

DICAS PROVA DE ALGORITIMO

#include <stdio.h>//aqui incluimos uma biblioteca, que e nada mais um local onde ja possui funções prontas como printf() e scanf()

#include <math.h> //biblioteca matemática

#define CONSTANCE 2; //definir constantes - boa prática colocar constante com letra maiúscula.

Sitaxe: <nome> <valor>

//todo código C necessita de uma função main (){}
int main (){

//declaração de variáveis

//regras para identificadores: não pode conter espaço, caracteres especiais e nem começar com números

/* palavras com a primeira letra maiúscula

/boa prática: variáveis são escritas em letras minúsculas ou upperCamelCase, ou seja, as próximas int a ;

//int e para números inteiros, a letra "a" após o int e o nome da variável, ou seja, seu identificador

float b,c,f;//float são pra números reais , ou seja numero casas decimais. A virgula entre os identificadores "b" e "c", faz com que se crie duas variáveis com nomes diferente, porem mesmo tipo

double d;//double e uma versão do float mais estendida , ocupando mais espaço na memoria e por isso, gera maior precisão numérica

char e ;//char e do tipo caractere (só recebe uma única letra), e o ";" encerra a linha de código

char nome[10],lugar[50]; //se forem mais letras, como um nome, é preciso definir a quantidade máxima de caracteres

printf("Uma ótima prova a todos :");//função apresenta na tela o texto ou caractere que for colocado dentro das aspas duplas

scanf("%d",&a);//função que armazena o valor digitado na variável indicada

//Obs : e preciso colocar no scanf & pois ele indica o endereço de memoria que o valor digitado será armazenado

printf("Me forneça o double :");

scanf("%lf",&d);*/

printf("Me forneça a outros floars :");

scanf("%f%f",&c,&f);//aqui o scanf() pega duas informações diferentes e aloca ela em espaços de memoria diferente e determinados

//%d e uma máscara para números inteiros

//%s para strings

//%c para char

//%f para float

//%lf para double

```
printf ("Me forneça a string :");
```

```
fflush(stdin); //limpa o lixo de memória ,liberando espaço pra string
```

```
scanf("%s",&nome);
```

```
printf("Me forneça o lugar : ");
```

```
fgets(lugar,50,stdin); limpa e ja adquire o text (neste caso e permitido o uso do espaco)
```

```
if ( a == 1) //area destina a condicao
```

```
{
```

```
    //area para o codigo caso a condição de verdadeiro
```

```
    printf ("Vc digitou %d",a); //aqui não colocamos o & pq não estamos inserindo valor apenas pegando
```

```
} Else /*faz com que caso a condição de Cima nao de verdadeiro ele rode esse outro código*/ {printf("E ne ");}
```

```
if (a == 2){
```

```
    if ( a < b )
```

```
{
```

```
    printf("%d",a+b); //o printf pode também realizar soma dentro dele
```

```
    printf("O real e maior que o inteiro ");
```

```
}
```

```
}
```

/*% pode ser usado em cálculos e o resultado desse calco e o resto de uma divisão

Operadores:

-Aritmético:

+ => SOMA

- => SUBTRAÇÃO

/ => RESULTADO COM NUMEROS REAIS (EX: 11/2

=5.5)

\ => RESULTADO COM NUMEROS INTEIROS (11\2 = 5)

% => MODULO (RESTO DA DIVISAO)

-RELACIONAIS:(todo operador relacional retorna um resultado logico)

< MENOR

<= MENOR OU IGUAL

> MAIOR

>= MAIOR OU IGUAL

!= DIFERENTE

== IGUAL

-Lógicos:

E - Conjunção --> verdadeiro se TODAS as condições forem verdadeiras

Ou - disjunção --> verdadeiro se UMA das condições for verdadeira

! Ou NAO negação

-Caractere:

+ concatenação

Obs:

& = e comercial = código de endereçamento (apenas entrada de dados)

SWITCH CASE

É uma forma de reduzir a complexidade de vários if ... Else encadeados.

É muito utilizado, principalmente para uso em estruturas de menu.

O conteúdo de uma variável é comparado com um valor constante, e caso a comparação seja verdadeira, um determinado comando é executado.

Pseudocódigo:

Em português estruturado o comando escolha...caso equivale ao switch...case

Veja estrutura escolha ... caso

Escolha (Variável)

Início

Caso (Valor1):

Instruções;

Caso (Valor2):

Instruções;

Caso (ValorN):

Instruções;

Fim;

Sintaxe do comando switch case em linguagem C

switch (variável)

{

case constante1:

Instruções;

break;

case constante2:

Instruções.

break.

default

Instruções;