



Roteiro de Aula Prática

ALGORÍTMOS E LÓGICA DE PROGRAMAÇÃO

Disciplina: ALGORÍTMOS E LÓGICA DE PROGRAMAÇÃO

[Clique aqui e veja orientações e exemplos de roteiro de aula prática.](#)

Visual G: <https://mdstrm.com/video/60da391e0cdfb41251e0e108.mp4>

Dec++: <https://mdstrm.com/video/60da397f75218e6469378eee.mp4>

ROTEIRO DE AULA PRÁTICA 1

Unidade: 1

Aula (White Label)/Seção (KLS): 1

SOFTWARE

Software / Acesso on-line

Pago / Não Pago

Infraestrutura:

Computador

Descrição do software:

VisualG

<https://mdstrm.com/video/60da391e0cdfb41251e0e108.mp4>

O VisualG é um software que permite criar, editar, interpretar e que também executa os algoritmos em português (estruturado português) como se fosse um “programa” normal de computador. A ferramenta é de uso gratuito e de domínio público.

ATIVIDADE PRÁTICA 1

Atividade proposta:

Criar dois algoritmos e testar utilizando a ferramenta VisualG.

- 1. Para o primeiro algoritmo, o usuário deverá informar a sua idade e em seguida deverá ser informada uma mensagem se ele for menor ou maior de idade.

- 2. O segundo algoritmo, você deverá criar uma estrutura de repetição com teste no início para calcular a potenciação. Você deverá escrever um algoritmo que resolva a expressão a^b , sendo a e b números inteiros e positivos.

Objetivos:

Entender a ferramenta VisualG.

Elaborar algoritmos e testar utilizando a ferramenta VisualG.

Procedimentos para a realização da atividade:

Nessa aula, você deverá utilizar a ferramenta VisualG para criar e testar os seus pseudocódigos que foram propostos.

- Primeiramente, você deverá fazer download da ferramenta VisualG, acessando o seguinte link: <https://visualg3.com.br/>
- Descompacte a pasta do VisualG.
- Dentro da pasta descompactada, acesse o ícone do aplicativo, o ícone é chamado visualg30, conforme a Figura 1 a seguir.

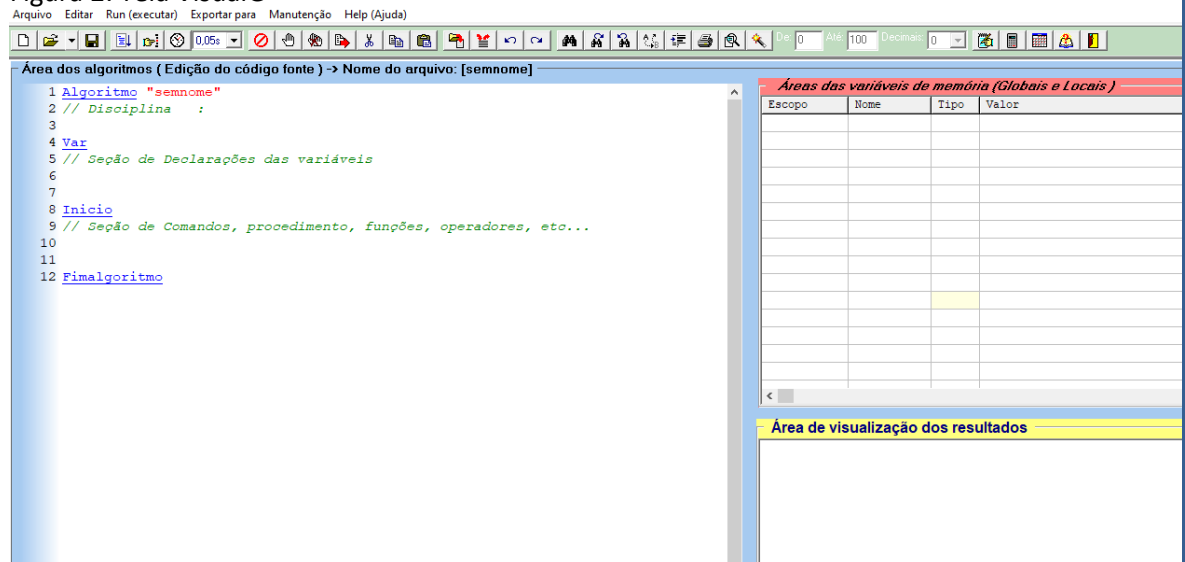
Figura 1. Pasta descompactada VisualG.

Nome	Data de modificação	Tipo	Tamanho
Exemplos	18/05/2020 17:25	Pasta de arquivos	
help	14/11/2018 18:40	Pasta de arquivos	
skins	14/11/2018 18:40	Pasta de arquivos	
dicas	20/06/2008 07:59	Parâmetros de co...	11 KB
help	13/07/2015 18:13	Arquivo de Ajuda ...	691 KB
LEIAME	12/09/2015 00:23	Documento de Te...	1 KB
LEIA-ME	12/09/2015 00:23	Documento de Te...	1 KB
listas	07/04/2019 08:38	Documento de Te...	5 KB
Menu do Visualg autalizado	20/07/2015 02:47	Adobe Acrobat D...	1.236 KB
README	12/09/2015 00:23	Documento de Te...	1 KB
RELAÇÃO DOS COMANDOS DO VISUAL...	04/10/2015 01:59	Documento de Te...	7 KB
TESTE.alg	16/02/2019 16:12	Arquivo ALG	1 KB
VISUALG	26/01/2017 16:51	Parâmetros de co...	1 KB
VISUALG30	13/07/2015 19:13	Arquivo de Ajuda ...	691 KB
visualg30	21/03/2019 22:45	Aplicativo	2.110 KB
VISUALG30	30/07/2020 18:07	Documento de Te...	2 KB

Fonte: Captura de tela Pasta VisualG.

- Abrir o visual G e criar a estrutura do pseudocódigo.

Figura 2: Tela VisualG



Fonte: Captura de tela VisualG.

- Lembre-se que a estrutura no VisualG deve obedecer a seguinte sequência:

Algoritmo "Nome do algoritmo"

Var
// Crie as suas variáveis

Início

// Seção de Comandos, procedimento, funções, operadores, etc...

Fimalgoritmo

Lembrando das estruturas das variáveis:

Variáveis - Tipos

Numéricas: Específicas para armazenamento de números, que posteriormente poderão ser utilizados para cálculos. Ex: idade, salário... Podem ser ainda classificadas como Inteiras ou Reais.

Inteiro: são caracterizados pelos números inteiros, positivos ou negativos.

Real: são caracterizados por números inteiros e decimais (fracionais), sendo positivos ou negativos.

Literal : São variáveis que armazenam letras, números e símbolos especiais. São chamados de caracteres ou, em inglês, string. Ex: (Marcio, Karina38, 2017).

Lógicas: Armazenam somente dados lógicos que podem ser Verdadeiro ou Falso. Ex: O Carlos tem filho? Sim ou Não. A Carol é engenheira? Verdadeiro ou Falso.

Operadores matemáticos:

Operadores	Função	Exemplos
+	Adição	$Y + x$, $7 + 8$
-	Subtração	$10 - 8$, nota - extra
*	Multiplicação	$5 * 5$, nota *4
/	Divisão	$100/5$, valor1/valor2
pot (x,y)	Potenciação	Pot (3,3) = 3 elevado a 3 (3^3)
rad(x)	Radiciação	rad(25) = raiz quadrada de 25 ($\sqrt{25}$)
mod	Resto da divisão	$10 \text{ mod } 3$ resulta em 1
Div	Quociente da divisão	$15 \text{ div } 5$ resulta em 3

Atribuição

Atribuição pode ser utilizada em algoritmos para determinar um valor a uma variável ou até mesmo para uma expressão, e é representada por uma seta (\leftarrow).

Entrada e saída

Podemos descrever os seguintes comandos de entrada e saída:

Comando "escreva" – utilizado para mostrar algo na tela do computador, conhecido como comando de saída.

Comando "leia" – utilizado para armazenar os dados de uma variável, também conhecido como comando de entrada.

1º Algoritmo: para o primeiro algoritmo, você deverá elaborar o algoritmo que o usuário informe a sua idade e caso a idade seja maior que 18 incluir uma mensagem de Maior de idade e caso contrário menor de idade.

2º Algoritmo: para o segundo algoritmo você deverá apresentar o resultado da potenciação, lembrando que a estrutura da potenciação é a seguinte:

$$a^b$$

$$2^2 \Rightarrow 2 \cdot 2$$

$$2^3 \Rightarrow 2 \cdot 2 \cdot 2$$

$$2^4 \Rightarrow 2 \cdot 2 \cdot 2 \cdot 2$$

Você deverá atentar-se para a lógica entre o número de repetições e a variável b (Dica: observe que a variável b é quem irá controlar a quantidade de repetições)

Escreva o algoritmo:

1. Início e declaração de variáveis
2. Obtenção dos valores das variáveis pelo usuário
3. Cálculo do exponencial
4. Impressão do resultado .

Checklist:

- Abrir a interface da ferramenta VisualG
- Criar o pseudocódigo para o primeiro algoritmo que informa a maior ou menor idade.
- Deixar o algoritmo indentado.
- Testar o algoritmo.
- Você deverá entregar o algoritmo e o print do resultado do algoritmo.
- Criar o pseudocódigo do algoritmo para o cálculo da potência.
- Indentar o algoritmo.
- Executar o algoritmo.

Resultado: Aluno, você deverá entregar:

Aluno, você deverá entregar o algoritmo e o print do resultado do algoritmo. algoritmo e o print do resultado do algoritmo.

Referências:

MANZANO, José Augusto N. G. **Algoritmos: lógica para desenvolvimento de programação de computadores**. 29.ed. São Paulo: Érica, 2019

Livro SOUZA, Marco A. **Algoritmos e lógica de programação**. 3.ed. São Paulo: Cengage, 2019

SOARES, Márcio Vieira; GOMES, Marcelo Marques; Souza, Marco Antônio. **Algoritmos e Lógica de Programação**. 2. ed. São Paulo: Cengage Learning, 2011.

ROTEIRO DE AULA PRÁTICA 2

Unidade: 1

Aula (White Label)/Seção (KLS): 3

SOFTWARE

Software / Acesso on-line

Pago / Não Pago

Infraestrutura:

Computador.

Descrição do software:

DevC++

<https://mdstrm.com/video/60da397f75218e6469378eee.mp4>

O DevC++ é um ambiente de desenvolvimento de programas em C/C++ que utiliza o como compilador o GNU g++ ou o GNU gcc. pacote compõe um ambiente completo de desenvolvimento para a criação, debug e compilação do código de programação..

ATIVIDADE PRÁTICA 2

Atividade proposta:

Criar algoritmos para cálculos matemáticos.

Objetivos:

Construção de programas com processamento de dados utilizando os operadores aritméticos, relacionais e lógicos, além de funções matemáticas pré-definidas.

Procedimentos para a realização da atividade:

Segundo Damas (2016), um programa é uma sequência de código organizada de tal forma que permita resolver um determinado problema. Um programa pode ser desenvolvido em módulos distintos e/ou em subprogramas.

Terá que existir um critério ou um formato de escrita bem definido que indique ao compilador todo o conjunto de código escrito pelo programador, onde se define em qual instrução ou local ele começará a executar.

Prestar atenção nas bibliotecas:

- ✓ stdio – essa biblioteca é responsável pelas funções de entradas e saídas, como é o caso da função printf e scanf.
 - Exemplo: #include <stdio.h>
- ✓ stdlib – essa biblioteca transforma as strings (vetores de caracteres) em números.
 - Exemplo: #include <stdlib.h>
- ✓ string – biblioteca responsável pela manipulação de strings.
 - Exemplo: #include <string.h>
- ✓ time – biblioteca utilizada para manipulação de horas e datas.

- Exemplo: #include<time.h>
- ✓ math – biblioteca utilizada para operações matemáticas.
- Exemplo: #include<math.h>...

Operadores:

- ✓ Operadores de atribuição

Para atribuir um valor a uma variável utilizamos o sinal de igual “=”. Exemplo: $y = x + 100$

- ✓ Operadores aritméticos

Os operadores aritméticos em Linguagem C são representados por operadores binários e unários.

- ✓ Operadores Binários

+ soma - subtração * multiplicação / divisão % resto de divisão

Por exponenciação – lembre-se de que para esse operador a biblioteca math.h deve ser inicializada no programa.

- ✓ Operador Unário

Uma das representações unárias é caracterizada pela utilização do resto da divisão, representada pela porcentagem “%”...

- ✓ Operadores de Incremento e Decremento

- Ex: $y = y + 1$; utilizando incremento ficaria desta maneira: $++y$; $y = y - 1$; utilizando o decremento ficaria desta maneira: $--y$;

- ✓ Operadores Relacionais

- $>$ = Maior; $<$ = Menor; $>=$ = Maior ou igual; $<=$ = Menor ou igual; $==$ = Igual; $!=$ = Diferente; Ex : soma = (a $>=$ B);

- ✓ Operadores Lógicos

- $\&\&$ = Operador E; $\|\|$ = Operador OU; $!$ = Operador NÃO;

Funções:

1) Função main()

A função main() é reservada para iniciar um programa em C, sendo a primeira a ser executada.

```
main() { }
```

2) Função printf ()

A função printf () é um comando de saída onde possui um vínculo com a biblioteca stdio.h. É utilizada quando se pretende obter uma resposta na tela do computador. A sua síntese é definida por: printf (“expressão de controle”, listas de argumentos);

3) Função scanf()

A função scanf() é um comando de entrada, ou seja, são informações que possibilitam a entrada de dados pelo teclado, assim, a informação será armazenada em um determinado espaço da memória.

A sintaxe é definida por uma expressão de controle (sempre entre aspas duplas) e pela lista de argumento. A sintaxe da função scanf() é definida por: scanf(“expressão de controle”, lista de argumentos);

- Na primeira etapa o objetivo é criar um programa em C, que calcule a média aritmética das notas de 4 bimestres de um aluno do ensino médio. O programa deverá pedir para que o professor digite as 4 notas, em seguida, deverá ser calculada a média aritmética anual do aluno

- Como segundo desafio você deve criar um programa em C para calcular o valor de um produto com juros. O programa deverá pedir o valor bruto de uma conta e a taxa de juros que será aplicada. O programa deverá fazer o cálculo e dizer qual o valor bruto, o valor líquido e quanto o cliente irá pagar de juros. .

Checklist:

- Abrir a interface da ferramenta DevC++
- Criar um programa em C e incluir as bibliotecas.
- Criar todas as variáveis necessárias.
- Fazer a leitura correta de todos os valores.

- Criar a fórmula para calcular a média.
- Imprimir os resultados.
- Criar um novo programa em C e incluir as bibliotecas.
- Criar todas as variáveis necessárias.
- Fazer a leitura correta de todos os valores.
- Criar a fórmula (ou as fórmulas) para calcular os juros.
- Imprimir os resultados.

Resultados da aula prática: Aluno, você deverá entregar:

- Você deverá entregar o algoritmo e o print do resultado do algoritmo.

Referências:

SZWARCFITER, J. L.; MARKENZON, L. Estruturas de dados e seus algoritmos. 3.ed. Rio de Janeiro: LTC, 2015.

LIVI, Maria Aparecida de Castro. Algoritmos e Programação com exemplos em Pascal e C. Porto Alegre: Bookman, 2014

Backes, André. Linguagem C : completa e descomplicada - 2. ed. - [Reimpr.]. – Rio de Janeiro: LTC, 2022 – 3ª impressão : il. ISBN 978-85-3528-902-2 [Minha Biblioteca]

ROTEIRO DE AULA PRÁTICA 3

Unidade: 2

Aula (White Label)/Seção (KLS): 1

SOFTWARE

Software / Acesso on-line

Pago / Não Pago

Infraestrutura:

Computador.

Descrição do software:

DevC++

<https://mdstrm.com/video/60da397f75218e6469378eee.mp4>

O **DevC++** é um ambiente de desenvolvimento de programas em C/C++ que utiliza o como compilador o GNU g++ ou o GNU gcc. pacote compõe um ambiente completo de desenvolvimento para a criação, debug e compilação do código de programação.

ATIVIDADE PRÁTICA 3

Atividade proposta:

Trabalhar estruturas condicionais.

Objetivos:

Aula prática tem por objeto programar em Linguagem C atividades do cotidiano e treinar o conhecimento em estrutura de decisão condicional - Se/Então, Se/Então/Senão, as estruturas de seleção e finalizar com o encadeamento de estruturas condicionais.

Procedimentos para a realização da atividade:

Condicional Simples/ Condicional Composta:

Segundo Manzano (2013), para a solução de um problema, podemos utilizar a instrução "if", em português "se", onde sua função é tomar uma decisão e criar um desvio dentro do programa, onde, desta forma, podemos chegar a uma condição que pode ser verdadeira ou falsa.

A estrutura condicional composta é bem parecida com a estrutura condicional simples, a diferença é: quando uma condição não é satisfeita, um outro comando será executado. Neste caso, vamos usar o comando "else", que significa "então".

```

if <(condição)>
{
<conjunto de comandos>;
}

```

```

if <(condição)>
{
<conjunto de comandos>;
}
else
{
<conjunto de comandos>;
}

```

Condicional de Seleção de casos

A Estrutura Condicional de seleção de casos “switch-case” é aplicada quando usamos uma variável do tipo inteiro para comparar ou testar determinados valores.

Quando os valores são avaliados, o comando é executado. Em alguns casos é uma ótima opção para substituir a programação de estrutura condicional encadeada.

```

switch (variável)
{
case constante1:
<comandos>
break;
case constante2:
<comandos>
break;
default: <comandos>
}

```

DESAFIO 1: Como primeiro desafio, implemente um programa que calcula um intervalo de tempo, a partir de informações digitadas pelo usuário. Como o cálculo envolve “tempo” é tomado como premissa que as informações de hora e minuto informados sejam valores válidos.

O usuário deve informar um valor inicial para hora e outro para minuto. Em seguida, deve informar um valor final para ambas variáveis. O programa deve fazer a diferença entre os valores, e verificar se o valor final não é maior que o inicial, o que retornará uma mensagem.

Para criar o programa você pode sugerir a sintaxe da estrutura condicional encadeada:

```

if (condição) comando;
else
  if (condição) comando;
  else(condição) comando;
.
.
.
else comando;

```

- Crie o primeiro algoritmo.

DESAFIO 2: Agora você vai criar um novo programa para explorar outra estrutura condicional. Nesse programa o usuário deverá escolher um número de 1 a 7, e o programa imprime o dia da semana correspondente. Por exemplo, se o usuário digitar 1, deverá ser mostrado Domingo!

Nesse tipo de solução, onde se tem mais que três opções a serem testadas, o ideal é usar a estrutura condicional:

```
switch (variável)
{
case constante1: <comandos>
break;
case constante2: <comandos>
break;
default: <comandos>
}

```

Checklist:

Criar um programa na linguagem C.
Criar as variáveis necessárias.
Armazenar os valores iniciais e finais informados pelo usuário.
Realizar o cálculo da diferença entre os tempos informados.
Imprimir o resultado.
Criar um novo programa na linguagem C.
Criar as variáveis necessárias.
Exibir o dia da semana conforme informação numérica inserida pelo usuário.]

Resultados da aula prática: Aluno, você deverá entregar:

Ao final da aula prática você deverá entregar os dois códigos na linguagem C, com as estruturas condicionais propostas.

Referências:

Soffner, Renato Kraide .Algoritmos e Programação em Linguagem C, 1ª edição - São Paulo:Saraiva,2013.[Minha Biblioteca]
DOBRUSHKIN, V. A. Métodos para análise de algoritmos. Rio de Janeiro: LTC, 2012. [Biblioteca Virtual universitária 3.0 Pearson].

ROTEIRO DE AULA PRÁTICA 4

Unidade: 4

Aula (White Label)/Seção (KLS): 1

SOFTWARE

Software / Acesso on-line

Pago / Não Pago

Infraestrutura:

Computador.

Descrição do software:

DevC++

<https://mdstrm.com/video/60da397f75218e6469378eee.mp4>

O **DevC++** é um ambiente de desenvolvimento de programas em C/C++ que utiliza o como compilador o GNU g++ ou o GNU gcc. pacote compõe um ambiente completo de desenvolvimento para a criação, debug e compilação do código de programação.

ATIVIDADE PRÁTICA 4

Atividade proposta:

Criar uma lista(struct) para armazenar dados de alunos de uma sala de aula, ler esses dados e apresentá-los.

Objetivos:

Conhecer e compreender as estruturas heterogêneas, sua construção e uso adequados, e sua aplicação em programas de computador.

Procedimentos para a realização da atividade:

Criar uma estrutura heterogênea (struct) que armazene os dados de uma sala de aula com oito alunos. Para cada aluno deve-se considerar o nome, quatro notas e uma média simples. Deve também ler os dados dos alunos e apresentá-los.

Um recurso muito utilizado em programação é armazenar um conjunto de dados dentro de uma mesma variável, formando assim uma estrutura heterogênea ou, como é conhecido na linguagem C, struct (DEITEL; DEITEL, 2011).

É importante ressaltar que uma struct funciona como um “tipo de dado”, e para que seja possível utilizá-la é necessário atribuir uma (ou mais) variáveis à estrutura criada.

A sintaxe da struct é:

Sintaxe:

struct <identificador>

{

```
<listagem dos tipos e membros>;  
}  
struct <identificador> <variavel>;
```

Passo1 : Definição das variáveis e configurações do ambiente;

Passo 2: Rotina para entrada dos nomes e notas;

Passo 3 : Processamento do cálculo da médias;

Passo 4: Classificação dos dados;

Passo 5: Apresentação da tabela nome/notas/média.]

Checklist:

Criar uma struct para armazenar dados de um aluno em forma de lista;

Inicializar a lista;

Fazer os cálculos;

Fazer a apresentação desses dados.

Resultados da aula prática: Aluno, você deverá entregar:

Ao final da aula prática você deverá entregar um código na linguagem C, que crie uma lista l para armazenar dados de um aluno com a opção de realizar operações sobre essa lista.]

Referências:

MANZANO, José Augusto N. G. **Algoritmos: lógica para desenvolvimento de programação de computadores**. 29.ed. São Paulo: Érica, 2019

Soffner, Renato Kraide .Algoritmos e Programação em Linguagem C, 1ª edição - São Paulo:Saraiva,2013.[Minha Biblioteca]

DOBRUSHKIN, V. A. Métodos para análise de algoritmos. Rio de Janeiro: LTC, 2012. [Biblioteca Virtual universitária 3.0 Pearson]

ORIENTAÇÕES PARA ELABORAÇÃO E ENTREGA DO RELATÓRIO

1. O trabalho será realizado **individualmente**.
2. Para a instalação dos softwares seguir as orientações dos vídeos (link no início do roteiro).
3. **Importante:** Você deverá postar o trabalho finalizado no AVA, o que deverá ser feito na pasta específica, obedecendo ao prazo limite de postagem, conforme disposto no AVA. **Não existe prorrogação para a postagem da atividade.**
4. As atividades a serem entregues deverão estar consolidadas em um único relatório. As entregas de cada atividade está no item **“Resultados da aula prática”.**
5. Esse relatório deverá ser original e, portanto, não poderá haver documentos idênticos aos de outros alunos ou com reprodução de materiais extraídos da internet. Os trabalhos plagiados serão **invalidados**, sendo os alunos reprovados na atividade. Lembre-se de que a prática do plágio constitui crime, com pena prevista em lei ([Lei n.º 9.610](#)), e deve ser evitada no âmbito acadêmico.
6. **Importante:** O relatório deverá ser enviado preferencialmente em formato Word.

Lembre-se de que seu Tutor a Distância está à disposição para lhe atender em suas dúvidas e, também, para repassar orientações sempre que você precisar.