|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Natureza da Avaliação | Valores | | Disciplina: Algoritmos e Programação II | Curso | Identificação da Turma | | |
| Valor | Nota | Sistemas de informação | Turno | Período | Turma |
| Atividade |  |  | Prof.ª Maíza Cristina de Souza Dias | NOT | 3º e 4º | ÚNICA |
| NOME: | | | | | DATA: 11/08/2021 | | |

**REVISÃO / DIAGNÓSTICO**

**LISTA 01**

1. **Estrutura Sequencial**

Faça um algoritmo que receba 4 números inteiros, calcule e mostre a soma destes números.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Grau de Dificuldade: | Muito Fácil | Fácil | Médio | Difícil | Muito Difícil |
|  | x |  |  |  |

#include <stdio.h>

int main()

{

int a, b, c, d, soma;

printf("Digite o primeiro numero: ");

scanf("%d", &a);

printf("Digite o segundo numero: ");

scanf("%d", &b);

printf("Digite o terceiro numero: ");

scanf("%d", &c);

printf("Digite o quarto numero: ");

scanf("%d", &d);

//soma= a+b+c+d;

printf("Resultado da soma dos quatro numeros: %d ", a+b+c+d);

return 0;

}

1. **Estrutura Sequencial**

Faça um algoritmo que receba o peso de uma pessoa em quilos, calcule e mostre o peso em gramas.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Grau de Dificuldade: | Muito Fácil | Fácil | Médio | Difícil | Muito Difícil |
|  |  |  |  |  |

#include <stdio.h>

int main()

{

float quilos, gramas;

printf("Digite peso em quilos: ");

scanf("%f", &quilos);

gramas = quilos \*1000;

printf("O peso da pessoa em gramas é: %.0f ", gramas);

return 0;

}

1. **Estrutura Sequencial**

Faça um algoritmo que receba um número positivo e maior que zero, calcule e mostre:

* 1. O número digitado ao quadrado;
  2. o número digitado ao cubo;
  3. a raiz quadrada do número digitado.
  4. Calcule o logaritmo natural do número digitado.

*Dica: use a biblioteca <math.h> - slide 91*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Grau de Dificuldade: | Muito Fácil | Fácil | Médio | Difícil | Muito Difícil |
|  |  |  |  |  |

#include <stdio.h>

#include <math.h> //biblioteca que nos ajuda a realizar algumas funÃ§Ãµes matemÃ¡ticas

int main()

{

int numero, quadrado, cubo, raiz, loga;

printf("Digite um numero inteiro maior que zero: ");

scanf("%i", &numero);

quadrado= pow(numero,2); //pow(x,y), onde x Ã© o numero que vc quer calcular, y Ã© potencia

cubo = pow(numero,3); //pow(x,y), onde x Ã© o numero que vc quer calcular, y Ã© potencia

raiz = sqrt(numero);

loga = log(numero);

printf("\nO quadrado do numero digitado é: %i", quadrado);

printf("\nO numero digitado ao cubo é: %i ", cubo);

printf("\nA raiz quadrada do numero digitado é: %i ", raiz);

printf("\nO logaritmo natural do numero digitado: %i ",loga);

return 0;

}

*Tabela

Descrição gerada automaticamente*

1. **Estrutura Condicional**

Faça um algoritmo que leia um número inteiro e escreva na tela “*O número digitado é negativo*”, caso o número seja menor que zero.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Grau de Dificuldade: | Muito Fácil | Fácil | Médio | Difícil | Muito Difícil |
|  |  |  |  |  |

#include <stdio.h>

int main()

{

int numero, x;

printf("Digite um numero: ");

scanf("%d", &numero);

if (numero < 0)

printf("\nO número digitado é negativo");

return 0;

}

1. **Estrutura Condicional Composta**

Faça um algoritmo que determine (imprima) se um dado número N inteiro (recebido através do teclado) é PAR ou ÍMPAR.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Grau de Dificuldade: | Muito Fácil | Fácil | Médio | Difícil | Muito Difícil |
|  |  |  |  |  |

#include <stdio.h>

int main()

{

int numero, x;

printf("Digite um numero inteiro maior que zero: ");

scanf("%d", &numero);

x= (numero % 2);

if (numero == 0)

printf("\nO número %d é par", numero);

else

printf("\nO número %d é ímpar", numero);

return 0;

}

**Aula 12/08/2021**

1. **Estrutura Condicional Composta**

Dado três valores, A, B e C, construa um programa para verificar se estes valores podem ser valores dos lados de um triângulo. Regra: (A+B > C) e (A+C > B) e (B+C > A).

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Grau de Dificuldade: | Muito Fácil | Fácil | Médio | Difícil | Muito Difícil |
|  |  |  |  |  |
|  |  |  |  |  |  |

Tabela

Descrição gerada automaticamente

1. **Estrutura Condicional Composta**

Dado três valores, A, B e C, construa um programa para verificar se estes valores podem ser valores dos lados de um triângulo, **e se for, se é um triangulo escaleno, um triangulo equilátero ou um triangulo isósceles.**

***Dica:***

*Equilátero (três lados iguais)*

*Isósceles (dois lados iguais)*

*Escaleno (não possui lados iguais)*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Grau de Dificuldade: | Muito Fácil | Fácil | Médio | Difícil | Muito Difícil |
|  |  |  |  |  |
|  |  |  |  |  |  |

1. **Switch**

Construa um algoritmo utilizando o comando switch para escrever o nome do dígito lido de zero a cinco. Caso o número seja menor que zero ou maior que cinco o algoritmo deverá escrever a mensagem “O número digitado é menor que zero ou maior que cinco.”.

0 -> “zero”

1 -> “um”

etc.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Grau de Dificuldade: | Muito Fácil | Fácil | Médio | Difícil | Muito Difícil |
|  |  |  |  |  |
|  |  |  |  |  |  |

1. **Switch**

Construir o switch um algoritmo para ler dois números inteiros e realizar umas das cinco operações de acordo com a opção escolhida:

+ adição

- subtração

\* multiplicação

/ divisão

% resto da divisão

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Grau de Dificuldade: | Muito Fácil | Fácil | Médio | Difícil | Muito Difícil |
|  |  |  |  |  |

1. **Se Composto**

Construir um programa que leia dois números e efetue a adição. Caso o valor somado seja maior que 20, este deverá ser apresentado somando-se a ele mais 8; caso o valor somado seja menor ou igual a 20, este deverá ser apresentado subtraindo-se 5.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Grau de Dificuldade: | Muito Fácil | Fácil | Médio | Difícil | Muito Difícil |
|  |  |  |  |  |

1. **Estruturas de Repetição**

Imprimir na tela do número 1 até o 100.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Grau de Dificuldade: | Muito Fácil | Fácil | Médio | Difícil | Muito Difícil |
|  |  |  |  |  |

1. **Estruturas de Repetição**

Imprimir na tela do número 100 até o 1.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Grau de Dificuldade: | Muito Fácil | Fácil | Médio | Difícil | Muito Difícil |
|  |  |  |  |  |

1. **Estruturas de Repetição**

Apresentar o total da soma obtida dos cem primeiros números inteiros.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Grau de Dificuldade: | Muito Fácil | Fácil | Médio | Difícil | Muito Difícil |
|  |  |  |  |  |

1. **Estruturas de Repetição**

Fazer um programa que receba um valor n no teclado e determine o menor. A condição de término do programa é quando o usuário digitar zero.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Grau de Dificuldade: | Muito Fácil | Fácil | Médio | Difícil | Muito Difícil |
|  |  |  |  |  |

1. **Estruturas de Repetição**

Solicitar um número entre 1 e 4. Se a pessoa digitar um número diferente, mostrar a mensagem "entrada inválida" e solicitar o número novamente. Se digitar correto mostrar o número digitado. O algoritmo será finalizado quando a pessoa digitar o número zero.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Grau de Dificuldade: | Muito Fácil | Fácil | Médio | Difícil | Muito Difícil |
|  |  |  |  |  |

1. **Estruturas de Repetição**

A série de Fibonacci é formada pela sequência:

**1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144...**

Escreva um programa que gere a série de FIBONACCI até o N-ésimo termo.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Grau de Dificuldade: | Muito Fácil | Fácil | Médio | Difícil | Muito Difícil |
|  |  |  |  |  |

1. **Estruturas de Repetição**

Escreva um programa que receba quinze números do usuário e imprima a raiz quadrada de cada número.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Grau de Dificuldade: | Muito Fácil | Fácil | Médio | Difícil | Muito Difícil |
|  |  |  |  |  |

1. **Estruturas de Repetição**

Escreva um programa que leia 200 números inteiros e imprima quantos são pares e quantos são ímpares.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Grau de Dificuldade: | Muito Fácil | Fácil | Médio | Difícil | Muito Difícil |
|  |  |  |  |  |

1. **Estruturas de Repetição**

Escreva um programa que leia **N** números inteiros e imprima quantos são pares e quantos são ímpares.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Grau de Dificuldade: | Muito Fácil | Fácil | Médio | Difícil | Muito Difícil |