

<b>Objetivos</b>	a. Demonstrar o funcionamento de diversas ED sob os aspectos: ocupação de memória, armazenamento e recuperação de conteúdo, operações principais b. Revisar conceitos e construções básicas de programação em Linguagem C c. Apresentar situações em que se pode aplicar uma função ou um procedimento
<b>Referências gerais</b>	SCHILDT, H. <i>C Completo e Total</i> . São Paulo: Pearson, 1997. TENENBAUM, A.M.; LANGSAM, Y.; AUGENSTEIN, M.J. <i>Estruturas de Dados usando C</i> . São Paulo: Pearson Makron Books, 1995.

Objetivo específico desta aula prática: trabalhar com programas simples, funções e procedimentos (linguagem C).

Suporte teórico 1. Construções básicas em linguagem C: estruturas de decisão e de sequência.

Estrutura	Descrição	Exemplo	Obs.
<pre>if ( condição ) {     comandos; }</pre>	Executa os <b>comandos</b> somente se a <b>condição</b> for verificada (isto é, avaliada e produzir como resultado o valor <b>true</b> )	<pre>if ( x == 2 ) {     y = 35;     z = x * y; }</pre>	Note a diferença entre os símbolos de atribuição (=) e de comparação de igualdade (==).
<pre>if ( condição ) {     comandos_1; } else {     comandos_2; }</pre>	Dependendo do resultado ( <b>true</b> ou <b>false</b> ) da avaliação da <b>condição</b> , é executado o grupo <b>comandos_1</b> ou <b>comandos_2</b>	<pre>if ( x &gt; 2 ) {     y = 35 * x; } else {     z = 35 - x; }</pre>	Neste exemplo as chaves { e } são opcionais porque os comandos em cada ramo são únicos.
<pre>for(     inicialização;     condição de     continuidade;     incremento ) {     comandos; }</pre>	Executa os <b>comandos</b> enquanto a <b>condição de continuidade</b> for avaliada como <b>true</b> (o incremento deve ser tal que em algum momento essa avaliação produza <b>false</b> e a repetição seja interrompida)	<pre>x = 1; for ( i = 0; i &lt; 10; i++ ) {     x = x + 2; } printf ( "%d", x );</pre>	A impressão mostrará o valor <b>21</b> (confira!).
<pre>inicialização; while (     condição de     continuidade ) {     comandos;     incremento; }</pre>	Idem acima	<pre>x = 1; i = 0; while ( i &lt; 10 ) {     x = x + 2;     i = i + 1; } printf ( "%d", x );</pre>	Formatos de impressão: %d : inteiros %f : reais %c: caracteres
<pre>do {     comandos;     incremento; } while (     condição de     continuidade );</pre>	Idem acima, mas os <b>comandos</b> são executados ao menos uma vez !	<pre>x = 1; i = 0; do {     x = x + 2;     i++; } while ( i &lt; 10 ); printf ( "%d", x );</pre>	Qual será o valor impresso ?

**Atividade 1 (uso de `if...else`).** Crie um programa em linguagem C no qual você calcula o novo salário de um funcionário. Sabe-se que os funcionários que possuem atualmente salário até R\$ 800,00 terão aumento de 20% ; os demais terão aumento de 10%. Será preciso então :

Passo 1 – receber o salário atual do funcionário

Passo 2 – se o salário atual do funcionário é menor ou igual a R\$ 800,00 , calcular o novo salário com 20% de aumento; caso contrário, calcular o novo salário com 10% de aumento

Passo 3 – mostrar o novo salário

**Atividade 2 (uso de `for`).** Repita a Atividade 1, sabendo agora que são 5 funcionários, os quais têm salários diferentes.

**Atividade 3 (uso de `while`).** Escreva um programa em linguagem C que recebe um número inteiro qualquer entre 1 e 50 e mostra os números ímpares existentes entre esse inteiro e 99.

**Atividade 4 (uso de `do...while`).** Faça um programa que receba uma senha formada de quatro dígitos, verifique se a senha está correta e, caso não esteja, solicite novamente a senha. Se a senha estiver correta, deverá ser apresentada a mensagem “Senha Correta”, caso contrário, “Senha Incorreta”. A senha correta é composta pelos 4 últimos números do seu RA.

**Suporte teórico 2.** Diferença *conceitual\** entre “função” e “procedimento”.

O que	Descrição	Exemplo	Obs.
<b>Função</b> ( <i>function</i> )	Pode receber um ou mais argumentos e retorna um valor	<pre>#include&lt;math.h&gt;  int main() {     double y;     y = sqrt(33.49);     .. }</pre>	O cabeçalho <code>math.h</code> permite ao compilador anexar, ao seu código, a biblioteca matemática fornecida pela linguagem
<b>Procedimento</b> ( <i>procedure</i> )	Pode receber um ou mais argumentos e <u>não</u> retorna nenhum valor	<pre>// cabeçalho do procedimento void myProc( int );  int main() {     int x;     myProc(x);     ... }  // código de myProc void myProc( int m ) {...}</pre>	O cabeçalho (também chamado de <i>protótipo</i> ou <i>assinatura</i> ) permite ao compilador conferir se a chamada ao procedimento está correta do ponto de vista da entrada (argumentos) e saída (retorno)

\* Na prática, os termos “função” e “procedimento” são usados como sinônimos, independentemente de haver retorno de valor ou não

**Atividade 5.** Crie um programa em linguagem C no qual você utiliza uma função que recebe três valores do tipo `int`, correspondentes a “horas” (0 a 23), “minutos” (0 a 59) e “segundos” (0 a 59), calcula o valor equivalente em segundos e retorna esse valor ao programa principal, onde ele é exibido na tela.

**Atividade 6.** Crie um programa em linguagem C no qual você utiliza um procedimento que recebe um valor do tipo `int`, correspondente a “minutos” (0 a 1339), calcula e exibe o correspondente valor em “horas” (0 a 23) e “minutos” (0 a 59).

**Atividade 7 (uso de `switch`).** Crie um programa em linguagem C no qual você escolhe qual das seis atividades acima será executada. Cada uma delas é implementada como função ou procedimento, conforme o caso.