**ATIVIDADE AVALIATIVA – 1º BIMESTRE (2 PONTOS)**

**DATA DA ENTREGA: 03/10/2022**

**Enviar arquivo em PDF para** [**https://drive.google.com/drive/folders/1c16sFy41OntHGgSIg8Kxkm\_NiIMCUVwb?usp=sharing**](https://drive.google.com/drive/folders/1c16sFy41OntHGgSIg8Kxkm_NiIMCUVwb?usp=sharing)

**LP1 - Vitor Murgi Beloto - 22252903 - 26/09/2022**

1. Conceitue:
2. Lógica de Programação

O sentido da programação lógica é trazer o estilo da lógica matemática à programação de computadores. Matemáticos e filósofos encontram na lógica uma ferramenta eficaz para desenvolvimento de teorias. Vários problemas são normalmente expressos como teorias. Dizer que um problema precisa de solução frequentemente equivale a perguntar se uma nova hipótese é consistente com uma teoria existente ou se é consequência dela. A lógica proporciona uma maneira de demonstrar se uma questão é verdadeira ou falsa.

1. Algoritmo

Em matemática e ciência da computação, um algoritmo é uma sequência finita de ações executáveis que visam obter uma solução para um determinado tipo de problema. Segundo Dasgupta, Papadimitriou e Vazirani; "Algoritmos são procedimentos precisos, não ambíguos, padronizados, eficientes e corretos.". As suas características são: finitas, o algoritmo deve eventualmente resolver o problema; bem definidas: os passos devem ser definidos de modo a serem entendidos; efetivas, deve sempre resolver o que tem para solucionar, antecipando falhas.

1. Linguagem de Programação

A linguagem de programação é um método padronizado por um conjunto de regras sintáticas e semânticas, de implementação de um código fonte que pode ser compilado e transformado em um programa de computador, ou utilizado com script interpretado, que formará instruções de processamento ao computador.

1. Compilação

Compilação é o processo de tradução do programa escrito em uma linguagem de programação para um formato no qual o computador entenda. A compilação gera um ficheiro executável a partir do código fonte.

1. Sintaxe

A sintaxe de programação é o conjunto de normas que regulam e coordenam as diferentes variáveis e sua associação. Na linguagem, a sintaxe é o conjunto de normas e leis combinatórias que estruturam a construção de orações e textos.

1. Variável

Em programação, uma variável é um objeto capaz de reter e representar um valor ou uma expressão. Enquanto as variáveis só existem em tempo de execução, elas são associadas a nomes, chamados identificadores, durante o tempo de desenvolvimento.

1. Biblioteca

Na programação, trata-se de uma coleção de dados e recursos usados por um programa de computador, trazendo assim, um conjunto de funções que já foram escritas por um desenvolvedor.

1. Estrutura Condicional

Permite a escolha do grupo de ações e estruturas que serão executadas, quando determinadas condições, representadas por expressões lógicas forem ou não satisfeitas.

1. Explica a funcionalidade dos seguintes comandos/funções:
2. scanf()

Função responsável por ler um dado do teclado e imediatamente atribuí-lo a uma variável.

1. printf()

Sendo essa, a função responsável por imprimir algo na tela, pode ser um texto, ou o conteúdo de uma variável ou constante.

1. getch()

Função que retorna a um carácter assim que ele for digitado, ou seja não é aguardado até que seja digitado o retorno.

1. pow()

Função utilizada para obter a potência de um número, porém requer a inclusão da biblioteca “<math.h> para ser utilizada.

1. sqrt()

Função utilizada para obter a raíz quadrada de um número, porém requer a inclusão da biblioteca “<math.h> para ser utilizada.

1. fflush(stdin)

Função essa que é normalmente usada para apenas fluxo de saída. Sua finalidade é limpar o buffer de saída e mover dados do buffer para o console ou disco.

1. Sabe-se que uma variável corresponde a uma posição de memória, cujo conteúdo pode variar ao longo do tempo durante a execução de um programa. Porém, ela só pode armazenar um valor a cada instante. Por que é necessário informar o tipo de dado a ser armazenado na declaração de variáveis?

Pois cada variável dada no programa pode assumir uma função especifica para o desenvolvimento e organização do programa estruturado, por conta do seu tamanho em bytes, que diferencia cada uma, por isso só armazenam um valor a cada instante, apesar de poderem assumir diferentes valores.

1. Explique de forma sucinta os passos (estruturas) necessários para o desenvolvimento de um algoritmo (Linguagem C)

Primeiramente, devem ser informadas as bibliotecas (Coleção de dados já escritas por outros desenvolvedores) que serão utilizadas no algoritmo em desenvolvimento.

1. Em uma loja de sapatos, Ronaldo recebe uma remuneração fixa e uma comissão. Considerando que Ronaldo recebe 10% de comissão sobre o valor total de suas vendas e um salário fixo, faça um algoritmo que receba o valor total das vendas, o salário fixo de Ronaldo e calcule quanto ele receberá de salário por mês.

**Devem ser multiplicados os valores da venda de Ronaldo por 0.1, em seguida adicionar o valor ao fixo de Ronaldo.**

1. Fale sobre a funcionalidade e a importância dos seguintes operadores:
2. %

Função módulo

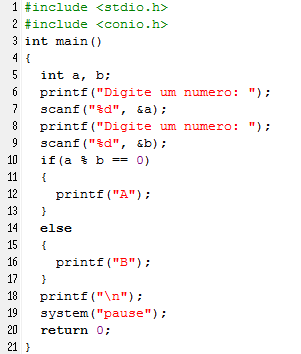
1. &&

& é um operador que nos permite acessar e modificar um endereço de memória de uma variável.

1. ||

Para expressões que possuem a condição OU (||), se uma das condições for verdadeira, toda a sentença será. E para a expressão ser falsa, todas as condições devem ser falsas.

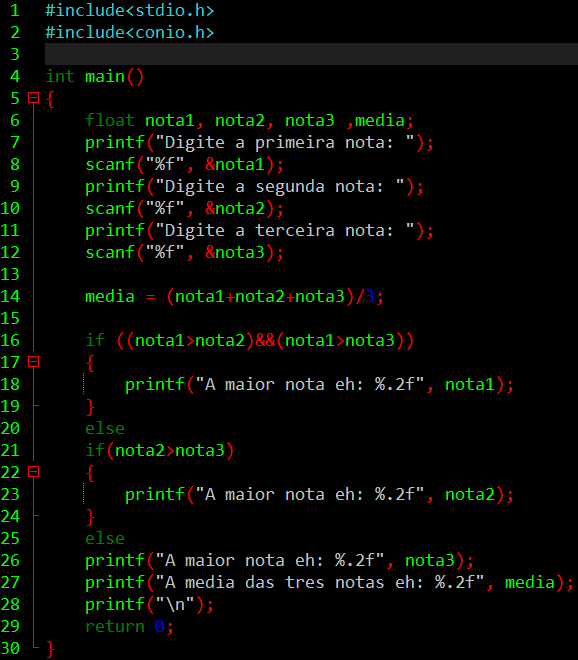
1. Considere o seguinte programa escrito em Linguagem C.



Qual o valor apresentado na saída para as entradas: a = 130 e b = 13.

**A saída apresenta o valor “A”**

1. Escreva um **programa em C** que receba três notas, calcule e mostre a média e a nota mais alta.



1. Descubra e corrija os erros de **sintaxe** **e semântica** do programa abaixo:

#include<**stdio.h**>

**#include<stdlib.h>**

**int** main()

**{**

int n1,n2,n3;

printf("Digite o 1 numero: ");

scanf("%d",&n1);

printf("Digite o 2 numero: ");

scanf("%d",**&**n 2);

printf("Digite o 3 numero: ");

scanf("**%f**",&n 3);

if ( (n1<n2) && (n1<n3)**)**

{

printf("\n O menor numero e: %d", n1);

}

else

{ if (n2<n3)

{

printf("\n O menor numero e: "**%d** ,n2);

}

else

{

printf("\n O menor numero e: %d",n3)**;**

}

}

getch**ar**();

}

1. Considere o seguinte programa (em C):

**#include <stdio.h>**

**main()**

**{**

**int** A, B, AUX;

**float** SOMA;

**char** LETRA;

**printf**( “Digite um numero: “);

**scanf**(“%d”, &A);

B = A \* 2;

SOMA = A + B;

A = 10;

**if** ( (SOMA >= 15 ) &&( A < B ) )

{

AUX=A;

LETRA = ‘M’;

}

**else**

{

AUX=B;

LETRA = ‘F’;

}

**printf** ( “O valor de Aux e: %d “, AUX);

**getch**();

**}**

Considerando que o valor lido para a variável **A** foi **5**, Quais os valores armazenados nas variáveis **AUX** e **LETRA**.

**AUX = 10**

**LETRA = F**