duas notas. O programa de exibir a média e também uma mensagem informando se o aluno foi aprovado ou reprovado. Sabe-se que o aluno é aprovado se sua média é igual ou superior a 6.

Saída:

Quais os resultados que o algoritmo deve fornecer?
média e mensagem (aprovado ou reprovado).

Entrada:

Quais os insumos necessários para obter a saída?
notas (nota1 e nota2)

Processamento:

Como transformar os insumos na saída?

$media = (nota1 + nota2) / 2$

Assim, o programa completo para imprimir a media e a mensagem se ele foi aprovado ou reprovado é a seguinte:

Inicialmente declara as variáveis: nota1, nota2, media: real

Veja na prática conforme figura 6 a seguir, como fica em português estruturado o algoritmo do cálculo da média do aluno.

```
media.c
1  #include <stdio.h>
2  #include <locale.h>
3
4  int main() {
5      setlocale(LC_ALL, "portuguese");
6
7      float nota1, nota2, media;
8      printf("Digite primeira nota: ");
9      scanf("%f", &nota1);
10     printf("Digite segunda nota: ");
11     scanf("%f", &nota2);
12     media = (nota1 + nota2) / 2;
13
14     if (media >= 6.0) {
15         printf("Aprovado ");
16     } else {
17         printf("Reprovado ");
18     }
19
20     printf(" média = %.2f ", media); // 2.2 significa imprimir com 2 casas decimais
21
22     return 0;
23 }
24
```

Figura 6 - Programa completo do cálculo da média em (if...else).

Quando executar o algoritmo (F11) e dar entrada das 2 notas conforme exemplo se obtém a média seguido da mensagem Aprovado, se média for maior ou igual a 6, caso contrário, reprovado.

```
Digite primeira nota: 6,6
Digite segunda nota: 6,0
Aprovado média = 6,30
-----
Process exited after 7.285 seconds with return value 0
Pressione qualquer tecla para continuar. . .
```

Figura 7 - Tela de saída de dados.

Observem, quando se usa estrutura de seleções simples ou composta, o programador tem de usar a técnica de tabular o código, fazer endentação entre a

estrutura e as linhas de instruções (o que é para ser executado dependendo da condição verdadeira(1) ou falsa(0)).

Exemplo:

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

Isso é essencial para futuras alterações nas linhas de códigos do sistema, possibilitando uma melhor compreensão para o profissional, principalmente se tratando de códigos longos.

Lembrando:

A função printf() tem a seguinte forma geral:

```
printf (string_de_controle,lista_de_argumentos);
```

Teremos, na string de controle, uma descrição de tudo que a função vai colocar na tela. A string de controle mostra não apenas os caracteres que devem ser colocados na tela, mas também quais as variáveis e suas respectivas posições. Isto é feito usando-se os códigos de controle, que usam a notação %. Na string de controle indicamos quais, de qual tipo e em que posição está as variáveis a serem apresentadas. É muito importante que, para cada código de controle, tenhamos um argumento na lista de argumentos. Apresentamos agora alguns dos códigos %:

Código	Significado
%d	Inteiro
%f	Float
%c	Caractere
%s	String
%%	Coloca na tela um %

Vamos ver alguns exemplos de **printf()** e o que eles exibem:

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

```
int main() {  
    setlocale(LC_ALL,"portuguese");
```

// barra barra significa uma única linha de comentário, não surge efeito no código

/* esse exemplo é para mais de uma linha de

comentário */

```
printf ("Teste %% %% \n"); //-> "Teste % %"
printf ("%f \n",40.345); //-> "40.345"
printf ("Um caractere %c e um inteiro %d \n",'D',120); //-> "Um caractere D e um
inteiro 120"
printf ("%s e um exemplo \n","Este"); //-> "Este e um exemplo"
printf ("%s%d%% \n","Juros de ",10); // -> "Juros de 10%"
return 0;
}
```

no DEV-C++:

```
1  #include <stdio.h>
2  #include <locale.h>
3
4  int main() {
5      setlocale(LC_ALL,"portuguese");
6      //barra barra significa uma única linha de comentário, não surge efeito no código
7      /* esse exemplo é para mais de uma linha de
8      comentário */
9
10     printf ("Teste %% %% \n"); //-> "Teste % %"
11     printf ("%f \n",40.345); //-> "40.345"
12     printf ("Um caractere %c e um inteiro %d \n",'D',120); //-> "Um caractere D e um inteiro 120"
13     printf ("%s e um exemplo \n","Este"); //-> "Este e um exemplo"
14     printf ("%s%d%% \n","Juros de ",10); // -> "Juros de 10%"
15 }
16
```

Figura 8 - Parâmetro no comando de saída printf()

O formato geral da função scanf() é:

scanf (string-de-controle,lista-de-argumentos);

Usando a função scanf() podemos pedir dados ao usuário. Um exemplo de uso, pode ser visto acima. Mais uma vez, devemos ficar atentos a fim de colocar o mesmo número de argumentos que o de códigos de controle na string de controle. Outra coisa importante é lembrarmos de colocar o & antes das variáveis da lista de argumentos, isso mostra o endereço onde está a variável na memória do computador razão para este procedimento.

Exercícios

1) Elaborar um programa em linguagem C que leia dois valores numéricos reais desconhecidos. Em seguida o programa deve efetuar a adição dos dois valores lidos e apresentar o resultado caso seja maior que 10.

2) Elaborar um programa em linguagem C que leia três notas de um aluno. Em seguida o programa deve efetuar o cálculo da média desse aluno e apresentar o resultado APROVADO se a media maior ou igual a 7.0.

3) Elabore o programa em C que efetua a leitura de um valor numérico, verifique se este valor é par ou ímpar e apresente o resultado dessa verificação na tela.

4) Elaborar um programa em linguagem C que leia dois valores numéricos reais desconhecidos. Em seguida o programa deve efetuar a adição dos dois valores lidos e caso o resultado seja maior ou igual a 10, deve ser somado ao resultado obtido o valor 5, obtendo-se novo resultado. Caso contrário, o valor do resultado inicialmente obtido deve ser subtraído de 7, gerando-se outro resultado. Após a obtenção de um dos novos resultados o novo resultado deve ser apresentado.

5) Elabore um programa em linguagem C que simule um caixa eletrônico quando vamos sacar dinheiro. O caixa eletrônico verifica se o valor que desejamos sacar é menor que o saldo disponível. Assumiremos que há R\$ 2.000,00 de saldo disponível para o saque. Se valor do saque for menor que o saldo disponível, subtrair do saldo disponível o valor do saque e apresente o valor do saque. Caso contrário, informar que o valor solicitado é maior que o valor disponível para saque. Independente se tem ou não saldo suficiente para saque o algoritmo ao final deve mostrar o saldo disponível.

6) Elabore um programa em linguagem C para ler um número e verificar se ele é par ou ímpar. Quando for par armazenar esse valor em P e quando for ímpar armazená-lo em I. Exibir P e I no final do processamento.

7) Elaborar um programa em C para ler uma variável numérica N e imprimi-la somente se a mesma for maior que 50, caso contrário imprimi-la com o valor zero.

8) Os números inteiros são os números positivos e negativos. Estes números formam o conjunto dos números inteiros, indicado por Z. O conjunto dos números inteiros é infinito e pode ser representado da seguinte maneira: $Z = \{..., -3, -2, -1, 0, 1, 2, 3, \dots\}$ Os números inteiros negativos são sempre acompanhados pelo sinal (-), enquanto os números inteiros positivos podem vir ou não acompanhados de sinal (+). O zero é um número neutro, ou seja, não é um número nem positivo e nem negativo. Com base nisso, elabore um programa que leia um número. Se positivo armazene-o em A, se for negativo, em B. No final mostrar o resultado.