

CAPÍTULO 3

Estruturas Condicionais

Simple, Composta, Encadeada e Caso

(if-else ; if else if ; switch)

ESTRUTURAS CONDICIONAIS.....	36
ESTRUTURA CONDICIONAL SIMPLES	36
SINTAXE DA ESTRUTURA CONDICIONAL SIMPLES EM ALGORITMOS	37
SINTAXE DA ESTRUTURA CONDICIONAL SIMPLES EM PROGRAMAS EM C++	37
ESTRUTURA CONDICIONAL COMPOSTA	38
SINTAXE DA ESTRUTURA CONDICIONAL COMPOSTA EM ALGORITMOS.....	39
SINTAXE DA ESTRUTURA CONDICIONAL COMPOSTA EM PROGRAMAS EM C++	39
ESTRUTURAS CONDICIONAIS ENCADEADAS	42
IFS ANINHADOS	43
ESTRUTURA CONDICIONAL DE SELEÇÃO MÚLTIPLA (CASO)	45
SINTAXE DA ESTRUTURA CONDICIONAL DE MÚLTIPLA ESCOLHA EM ALGORITMOS	45
SINTAXE DA ESTRUTURA CONDICIONAL DE MÚLTIPLA ESCOLHA EM PROGRAMAS EM C++	45
EXERCÍCIOS PROPOSTOS DO CAPÍTULO 3:	48

CAPÍTULO 3

ESTRUTURAS CONDICIONAIS

Dentro dos programas temos comandos especiais utilizados quando temos que tomar determinada decisão baseado no valor de uma variável ou de uma expressão. Estas estruturas são chamadas de desvio condicional, pois desviam o fluxo do programa para um determinado bloco de comandos de acordo com uma condição. Existem basicamente duas estruturas deste tipo que são o **if** e o **switch**.

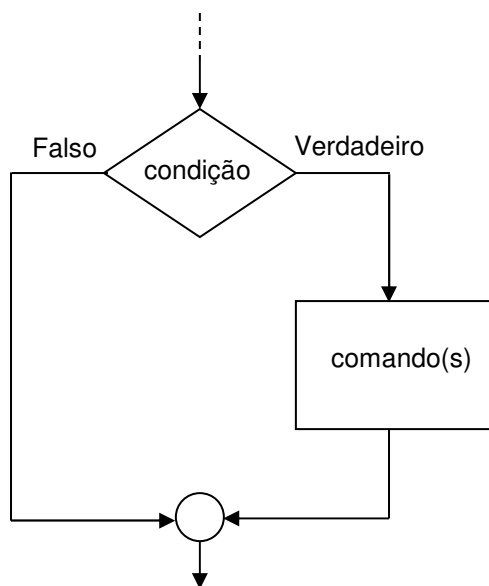
A estrutura condicional **if** pode ser abordada de 3 formas: estrutura condicional simples, composta ou encadeada:

ESTRUTURA CONDICIONAL SIMPLES

Quando necessitarmos de situações onde uma alternativa depende de uma condição para ser realizada, usamos a estrutura condicional simples.

Nesta estrutura, se a condição for verdadeira, serão executados os comandos que estão dentro do bloco verdadeiro (*true*); caso contrário, se a condição for falsa não será executado nenhum comando.

A estrutura condicional simples pode ser vista, em termos de funcionamento (semântica), da seguinte forma:



O bloco **comando** só será executado **se** o resultado da **condição** for verdadeiro.

A **condição** deve ser uma comparação relacional e/ou lógica entre variáveis e/ou resultados de expressões. O resultado da condição será um dos valores *booleanos* possíveis (*true* ou *false*).

Observe que se o resultado da **condição** for verdadeiro, podemos executar um bloco de comandos, se for falso encerra-se a estrutura condicional simples sem executar nenhum comando.

SINTAXE DA ESTRUTURA CONDICIONAL SIMPLES EM ALGORITMOS

```
se (condição)
    então comando;
```

Na sintaxe acima, podemos executar somente um único comando dentro da cláusula então.

Quando necessitarmos executar mais de um comando dentro da cláusula então, devemos seguir a seguinte sintaxe, abrir o bloco início/fim:

```
se (condição)
    então início
        comando 1;
        comando 2;
        ....
        comando n;
    fim;
```

SINTAXE DA ESTRUTURA CONDICIONAL SIMPLES EM PROGRAMAS EM C++

```
if (condição)
    comando;
```

Na sintaxe acima, podemos executar somente um único comando dentro da estrutura.

Quando necessitarmos executar mais de um comando dentro da estrutura, devemos seguir a seguinte sintaxe, abrir o bloco { } (início/fim):

```
if (condição)
{
    comando 1;
    comando 2;
    ....
    comando m;
}
```

Exemplo 1) Escreva um programa para calcular o salário final de um funcionário, sendo que se o funcionário tem mais de 5 anos de serviço, ele tem 10% a mais no salário base. O salário base e o tempo de serviço são fornecidos pelo usuário.

```
int main ()    // programa exemplo1
{
    float salario;        // salário do funcionário
    int tempo;           // tempo de serviço em anos

    // entrada de dados
    cout << "Entre com o salário base do funcionário ";
    cin >> salario;
    cout << "Entre com o tempo de serviço (em anos) ";
    cin >> tempo;

    // processamento
```

```

if (tempo > 5)
    salario = salario + (salario * 0.1);

// saída de dados
cout << "O salário do funcionário é " << salario << endl;

return 0;
}

```

Exercício 3.1) Após a execução do seguinte trecho de programa:

```

d = 0;

if ((a <= b) && (c > b))
    d = 5;

cout << d << endl;

```

Na variável d estará armazenado o valor de 5 se:

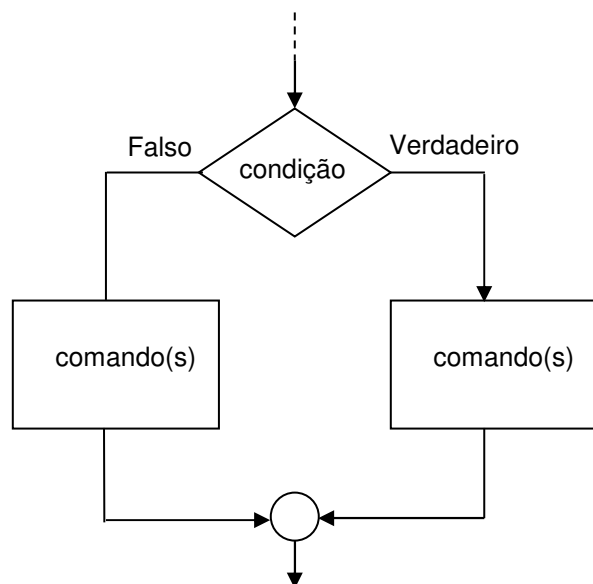
- a) $a \leq b \leq c$
- b) $b < c < a$
- c) $a < b \leq c$
- d) $a \leq b < c$
- e) Nenhuma das alternativas

ESTRUTURA CONDICIONAL COMPOSTA

Quando necessitarmos de situações onde duas alternativas dependem de uma mesma condição, usamos a estrutura condicional composta.

Nesta estrutura, se a condição for verdadeira, serão executados os comandos que estão dentro do bloco verdadeiro; caso contrário, se a condição for falsa os comandos dentro do bloco falso serão executados.

A estrutura condicional composta pode ser vista, em termos de funcionamento da seguinte forma:



SINTAXE DA ESTRUTURA CONDICIONAL COMPOSTA EM ALGORITMOS

```

se (condição)
    então comando;
    senão comando;
  
```

Semelhante à estrutura condicional simples, quando necessitarmos executar mais de um comando dentro das cláusulas *então* ou *senão*, basta utilizarmos o delimitador de bloco início-fim, como mostrado a seguir:

```

se (condição)
    então início
        comando 1;
        comando 2;
        ....
        comando n;
    fim;
    senão início
        comando 1;
        comando 2;
        ....
        comando n;
    fim;
  
```

SINTAXE DA ESTRUTURA CONDICIONAL COMPOSTA EM PROGRAMAS EM C++

```

if (condição)
    comando;
else
    comando;
  
```

Semelhante à estrutura condicional simples, quando necessitarmos executar mais de um comando dentro das cláusulas *true* ou *false*, basta utilizarmos o delimitador de bloco o { } (início/fim), como mostrado a seguir:

```

if (condição)
{
    comando 1;
    comando 2;
    ....
    comando m;
}
else
{
    comando 1;
    comando 2;
    ....
    comando n;
}
  
```

Exemplo 2) Modifique o Exemplo 1 de forma que, caso o funcionário não tenha mais que 5 anos de trabalho, seja mostrada uma mensagem informando que o mesmo não tem direito ao bônus.

```
int main ()    // programa exemplo2
{
    float salario;        // salário do funcionário
    int tempo;            // tempo de serviço em anos

    // entrada de dados
    cout << "Entre com o salário base do funcionário ";
    cin >> salario;
    cout << "Entre com o tempo de serviço (em anos) ";
    cin >> tempo;

    // processamento e saída
    if (tempo > 5)
    {
        salario = salario + (salario * 0.1);
        cout << "O salário do funcionário é " << salario << endl;
    }
    else cout << "O funcionário não tem direito ao bônus " << endl;

    return 0;
}
```

Exemplo 3) Escreva um programa para ler o nome e a idade de uma pessoa e informar se a pessoa é maior ou menor de idade. O nome e a idade são fornecidos pelo usuário.

```
int main ()    // programa exemplo3
{
    string nome;        // nome da pessoa
    int idade;          // idade em anos

    // entrada de dados
    cout << "Qual é o seu nome? ";
    cin >> nome;
    cout << nome << ", qual é a sua idade? ";
    cin >> idade;

    // processamento e saída
    if (idade < 18)
        cout << nome << ", você é menor de idade !";
    else cout << nome << ", você é maior de idade !";

    return 0;
}
```

Exemplo 4) Escreva um programa para ler dois valores, determinar e imprimir a relação entre eles (de igualdade, ou não).

```

int main ()    // programa exemplo4
{
    float num1, num2;           // números quaisquer
    // entrada de dados
    cout << "Entre com os 2 números ";
    cin >> num1 >> num2;

    // processamento e saída
    if (num1 == num2)
        cout << num1 << " e " << num2 << " são iguais";
    else cout << num1 << " e " << num2 << " são diferentes";

    return 0;
}

```

Exercício 3.2) Indique a saída (qual o valor de x?) dos seguintes trechos de programas, considerando os seguintes valores: a= 2, b = 3, c = 5 e d = 9.

- a) **if** (! (b <= c) || (c >= 3) && (a != b))
 x = (d+1) % (c - b);
 else x = **pow**(c - **pow**(a,2), (1/3));
 cout << x;
- b) **if** (!(a >= 2) || !(b > 7) && (c >= 6) || !(a <= b))
 x = **pow**(a,2) - **sqrt**(d) - 2;
 else x = **exp** (3 * **log**(b)) - d;
 cout << x;

Exercício 3.3) Os blocos (1 e 2) de comandos descritos abaixo, são equivalentes? Ou seja, quando executados, **oferecem o mesmo resultado para a variável sim ?**

Considere as variáveis a e b numéricas e a variável sim, booleana.

(1)

```

cout << "Entre com valores para a e b ";
cin >> a >> b;

sim = (a == b);
sim = !(sim);

cout << "Resultado " << sim;

```

(2)

```

cout << "Entre com valores para a e b ";
cin >> a >> b;

if (a != b)
    sim = false;
else sim = true;
sim = !(sim);

cout << "Resultado " << sim;

```

Responda a pergunta acima, **justificando a sua resposta** dando valores para a e b.

ESTRUTURAS CONDICIONAIS ENCADEADAS

Quando necessitamos agrupar várias estruturas condicionais (simples ou composta), formamos então uma **estrutura condicional encadeada**. Normalmente, isto ocorre quando uma determinada ação deve ser executada se várias condições ou combinações de condições forem satisfeitas.

Exemplo 5) Escreva um programa para ler dois valores, determinar e imprimir a relação entre eles (de igualdade, maior ou menor).

Note que podemos construir uma estrutura condicional com e sem encadeamento, conforme exemplos 5a) e 5b) a seguir:

Exemplo 5a) Estrutura sem encadeamento

```
int main ()    // programa maior
{
    float num1, num2;        // números quaisquer

    // entrada de dados
    cout << "Entre com os 2 números ";
    cin >> num1 >> num2;

    // processamento e saída
    if (num1 == num2)
        cout << num1 << " e " << num2 << " são iguais";
    if (num1 > num2)
        cout << num1 << " é maior que " << num2 ;
    if (num2 > num1)
        cout << num2 << " é maior que " << num2 ;

    return 0;
}
```

Exemplo 5b) Estrutura com encadeamento

```
int main ()    // programa maior
{
    float num1, num2;        // números quaisquer

    // entrada de dados
    cout << "Entre com os 2 números ";
    cin >> num1 >> num2;

    // processamento e saída
    if (num1 == num2)
        cout << num1 << " e " << num2 << " são iguais";
    else if (num1 > num2)
        cout << num1 << " é maior que " << num2 ;
    else cout << num2 << " é maior que " << num1 ;

    return 0;
}
```

Note que podemos construir uma estrutura condicional encadeada de diversas formas, e sem limites de encadeamento.

Exemplos 6 (de encadeamentos):**a)**

.....

```

if (condição1)
{
    if (condição2)
    {
        W;
        X;
    }
}
else if (condição3)
{
    Y;
    Z;
}
else if (condição4)
{
    if (condição5)
    {
        A;
        B;
    }
}
else
{
    C;
    D;
}

```

Exercício 3.5) No exemplo acima, o que será executado se:

a) condição1 for verdadeira e condição2 for verdadeira:

b) condição1 for verdadeira e condição2 for falsa:

c) condição1 for falsa e condição3 for verdadeira:

d) condição1 for falsa e condição3 for falsa e condição4 for falsa:

e) condição1 for falsa e condição3 for falsa e condição4 for verdadeira e condição5 for falsa:

f) quais as condições para executar A e B?

b)....

```

if (condição1)
    if (condição2)
        if (condição3)
            if (condição4)
                ....
            }
        }
    }

```

Note que neste último tipo de encadeamento, podemos escrever a mesma estrutura da seguinte forma:

```

if (condição1 && condição2 && condição3 && condição4)
{
    ....
}

```

IF's ANINHADOS

Quando você tem um certo número de **if**(s) e **else**(s), como saber (como o computador decide) qual **else** é de qual **if** ?

Por exemplo, considere o trecho de programa a seguir:

```

if (x > y)
    if (y != 0)
        a = x/y;
else
    a = y;

```

Quando será executada a instrução `a = y` ? Ou seja, o **else** está associado ao primeiro ou ao segundo **if** ?

O **else** está sempre associado ao mais recente **if** sem **else**. Então, se `x` não for maior que `y`, ou seja, `x <= y`, nada será executado. Ou seja:

```

if (x > y)
    if (y != 0)
        a = x/y;
    else
        a = y;

```

Caso não seja isso o desejado, devemos usar chaves, para abertura e fechamento de bloco de comandos:

```

if (x > y)
{
    if (y != 0)
        a = x/y;
}
else
    a = y;

```

ESTRUTURA CONDICIONAL DE SELEÇÃO MÚLTIPLA (CASO)

Quando vários valores (de igualdade) precisam ser testados e ações diferentes são executadas para cada um dos valores, podemos usar a estrutura de seleção múltipla. Este comando seletivo (CASO: **switch**) é o mais indicado quando existem comandos mutuamente exclusivos a serem executados, ou seja, se um comando for executado, os demais não serão. Esta situação é bastante comum na construção de algoritmos.

O comando CASO (**switch**) difere do SE (**if**) no sentido de que ele só testa igualdade, enquanto que, o SE (**if**) pode avaliar uma expressão lógica ou relacional. O CASO (**switch**) tem um formato mais limpo e claro.

SINTAXE DA ESTRUTURA CONDICIONAL DE SELEÇÃO MÚLTIPLA (CASO) EM ALGORITMOS

```

caso (expressão) ou (variável)
    seja valor1: comando;
    seja valor2: comando;
    ...
    seja valorn: comando;
    senão comando;
fim caso;
  
```

Onde:

- **caso**, **seja** e **fim caso** são palavras reservadas
- **expressão** é qualquer resultado numérico (inteiro) ou literal
- **variável** é do tipo numérico (inteiro) ou literal
- **valor1**, **valor2**,..... **valorn** podem assumir um número (uma constante inteira) ou um valor literal.
Exemplos: 7 'a' 24 'PAULO' 0 'Z'
- se dentro de **comando** existir mais de uma ação, o bloco (**início-fim**) deverá ser inserido
- o **senão** é opcional para representar uma alternativa que significa “em nenhum dos casos anteriores”

SINTAXE DA ESTRUTURA CONDICIONAL DE SELEÇÃO MÚLTIPLA (CASO) EM PROGRAMAS EM C++

```

switch (VARIÁVEL) ou (CONSTANTE)
{
    case VALOR1:
        comando1;
        comando2;
        break;
    case VALOR2:
        comando3;
        comando4;
        break;
    case VALORN:
        comandoN;
        break;
    default:
        comando7;
        comando8;
}
  
```

Veja seu funcionamento:

- O comando **switch** consiste na palavra-chave **switch** seguida do nome de uma variável ou de um valor numérico constante entre parênteses.
- O corpo de um comando **switch** deve estar entre chaves.
- O corpo do comando **switch** é composto de vários casos que devem ser rotulados com uma constante e opcionalmente um caso *default*.
- A expressão entre parênteses após a palavra **switch** determina para qual *case* será desviado o controle do programa.
- O corpo de cada *case* é composto por qualquer número de instruções. As instruções do corpo de cada *case* não precisam estar entre chaves { }. Em geral, a última instrução é o *break*. O comando *break* causa a saída imediata de todo o corpo do **switch**.
- Na falta do comando *break*, todas as instruções *case*, a partir do caso escolhido até o término do comando, serão executadas, mesmo sendo pertencentes aos casos seguintes.
- Você não poderá usar uma variável ou uma expressão lógica como rótulo de algum *case* (no lugar da constante).
- Se o rótulo (*case*/constante) for igual ao valor da expressão do comando **switch**, a execução começará nele. Se nenhum caso for satisfeito e existir um caso *default*, a execução se dá nele (*default*). Não pode haver *cases* com rótulos iguais.

Na sintaxe acima apresentada:

- caso o valor da VARIÁVEL ou CONSTANTE seja igual a VALOR1, os comandos 1 e 2 serão executados e o comando **break** causará a saída do comando **switch**;
- caso o valor da VARIÁVEL ou CONSTANTE seja igual a VALOR2, os comandos 3 e 4 serão executados e o comando **break** causará a saída do comando **switch**;
- e assim sucessivamente, até:
- caso o valor da VARIÁVEL ou CONSTANTE seja igual a VALORN, o comandoN será executado, e o comando **break** causará a saída do comando **switch**;
- o caso **default** é opcional, e quando usado, o comando (ou os comandos) indicados nele, por exemplo, 7 e 8, serão executados no caso de todas as demais condições anteriores não terem sido satisfeitas.

Exemplo 7) Escreva um programa que leia dois números e a operação desejada para esses números: adição (+) ou multiplicação (*). A seguir, mostre o resultado da operação escolhida entre esses dois números.

Implementação usando estrutura if – else:

```
int main ( )    // programa exemplo7 com if-else
{
    float n1, n2;
    int op;
    cout <<"Digite os dois números: ";
    cin >> n1 >> n2;
    cout <<"Digite a operação: 1 p/ soma ou 2 p/ produto: ";
    cin >> op;
    if (op==1)
        cout << endl << n1 << " + " << n2 << " = " << n1+n2 << endl;
    else if (op==2)
        cout << endl << n1 << " * " << n2 << " = " << n1*n2 << endl;
        else cout <<endl << "Você digitou uma operação inválida....." << endl;

    return 0;
}
```

Implementação usando estrutura switch:

```
int main ( )    // programa exemplo7 com switch
{
    float n1, n2;
    int op;
    cout << "Digite os dois números: ";
    cin >> n1 >> n2;
    cout << "Digite a operação: 1 p/ soma ou 2 p/ produto: ";
    cin >> op;
    switch (op)
    {
        case 1: cout << endl << n1 << " + " << n2 << " = " << n1+n2 << endl;
                 break;
        case 2: cout << endl << n1 << " * " << n2 << " = " << n1*n2 << endl;
                 break;
        default: cout << endl << "Você digitou uma operação inválida....." << endl;
    }
    return 0;
}
```

Exemplo 8) Escreva um algoritmo que leia o valor de custo de um produto e o tipo da moeda (1, 2, 3, ou 4) codificado da forma abaixo; e escreva o valor do produto, seguido da moeda.

- 1 – libra esterlina
- 2 – franco suíço
- 3 – dólar americano
- 4 – real

```
int main ( )    // programa exemplo8
{
    float valor;
    int moeda;
    cout << "Qual a moeda? " << endl;
    cout << "1 - libra esterlina" << endl;
    cout << "2 - franco suíço" << endl;
    cout << "3 - dólar americano" << endl;
    cout << "4 - real" << endl;
    cout << "Sua opção: ";
    cin >> moeda;
    cout << endl << "Qual o custo do produto? ";
    cin >> valor;
    switch (moeda)
    {
        case 1: cout << endl << "Custo de " << valor << " libras. " << endl;
                 break;
        case 2: cout << endl << "Custo de " << valor << " francos. " << endl;
                 break;
        case 3: cout << endl << "Custo de " << valor << " dólares. " << endl;
                 break;
        case 4: cout << endl << "Custo de " << valor << " reais. " << endl;
                 break;
        default: cout << endl << "Você digitou uma moeda inválida....." << endl;
    }
    return 0;
}
```

Exercícios propostos do Capítulo 3:

P3.1) Refaça o programa do exemplo 3 (maioridade) e quando a pessoa for menor de idade, apresente quantos anos para atingir a maioridade.

P3.2) Construa um programa que, dado um número, verifique se ele é inteiro.

P3.3) Construa um programa que, dado um número, verifique se ele é par.

P3.4) Uma empresa decide dar um aumento de 3% ao funcionário cujo salário é inferior a 3 salários mínimos. Escreva um programa que efetue o cálculo do salário e escreva o valor do salário reajustado, caso contrário, apresente uma mensagem de que não foi possível reajustar. O salário do funcionário e o valor do salário mínimo são fornecidos pelo usuário.

P3.5) Faça um programa que leia os valores A, B, C e diga se a soma de A + B é menor que C.

P3.6) Faça um programa que leia dois valores A e B. Se os valores forem iguais, o programa deverá somar os dois, caso contrário, multiplicar A por B, e, ao final do cálculo, atribuir o valor para uma variável C.

P3.7) Escreva um programa que permita ao usuário entrar com uma letra qualquer e informar se a letra digitada é uma vogal ou uma consoante.

P3.8) Faça um programa que leia o nome e as suas três notas em uma disciplina e ao final, escreva o nome do aluno, sua média e se ele foi aprovado (a média é 8).

P3.9) Tendo como dados de entrada, a altura (em metros) e o sexo (F/M) de uma pessoa, construa um programa que calcule seu peso ideal, utilizando as seguintes fórmulas:

para homens: $(72.7 * \text{altura}) - 58$

para mulheres: $(62.1 * \text{altura}) - 44.7$

P3.10) Elabore um programa que efetue a entrada de um valor numérico inteiro e verifique se esse valor é divisível por 4. Para se verificar se um valor (N) é divisível por outro (D), efetua-se o cálculo de R da seguinte forma, $R = N - D * \text{floor}(N/D)$. Se R for igual a zero é porque N é divisível por D.

P3.11) Desenvolva um programa que efetue a leitura de 2 números inteiros (N1 e N2) e verifique se eles são divisíveis por 2 e por 3 ao mesmo tempo.

P3.12) Um sinal elétrico é especificado pela seguinte equação:

$$x(t) = \begin{cases} 1 - |t|, & -1 \leq t \leq 1 \\ 0, & \text{caso contrário} \end{cases}$$

Escreva um programa que leia o valor de t, calcule e mostre o valor de x(t).

P3.13) Faça um programa que leia os valores de duas variáveis x e y, calcule e escreva o valor de z, dado por:

$$z = \frac{\sqrt{x^2 + y^2}}{x^2 - y^3}$$

(importante)....O programa deve verificar a possibilidade de divisão antes de fazer o cálculo de Z.

P3.14) Uma empresa decide dar um reajuste a seus funcionários de acordo com os seguintes critérios:

5% para aquele que ganha menos de 10 SM (salário mínimo);

2% para aquele que ganha entre 10 SM e 20 SM (inclusive);

1.5% para os demais

Faça um programa que leia o salário do funcionário, o salário mínimo, calcule e escreva o valor do salário reajustado.

P3.15) Faça um programa que leia 3 números distintos e imprima o menor deles.

P3.16) Dados três valores distintos, faça um programa que, após a leitura destes dados coloque-os em ordem crescente.

P3.17) Faça um programa que receba a hora de início de um jogo (HHMM) e a hora final do jogo (HHMM). Calcule e mostre a duração do jogo em horas e minutos.

P3.18) Assuma que a, b, c e d são variáveis e S1, S2, S3 e S4 são comandos. Expresse as condições necessárias para a execução de S1, S2, S3 e S4 no seguinte comando encadeado:

```

if (a > b)
    if (b <= c)
        if (c != d)
            S1;
        else
            S2;
    else
        S3;
else
    S4;

```

P3.19) Faça um programa que dados dois números A, B e um literal OP (sendo OP um símbolo de operação aritmética: +, -, *, /), calcule e imprima o resultado da operação de A por B. Caso a operação seja de divisão (OP = /) e B = 0, envie uma mensagem de erro para o usuário. Caso OP não seja um símbolo de operação, envie também uma mensagem para o usuário.

P3.20) Assuma que v1, v2, v3 e v4 são variáveis e C1, C2, C3, C4, C5, C6, C7 e C8 são comandos. Sobre o trecho de programa a seguir, responda:

```

if (v1 >= v2)
{
    C1;
    if (v2 < v3)
    {
        C2;
        if (v3 != v4)
            C3;
        else
            C4;
    }
    else
    {
        C5;
        C6;
    }
}
else
    C7;
C8;

```

- a) Se $v1 \geq v2$, $v2 < v3$ e $v3 == v4$, quais os comandos que serão executados?
- b) Se $v1 < v2$, $v2 < v3$ e $v3 == v4$, quais os comandos que serão executados?
- c) Se $v1 \geq v2$, $v2 \geq v3$ e $v3 != v4$, quais os comandos que serão executados?
- d) Há possibilidade de somente o comando C8 ser executado? Explique.

P3.21) Construa um programa que tendo como dados de entrada o preço de um produto e um código de origem, emita o preço junto de sua procedência. Caso o código não seja nenhum dos especificados, o produto deve ser considerado como importado.

Código de origem:

- 1 – SUL
- 2 – SUDESTE
- 3 ou 4 – NORDESTE/NORTE
- 5 – CENTRO-OESTE
- 10 – SUDESTE

P3.22) É dado o gabarito de uma prova que consta de 3 questões, cujas respostas podem ser qualquer uma das 3 alternativas: A, B ou C. São dadas também as respostas que o estudante forneceu para as 3 questões. Desenvolva um programa para imprimir a mensagem “Eliminado” caso o estudante tenha errado todas as questões ou as questões corretas (1, 2 e/ou 3), caso o estudante tenha acertado alguma questão.

Exemplo: Na saída ELIMINADO ou

Questão 1 ou

Questão 2

Questão 3 ou

Questão 2

P3.23) Faça um programa para ler quatro valores referentes a quatro notas escolares de um aluno (de 0 a 10) e imprimir uma mensagem dizendo que o aluno foi aprovado, se o valor da média escolar for maior ou igual a 7. Se o valor da média for menor que 7, solicitar a nota de exame, somar com o valor da média e obter nova média. Se a nova média for maior ou igual a 5, apresentar uma mensagem dizendo que o aluno foi aprovado em exame. Se o aluno não foi aprovado, apresentar uma mensagem também informando esta condição. Apresentar com as mensagens, para qualquer condição, o valor da média do aluno.

P3.24) Construa um programa para calcular e imprimir as raízes x_1 e x_2 de uma equação do 2º. grau, sendo que os coeficientes a , b e c são fornecidos (o coeficiente a deve ser diferente de zero para termos uma equação do 2º. grau). Se o valor de DELTA (Δ) for negativo, envie a mensagem “RAÍZES NÃO REAIS”.

$$\Delta = b^2 - 4ac \qquad x_1 = \frac{-b + \sqrt{\Delta}}{2a} \qquad x_2 = \frac{-b - \sqrt{\Delta}}{2a}$$

P3.25) Dados três valores x , y , z , faça um programa para verificar se eles podem ser os comprimentos dos lados de um triângulo, e se forem, verificar se é um triângulo equilátero, isósceles ou escaleno. Se eles não formarem um triângulo, escrever uma mensagem.

Antes da elaboração do programa, torna-se necessária a revisão de algumas propriedades e definições.

Propriedade - O comprimento de cada lado de um triângulo é menor do que a soma dos comprimentos dos outros dois lados.

Definição 1 - Chama-se triângulo equilátero o que tem os comprimentos dos três lados iguais,

Definição 2 - Chama-se triângulo isósceles o que tem os comprimentos de dois lados iguais.

Definição 3 - Chama-se triângulo escaleno o que tem os comprimentos dos três lados diferentes.

P3.26) Faça um programa que mostre o menu de opções a seguir, receba a opção do usuário e os dados necessários para executar cada operação.

Menu de opções:

1 – Somar dois números

2 – Subtrair dois números

Digite a opção desejada:

P3.27) Escreva um programa que mostre o menu de opções a seguir, receba dois números e execute uma das operações do menu, de acordo com a escolha do usuário. Se for digitada uma opção inválida, mostrar a mensagem de erro e terminar a execução do programa.

1-Média entre os dois números

2-Diferença do maior pelo menor

3-O produto entre os dois números

Digite a sua opção:

P3.28) Uma empresa decide dar uma gratificação de natal aos seus funcionários, baseada no número de horas extras e no número de horas que o funcionário faltou ao trabalho. O valor da gratificação é obtido pela consulta na tabela a seguir, em que:

H = número de horas extras – $2/3 \cdot$ número de horas-falta

H	Gratificação
> 2.400	R\$ 500,00
$1.800 < H \leq 2.400$	R\$ 400,00
$1.200 < H \leq 1.800$	R\$ 300,00
$600 \leq H \leq 1.200$	R\$ 200,00
$0 < H < 600$	R\$ 100,00
≤ 0	“Sorry, sem gratificação”

Faça um programa que leia o nome do funcionário, o número de horas extras e o número de horas-falta do ano, calcule e escreva, o valor de sua gratificação acompanhado de seu nome (desenvolva o programa na folha em paisagem).

P3.29) Elabore um programa que, dada a idade de um atleta, classifique-o em uma das seguintes categorias:

maternal : menor que 5 anos

infantil: de 5 a 10 anos

juvenil: de 11 a 15 anos

pleno: de 16 a 18 anos

sênior: de 19 a 21 anos

P3.30) Faça um programa que mostre o menu de opções a seguir, receba a opção desejada e execute uma das operações do menu, com a entrada de 3 números distintos, de acordo com a escolha do usuário. Se for digitada uma opção inválida, mostrar uma mensagem de erro e terminar a execução do programa.

1 - Média ponderada dos 3 números com pesos 5, 3 e 2, respectivamente

2 - O maior entre os 3 números

Digite a opção desejada:

P3.31) Escreva um programa que, tendo como dados de entrada os valores x e y , o mesmo retorne o valor de $f(x,y)$, tal que:

$$f(x,y) = \begin{cases} x^2 - y^2 + 2xy & \text{se } x > y \\ 2xy + x + y & \text{se } x = y \\ y^2 + x^2 + 2xy & \text{se } x < y \end{cases}$$

P3.32) Dado o programa abaixo, considerando $b1$, $b2$, $b3$, variáveis e $C1$, $C2$, $C3$, $C4$ e $C5$, comandos, responda a seguir:

```
int main ()    // programa encadeado
{
    bool b1, b2, b3;

    cout << "Entre com os valores lógicos para as 3 variáveis ";
    cin >> b1 >> b2 >> b3;

    if (b1)
        C1;
    else if (b2)
        if (b3)
            C2;
        else
        {
            C3;
            C4;
        }
    C5;
    return 0;
}
```

- Se $b1 = \text{true}$, $b2 = \text{true}$, $b3 = \text{false}$, quais os comandos que serão executados?
- Se $b1 = \text{false}$, $b2 = \text{true}$, $b3 = \text{false}$, quais os comandos que serão executados?
- Quais os valores de $b1$, $b2$, $b3$ para que somente o comando $C5$ seja executado?
- Se $b1 = \text{false}$, $b2 = \text{true}$, $b3 = \text{true}$, quais os comandos que serão executados?

P3.33) Um supermercado deseja reajustar os preços de seus produtos usando o seguinte critério: o produto poderá ter seu preço aumentado ou diminuído. Para ter seu preço alterado, o produto deve preencher **os dois** requisitos a seguir:

REQUISITOS		REAJUSTES	
<i>Venda média mensal</i>	<i>Preço atual</i>	<i>% de aumento</i>	<i>% de diminuição</i>
< 500	< 30.00	10	-
>= 500 e < 1200	>= 30.00 e < 80.00	15	-
>= 1200	>= 80.00	-	20

Faça um programa que leia o preço atual e a venda mensal de um produto, calcule e mostre o novo preço. Caso o produto não atenda aos requisitos, mande uma mensagem ao usuário que não foi possível alterar o preço daquele produto.