



COMPUTAÇÃO BÁSICA

Estruturas Condicionais

Prof. Bruno Macchiavello (bruno@cic.unb.br)

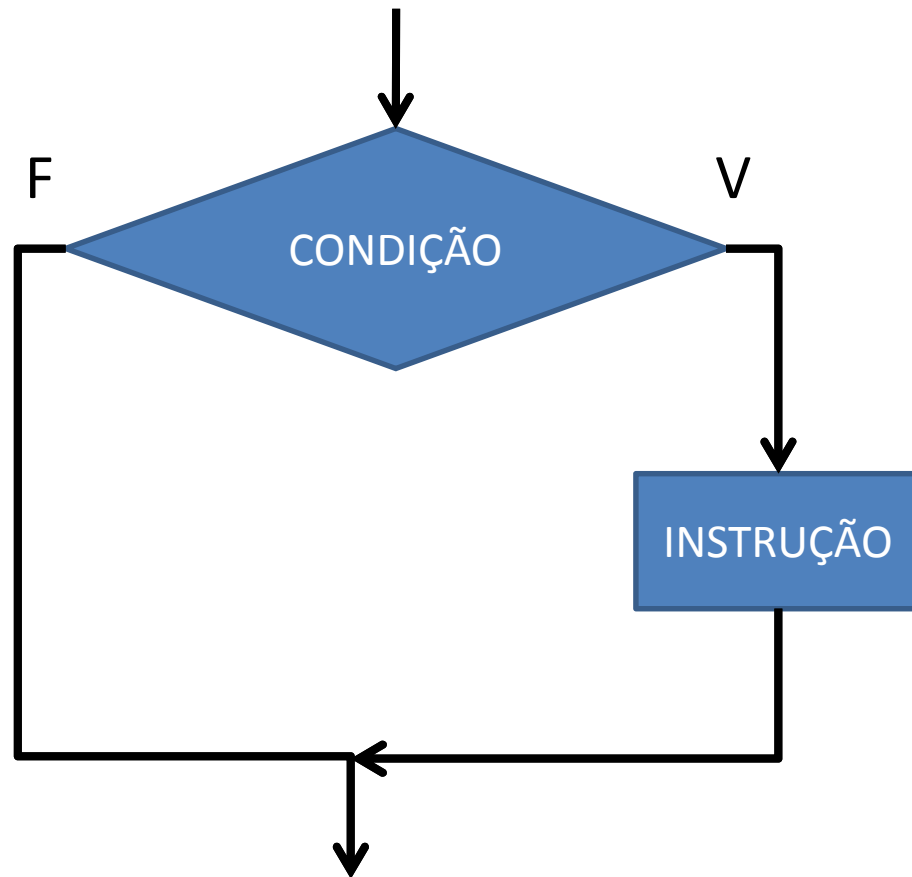
Universidade de Brasília – UnB
Instituto de Ciências Exatas – IE
Departamento de Ciência da Computação – CIC



Estruturas Condicionais

- Em alguns algoritmos a execução é direta, ou seja, todos os passos do algoritmo são executados em uma seqüência pré-definida. As instruções são executadas de cima para baixo e de esquerda para a direita, ou seja, de forma SEQUÊNCIAL.
- Porém em alguns casos DECISÕES precisam ser tomadas, assim como testes de CONSISTÊNCIA.
 - Exemplo de Decisão: Se esta fazendo frio vou colocar um casaco.
 - Exemplo de teste de Consistência: Verificar uma operação de divisão que o divisor seja diferente de zero.

Estruturas Condicionais Simples



- O comando (ou instrução) será executado somente se a condição for verdadeira.
- A condição nada mais é do que uma expressão lógica de comparação
- A comparação lógico pode ser utilizada para fazer uma decisão (exemplo: temperatura > 30°), ou para realizar um teste de consistência (divisor != 0)



Estruturas Condicionais Simples

- Em Pseudocódigo a estrutura condicional simples é:

Estrutura Condicional Simples
Se <condição> então <comando>

- Se mais de um comando deve ser executado caso a <condição> seja verdadeira, então esses comandos devem ser transformados em um comando composto (indicando o Fim-Se).

Estrutura Condicional Simples
Se <condição> então <comando 1> <comando 2> <comando 3> Fim-Se



Estruturas Condicionais Simples

- Em ANSI C a estrutura condicional simples fica:

Estrutura Condicional Simples
<pre>if (<condição>) <comando>;</pre>

- Se mais de um comando deve ser executado caso a <condição> seja verdadeira:

Estrutura Condicional Simples
<pre>if (<condição>) { <comando 1>; <comando 2>; <comando 3>; }</pre>



Estruturas Condicionais Simples

- Exemplo: Faça um algoritmo que receba o salário base de um funcionário, calcule e mostre o salário a receber, sabendo que esse funcionário tem gratificação de 5% sobre o salário base e se o salário com gratificação for maior que R\$ 1000 o funcionário paga imposto de 7% sobre o total.



Estruturas Condicionais Simples

Exemplo:

Algoritmo Salario

Variáveis

salBase, grat, imposto, salFinal : **real**

Início

Escreva ("Informe o salario base: ")

Leia (salBase)

grat \leftarrow salBase * 0.05

imposto \leftarrow 0

Se (salBase+grat) > 1000 **então**

 imposto \leftarrow (salBase + grat) * 0.07

salFinal \leftarrow salBase + grat - impostos

Escreva ("O salario a receber é: ", salFinal)

Fim



Estruturas Condicionais Simples

Exemplo:

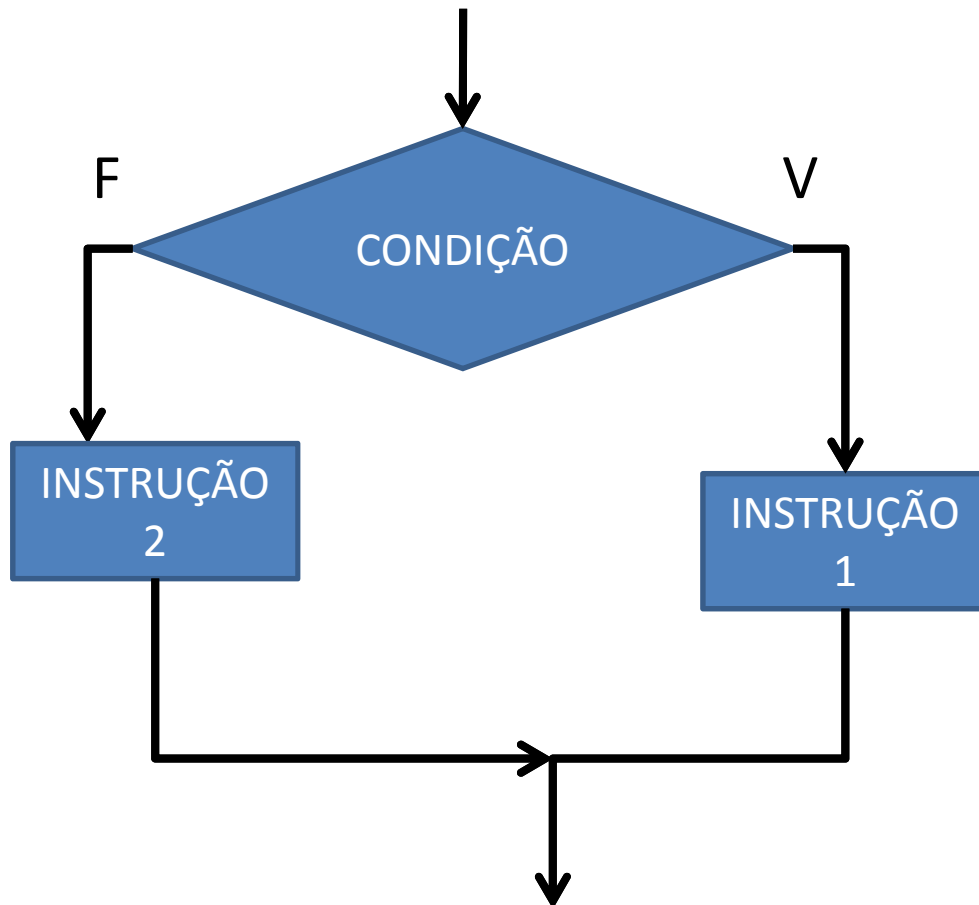
```
#include <stdio.h>
int main()
{
    float salBase, salFinal, grat, imposto=0;

    printf("Informe o salario Base:\n");
    scanf("%f",&salBase);
    grat = salBase * 0.05;

    if ( (salBase+grat) > 1000)
        imposto = (salBase + impostos) * 0.07

    salFinal = salBase + grat – impostos;
    printf("O salario a receber é: %.2f", salFinal);
    return(1);
}
```

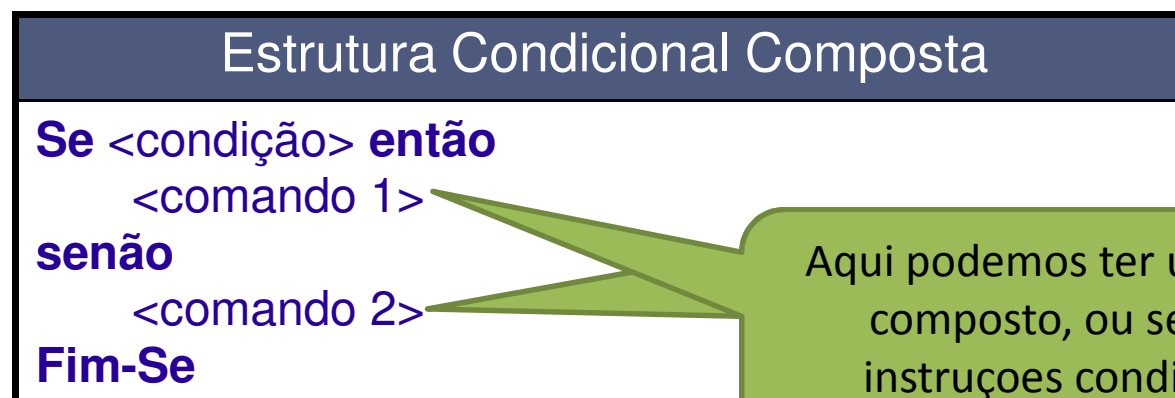

Estruturas Condicionais Composta



- A instrução 1 será executada se a expressão for verdadeira, senão a instrução 2 será executada

Estruturas Condicionais Composta

- Em Pseudocódigo a estrutura condicional composta fica:



Aqui podemos ter um comando composto, ou seja, várias instruções condicionadas.



Estruturas Condicionais Composta

- Na linguagem ANSI C fica:

```
if (<condição>
    <comando1>
else
    <comando2>
```

- No caso de vários comandos:

```
if (<condição>)
{
    <comando1>;
    <comando2>;
}
else
{
    <comando3>;
    <comando4>;
}
```



Estrutura Condicional Composta

Exemplo:

Algoritmo que calcula a área de um triângulo, dados os comprimentos dos lados A, B, e C.

Algoritmo AreaTri

Variáveis

A, B, C, P, area : **real**

Início

Leia (A)

Leia (B)

Leia (C)

Se $(A < B + C)$ e $(B < A + C)$ e $(C < B + A)$ **então**

$P \leftarrow (A + B + C) / 2$

area \leftarrow

Escreva (area)

senão

Escreva ("Os numeros A, B e,C nao formam um triangulo.")

Fim-se

Fim



Estrutura Condicional Composta

Programa em C do algoritmo anterior

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

int main () {
    float A, B, C, P, area;

    printf("Informe os lados do triângulo: \n");
    scanf("%f",&A);
    scanf("%f",&B);
    scanf("%f",&C);
    if ((A<B+C) && (B<A+C) && (C<B+A)) {
        P = (A+B+C)/2;
        area = sqrt(P*(P-A)*(P-B)*(P-C));
        printf("Area do triângulo: %.2f\n",area);
    }
    else
        printf("Os números A, B, e C não formam um triângulo\n");
    return(1);
}
```



Estrutura Condicional Composta

- Notar que as seguintes estruturas são equivalentes

Se <condição1> **então**
 <comando1>
Senão
 <comando2>

Se <!condição1> **então**
 <comando2>
Senão
 <comando1>

! = Negação



Operador ternário “?”

- Este operador de ANSI C pode ser usado para simplificar expressões de condicional composta, somente quando usamos uma única instrução.

- Exemplo: A expressão:

```
if (A > 0)
    B = -50;
else
    B = 100;
```

- Pode ser simplificada como

```
B = A > 0 ? -50 : 100;
```

Encadear Estruturas Condicionais

- Muita vezes é necessário realizar testes ou decisões mais de uma vez.

Exemplo de Encadeamento de Estruturas Condicionais

```
Se <condição1> então  
    Se <condição2> então  
        Se <condição3> então  
            <comando1>  
        Senão  
            <comando2>  
    Senão  
        <comando3>  
Senão  
    <comando4>  
Fim-Se
```

comando1 => executado quando condição1 = V, condição2 = V, condição3 = V

comando2 => executado quando condição1 = V, condição2 = V, condição3 = F

comando3 => executado quando condição1 = V, condição2 = F

comando4 => executado quando condição1 = F



Encadear Estruturas Condicionais

- Um bom programa deve usar **indentação**!

```
Se <condição1> então  
Se <condição2> então  
Se <condição3> então  
<comando1>  
Senão  
<comando2>  
Senão  
<comando3>  
Senão  
<comando4>  
Fim-Se
```



```
Se <condição1> então  
  Se <condição2> então  
    Se <condição3> então  
      <comando1>  
    Senão  
      <comando2>  
  Senão  
    <comando3>  
Senão  
  <comando4>  
Fim-Se
```



Encadear Estruturas Condicionais

- Exemplo: Faça um algoritmo que calcule a média ponderada de um aluno, a partir de suas 3 notas obtidas no curso, sabendo-se que a primeira avaliação tem peso 2, a segunda tem peso 4 e a terceira tem peso 4, e mostre a situação do aluno sendo que:
 - APROVADO caso a nota final esteja no intervalo $[10, 7]$
 - RECUPERAÇÃO.... caso a nota final esteja no intervalo $[5, 7)$
 - REPROVADO caso a nota final esteja no intervalo $[0, 5)$



Encadear Estruturas Condicionais

Algoritmo MediaAluno

Variaveis

aluno: **literal**

nota1, nota2, nota3, media: **real**

Inicio

Escreva (“Informe o nome do aluno”)

Leia (aluno);

Escreva (“Informe as 3 notas do aluno”)

Leia (nota1, nota2, nota3);

$media \leftarrow nota1 * 0.2 + nota2 * 0.4 + nota3 * 0.4$

Escreva (“a media de “, aluno, “foi “, media)

Se (media \geq 7,0) **então**

Escreva (“aprovado”)

senão

Se (media $<$ 7,0 e media \geq 5,0) **então**

Escreva (“recuperação”)

senão

Escreva (“reprovado”)

Fim-se

Fim



Encadear Estruturas Condicionais

Exemplo:

Dados três números inteiros distintos (assume-se que o serão), elabore um algoritmo que escreva o maior número digitado.

Algoritmo MaiorNúmero

Variáveis

A, B, C : inteiro

Início

Leia (A, B, C)

Se (A>B) então

Se (A>C) então

Escreva ("O maior número é A")

Senão

Escreva ("O maior número é C")

Senão

Se (B>C) então

Escreva ("O maior número é B")

Senão

Escreva ("O maior número é C")

Fim-Se

Fim



Encadear Estruturas Condicionais

- Podemos melhorar a solução utilizando um outro tipo de composição das estruturas condicionais, junto com condições mais elaboradas:

Algoritmo MaiorNúmero

Variáveis

A, B, C : inteiro

Início

Leia (A, B, C)

Se (A>B) e (A>C) **então**

Escreva (“O maior número é A”)

Senão Se (A<B) e (B>C) **então**

Escreva (“O maior número é B”)

Senão

Escreva (“O maior número é C”)

Fim-Se

Fim



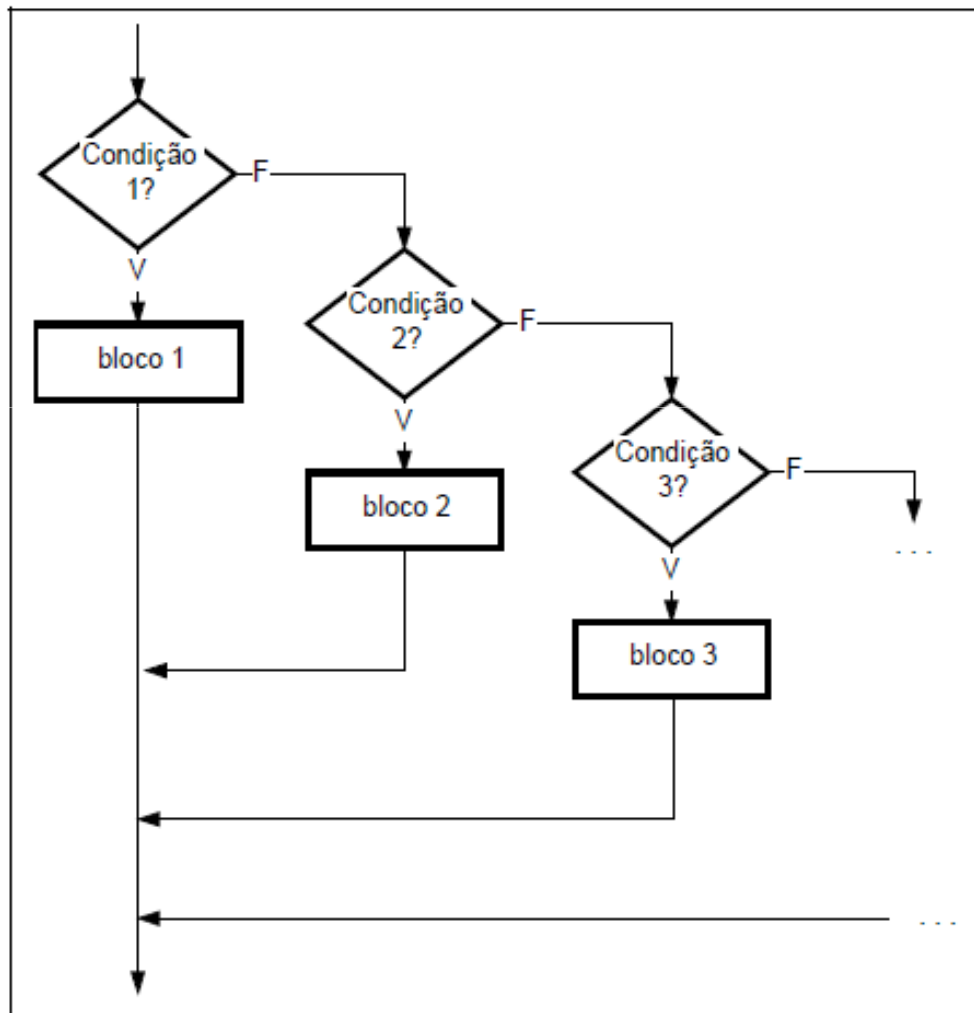
Encadear Estruturas Condicionais

- Note que as seguintes estruturas são equivalentes:

```
Se <condição1> então  
    <comando1>  
Senão  
    Se <condição2> então  
        <comando2>  
    Senão  
        <comando3>  
Fim-Se
```

```
Se <condição1> então  
    <comando1>  
Senão Se <condição2> então  
    <comando2>  
Senão  
    <comando3>  
Fim-Se
```

Estrutura de Seleção



- Uma instrução será executada somente se a sua respectiva condição for verdadeira.



Estrutura de Seleção

Estrutura Seleção (utilizando Se – Senão Se – Senão)

```
Se <condição1> então  
    <comando 1>  
Senão se <condição2>  
    <comando 2>  
Senão se <condição3>  
    <comando 3>  
    .  
    .  
    .  
Senão se <condição n>  
    <comando n>  
Senão  
    <instrução padrão>  
Fim-Se
```

- A instrução padrão serão executada caso todas as condições sejam falsas.
- A condição padrão é opcional



Estrutura de Seleção

Estrutura Seleção (utilizando caso)

```
Caso <identificador>  
  igual <valor1> então  
    <comando 1>  
  igual <valor2>  
    <comando 2>  
  igual <valor3>  
    <comando 3>  
  .  
  .  
  .  
  igual <valor n>  
    <comando n>  
Senão  
  <instrução padrão>  
Fim-Caso
```

- O comando **caso** compara uma variável de nome <identificador> com uma lista de valores (<valor1, valor2,...,valor n>) quando o valor da variável seja igual a um determinado valor da lista então certos comandos serão executados.



Estrutura de Seleção em ANSI C

Estrutura Seleção (utilizando Se – Senão Se – Senão)

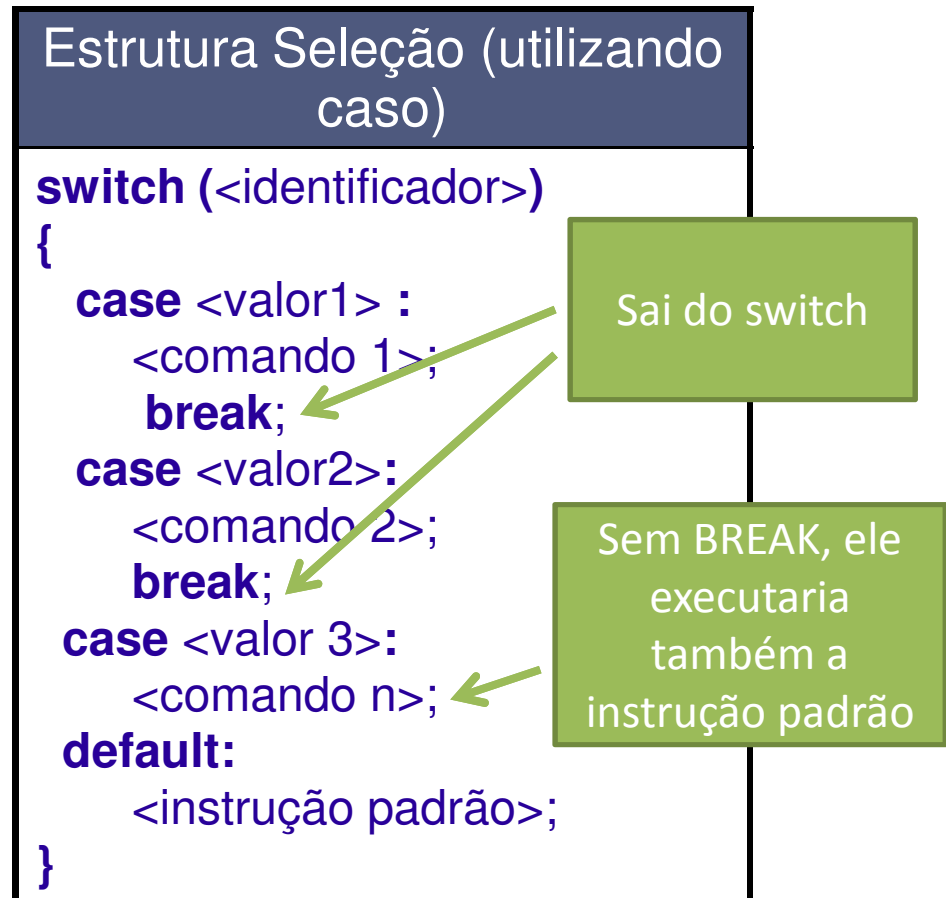
```
If (<condição1>)  
    <comando 1>;  
else if (<condição2> )  
    <comando 2>;  
else if (<condição3> )  
    <comando 3>;  
.  
.  
.  
else if (<condição n>)  
    <comando n>;  
else  
    <instrução padrão>
```

Estrutura Seleção (utilizando caso)

```
switch (<identificador>)  
{  
    case <valor1> :  
        <comando 1>;  
    case <valor2>:  
        <comando 2>;  
    case <valor3> :  
        <comando 3>;  
    .  
    .  
    .  
    case <valor n>:  
        <comando n>;  
    default:  
        <instrução padrão>;  
}
```

Estrutura de Seleção em ANSI C

- A estrutura **switch** normalmente é utilizada juntamente com o comando **break**.
- Se não se coloca **break** no final de cada bloco de comandas, então serão executados todos os comandos após o primeiro deles ser executado.





Exemplo

- Existem diversos pacotes numa academia: o pacote básico, pacote plus, pacote completo e o pacote com pilates. O pacote básico inclui somente o uso de aparelhos de musculação, o pacote plus além disso inclui as aulas de ginástica o pacote completo inclui também as aulas de spinning e o pacote com pilates é o pacote básico mas as aulas de pilates. Faço um algoritmo que monte o custo dos pacotes dado que:
 - Musculação: R\$100
 - Ginástica: R\$50
 - Spinning: R\$20
 - Pilates: R\$80



Algoritmo Academia

Variáveis

valor, pacote: **inteiro**

mus, pilates, ginastica, spinning: **inteiro**

Início

mus \leftarrow 100

ginastica \leftarrow 50

spinning \leftarrow 20

pilates \leftarrow 80

Escreva ("Escolha o pacote que deseja montar:")

Escreva ("1 – Pacote Básico")

Escreva ("2 – Pacote Plus")

Escreva ("3 – Pacote Completo")

Escreva ("4 – Pacote com Pilates")

Leia (pacote)

Caso (pacote)

igual 1 então

 valor \leftarrow mus

igual 2 então

 valor \leftarrow mus + ginastica

igual 3 então

 valor \leftarrow mus + ginastica + spinning

igual 4 então

 valor \leftarrow mus + pilates

Fim-caso

Escreva("O valor do pacote é:", valor)

Fim



```
#include <stdio.h>

int main ()
{
    int valor, pacote;
    const int mus=100, pilates=80, ginastica=50, spinning=20;
    printf ("Escolha o pacote que deseja montar:\n");
    printf ("1 – Pacote Básico\n");
    printf ("2 – Pacote Plus\n");
    printf ("3 – Pacote Completo\n");
    printf ("4 – Pacote com Pilates\n");
    scanf ("%d",&pacote);
    switch (pacote)
    {
        case 1:
            valor = mus; break;
        case 2:
            valor = mus + ginastica; break;
        case 3:
            valor = mus + ginastica + spinning; break;
        case 4:
            valor = mus + pilates; break;
    }
    printf("O valor do pacote é: %d\n", valor);
    getchar();
    return(1);
}
```



Algumas Funções

Função	Sintaxe	Exemplo	=	Objetivo	Linguagem C
QUOC	QUOC (A,B)	QUOC (10,3)	3	Retornar a parte inteira de uma divisão	Basta a variável que recebe o resultado da operação ser do tipo inteiro.
MOD()	MOD(A,B)	MOD(10,3)	1	Retornar o resto de uma divisão.	%
^	A^B	2^3	8	Elevar um número a um expoente	pow()
FRAC()	FRAC(A)	FRAC(10.345)	0.345	Retornar a parte fracionária de um número real.	Feito através da subtração do valor pela sua parte inteira e armazenando o resultado em um variável real.
INTEIRO()	INTEIRO(A)	INTEIRO(10.345)	10	Retornar a parte inteira de um número real.	Dado um valor real, armazene-o em uma variável inteira e então, apenas a parte inteira deste valor é armazenada.



Algumas Funções

Função	Sintaxe	Exemplo	=	Objetivo	Linguagem C
ARRED()	ARRED(A)	ARRED(10.4)	10	Arredondar um número. Até 0.4 retorna para baixo, de 0.5 a 0.9 arredonda para cima.	Para arredondamento acima ceil(). Para arredondamento abaixo floor()
RAIZ()	RAIZ(A)	RAIZ(16)	4	Retorna a raiz quadrada de um número.	sqrt()



Exercícios

1. Faça um algoritmo que receba dois números inteiros e verifique se são números pares ou ímpares. Também imprima uma mensagem indicando se os dois números são iguais, ou indique qual é o maior dos dois números.
2. Faça um algoritmo que leia três números diferentes (assuma que o sejam) e os imprima na tela em ordem crescente.
3. Faça um algoritmo que exibirá uma lista de operações matemáticas (Adição, Subtração, Multiplicação, Divisão e Potenciação). O usuário escolherá o número da opção. O programa pedirá então dois valores e fará a operação entre eles, mostrando o resultado em seguida. Nota: Verificar NÃO dividir por zero
4. Faça um algoritmo que leia a data de nascimento de uma pessoa (dia, mês e ano, todos inteiros). Verifique se a data está correta. Imprima uma mensagem indicando ao usuário se a data está correta, ou uma mensagem indicando qual é o erro, se a data estiver incorreta. Um exemplo de erro: 31/02/2003 - o mês 02 não pode ter 31. DESCONSIDERAR ANOS BISSEXTOS
5. O banco do Brasil abriu uma linha de crédito para os seus funcionários. O valor máximo da prestação não poderá ultrapassar 30% do salário bruto. Fazer um algoritmo que permita entrar com o salário bruto e o valor da prestação e informar se o empréstimo pode ou não ser concedido.