# Anotações de C

**Todo código termina com ;**

**Inserindo comentários**, bom para colocar nos códigos para ficar em alerta, dicas ou relatório

para ajudar outros programadores

Bom para colocar uma lógica para voce mesmo lembrar

para comentar uma linha basta colocar //

mas se quiser comentar um conjunto de linhas basta /\*isso aqui é um grande bloco de comentário\*/

**Printf(“”)**

ele serve para mostrar algo na tela, basta escrever printf(“Imprimindo esta menssagem”);

qualquer caractere colocado entre o par de aspas duplas será interpretado como texto e, consequentemente, impresso na tela.Qualquer coisa é considerado um caractere

No printf() quando você coloca %d serve para INTEIROS então voce pode colocar:

printf(“%d”, printf(“\nOi\n”));

ele vai retornar um inteiro que seria a quantidade de caracteres

4.

Dentro das aspas duplas voce pode saltar uma linha na hora da impressão, basta colocar \n aonde irá saltar uma linha.Pode colocar quantos \n quiser ele irá saltar

o system(“pause”) não é recomendado,

melhor utilizar printf(“\nPressione qualquer tecla para finalizar.”);

getchar();

**Variáveis**

Criar uma variável significa separar uma região de memória para guardar um valor.Se quisermos armazenar qualquer coisa precisamos de um “pedaço” da memória para guardar-lo.

Para criar uma variavel ela não pode começar com Numero e caracteres especiais

podemos criar variaveis como

int valor1; // uma variável para um número inteiro chamada valor1

float valor2; // uma variável para um número real chamada valor2

char valor3; // uma variável para uma letra chamada valor3

depois de criar uma variável voce atribui valores para ela.

**scanf()**

O scanf() é utilizado para ler qualquer tipo de dado que o usuário irá escrever

O %d entre as aspas duplas indica para a função scanf() que o dado a ser lido é um número inteiro. O & (e comercial) antes do nome da variável serve para obter o endereço de memória da variável que receberá o dado lido

printf("Digite sua idade: ");

scanf("%d", &valor1);

%d - inteiro

%f - float

%c char

no float (numeros reais, com virgula) por padrão ele irá mostrar sempre 6 casas dps da vírgula.

mas se quiser alterer isso é so colocar .e o numero de quantidades de casa que seja impresso ex

%.2f irá aparecer so 2 casas dps da vírgula

na programação é ponto( .) em vez de virgula ( , )

no char (caractere) para o computador entender o caractere tem que colocar com aspas simples ‘ x ’ quan do voce mesmo define ex:

char sexo = ‘M’;

getchar()

a função getchar é uma segunda opção para o tipo char, aonde ela vai pegar apenas o caractere digitado

char letra;

printf("Digite uma letra: ");

letra = getchar();

printf("Caracter lido: %c\n", letra);

função getc ()

ela serve TAMBÉM para pegar um caractere

ele tambem serve para fazer a leitura de outras fontes como por exemplo txt

voce precisa especificar dentro do parenteses.

char letra;

printf("Digite um caracter: ");

letra = getc(stdin);

printf("Caracter lido: %c\n", letra);

aqui esta sendo para ler um caracter do teclado

fgetc()

Ela é bem semelhante a função getc(). ela faz parte das funções especificas para leitura de dados em ARQUIVOS, por isso inicia com f de file

ela retorna o caracter lido em um arquivo

Na leitura de dados voce pode pedir para o usuario ler varios valores e ele le.

basta apenas colocar no scanf os %d de acordo com a quantidade quer voce quer quer leia

existe um error que pode acontecer se tiver varios juntos, aonde pede um caractere, o enter é considerado um caracter então recomenda-se colocar um espaço no scanf antes do caracter pois ele não considerar

ou uma alternativa facil basta colocar linha por linha o que vai ser digitado

o espaço entre o %c e um comando para o cpmputador desconsiderar o ENTER, o TAB e o ESPAÇO

ex

scanf(“%c %c, &a, &b);

Unidades de Medidas na Computação

Bit

A palavra bit é a menor unidade de armazenamento, suportando um unico dígito binário zero ou um.

8 bits = 1 byte

1024 bytes = 1 kilobyte

1024 kilobyte = 1 megabyte

1024 megabyte = 1 Gigabyte

1024 Gigabyte = 1 Terabyte

1024 Terabyte = 1 Petabyte

1024 Petabyte = 1 Exabyte

1024 Exabyte = 1 Zettabyte

1024 Zettabyte = 1 Yottabyte

Sizeof

O sizeof serve para saber o tamanho que ocupa na memoria

pode ser escrito de dois modos

float x = 1.0

printf("Tamanho de um float na memoria: %d bytes\n", sizeof x); pra saber o tamanho de um variável

printf("Tamanho em memoria de um int: %d bytes \n", sizeof(int));

o sizeof é um operador e não uma função

short

o operador short serve para “economizar memória”

quando você faz uma variável basta coloca short int = x;

um inteiro possui seu limite na memória

intervalo: -2.147.483.648 a 2.147.483.648

para armazenar um valor maior que este basta utilizar o operador long

o tamanho de um float em memória

e igual ao o int guarda as 4 memorias

o short não funciona em um float

para ter um número muito pequeno basta colocar um double

ele tem mais precisão de casas depois da vírgula.

é possivel utilizar o long double

e para ver na tela utiliza-se Lf%(funciona apenas no linux ou mac) para ver no windows basta utilizar \_\_mingw\_printf(“Valor de y: %Lf\n”,y)

Para imprimir caracteres de ESCAPE ou seja que é utilizado pela linguagem ex

quebra de linha -> \n

tabulação -> \t

aspas duplas \”

aspas simples \’

contrabarra \\

Operações matemáticos para:

soma +

subtração -

multiplicação \*

divisão /

% resto da divisão

tambem tem a operação de incremento que seria ++ ou - - que seria somar um ou diminuir 1

precisa colocar ela a esquerda para fazer antes da variavel e direita se for depois

na divisão basta colocar um numero em numero real

para saber o resto da divisão ele ira dividir e pegar apenas o resto

Voce pode fazer uma conversão de uma variavel basta colocar um parenteses a esquerda dela. ex:

printf(“Divisão: %f”, a/ (float)b);

o operador unsigned int ele não armazena numeros com SINAIS ent quando utiliza ele vai aumentar o limite de armazenamento

para utilizar ele voce colocar

unsigned int x = 21142142214

e no print precisa colocar %u

if na linguagem C é:

if(a < 0){

printf(“Valor negativo!\n);

}

Outra forma **Operador ternário**

O operador ternário é um ?

que é a:

condição ? verdadeiro : falso;

ex:

a < 0 ? (printf(“\n\tValor negativo!\n”) : printf(“\nValor positivo!”);

**switch case**

este caso é otimo para fazer escolhas, voce precisa fazer :

int opcao;

printf("1 - Cadastrar produto\n2 - Vender produto\n3 - Buscar produto\n4 - Imprimir relatório\n5 - Sair");

scanf("%d", &opcao);

switch(opcao){

case 1:

// seu código para o caso 1

break;

case 2:

// seu código para o caso 2

break;

case 3:

// seu código para o caso 3

break;

case 4:

// seu código para o caso 4

break;

case 5:

// seu código para o caso 5

break;

default:

// para informar uma opção inválida

}

Em todos os casos você precisa colocar um break menos no default, aonde seria um caso onde o usuário não digitou uma das opções.

os cases também pode ser caractere por exemplo ‘a’ é diferente de ‘A’.

ent no case, você precisa colocar aspas simples

case ‘a’:

case ‘b’:

case ‘c’: