

Introdução

Programação de Computadores

É um conjunto de instruções (comandos), que descrevem uma tarefa a ser realizada por um computador.

Esse conjunto de instruções é chamado de código fonte e é escrito em uma linguagem apropriada, também conhecida com linguagem de programação.

É a codificação de um algoritmo numa linguagem de programação.

Linguagem de Programação

Método padronizado para expressar instruções para o computador;

Conjunto de Regras que permite que um programador (pessoa que desenvolve programas/software), especifique precisamente quais informações um computador vai atuar.

Linguagem C

A linguagem C foi criada na década de 70, por Dennis Ritchie, no centro de pesquisa AT&T Bell Labs.

Derivada de outra linguagem chamada B, foi desenvolvida para reescrever o sistema operacional UNIX.

Hoje há versões da linguagem C para os mais variados sistemas operacionais.

Características da linguagem C

- Portabilidade entre máquinas e sistemas operacionais;
- Modularidade: Desenvolvimento através de rotinas interligadas;
- Recursos de baixo nível: controle direto do hardware;
- Simplicidade;
- Facilidade de uso;
- Pode ser utilizada para os mais variados propósitos;

Estrutura básica de um programa em C

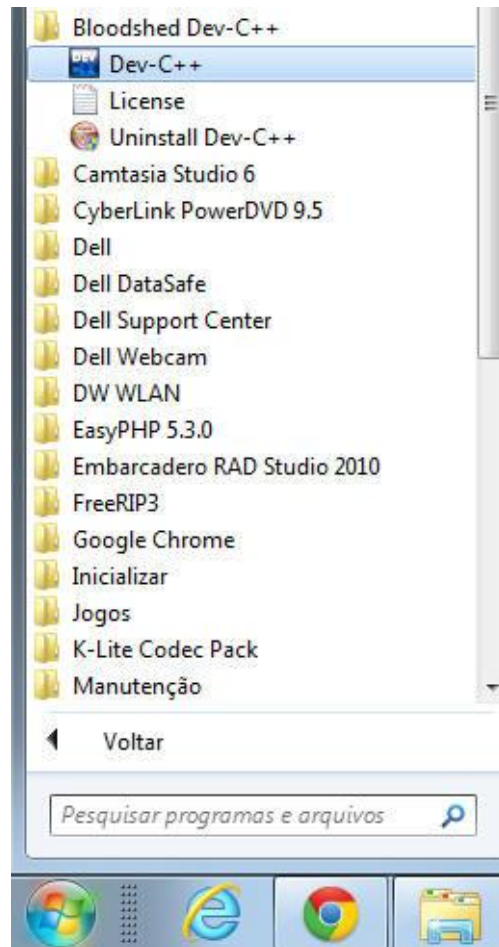
- Os programas começam com a função main()
- O corpo do programa fica entre chaves {}
main()

```
{  
  
}
```

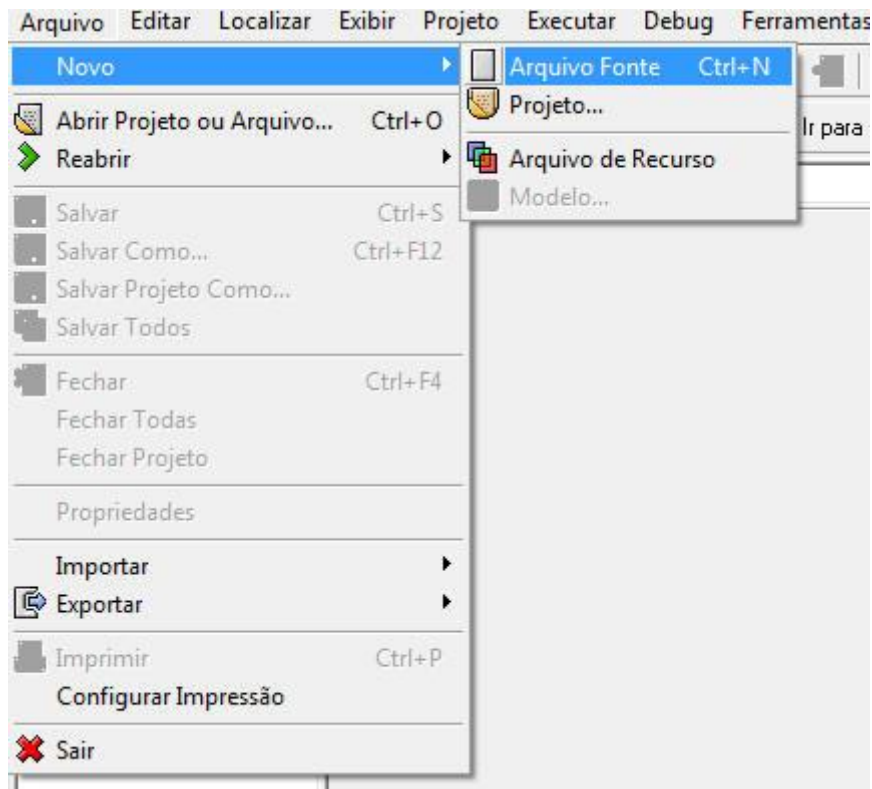
Utilizando a IDE DevC++ para desenvolver os programas.

Para executar os programas, iremos utilizar a IDE DevC++. Essa IDE é de simples utilização e é uma das mais utilizadas para o estudo da linguagem C:

1) Para executar a IDE, clique em todos os programas -> Informática ->BloodShed Dev-C++ -> Dev-C++

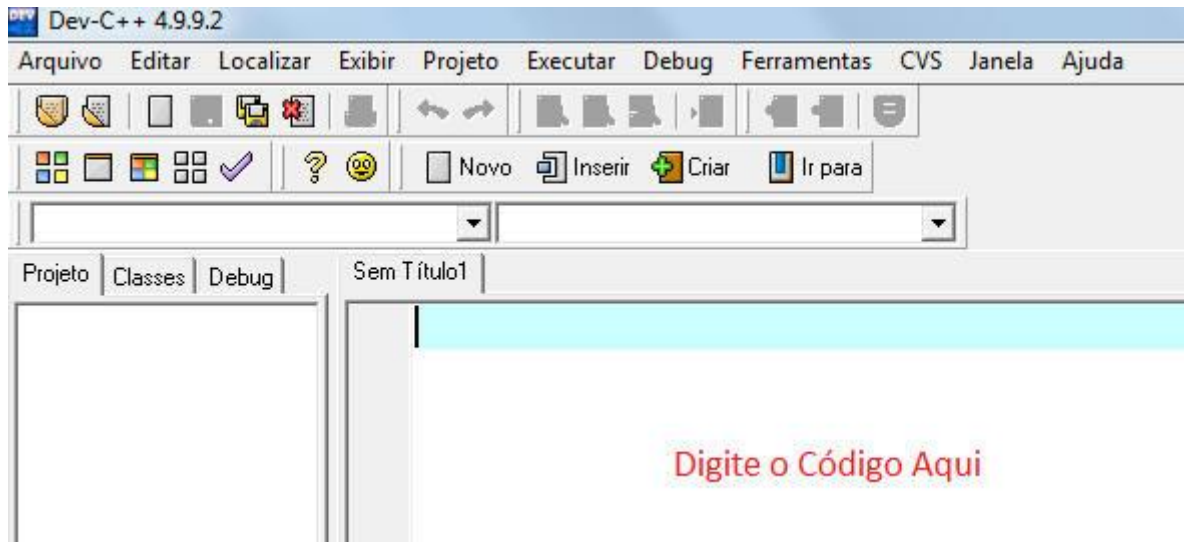


2) Para criar um novo programa, clique em Arquivo -> Novo -> Arquivo Fonte

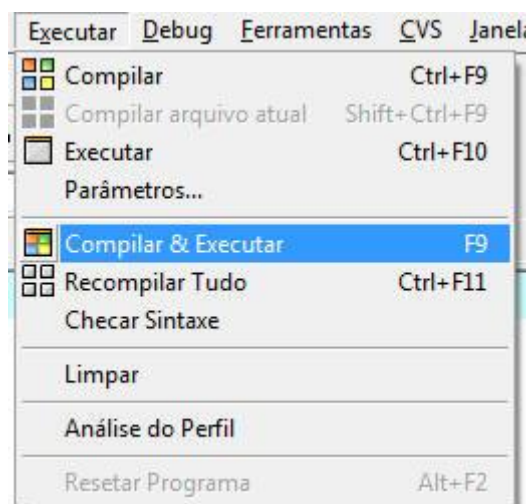


Linguagem C

Ausanello



3) Para executar o programa, clique em Executar -> Compilar & Executar ou pressione F9



Comando de Saída de Dados

Função `main()` e Função `printf()`

Função `main()`: “A função `main()` deve existir em algum lugar de seu programa e marca o ponto de início da execução do programa. Se o programa for constituído de uma única função esta será a `main()`”

```
main()
{
    Comandos que serão executados;
}
```

Função printf()

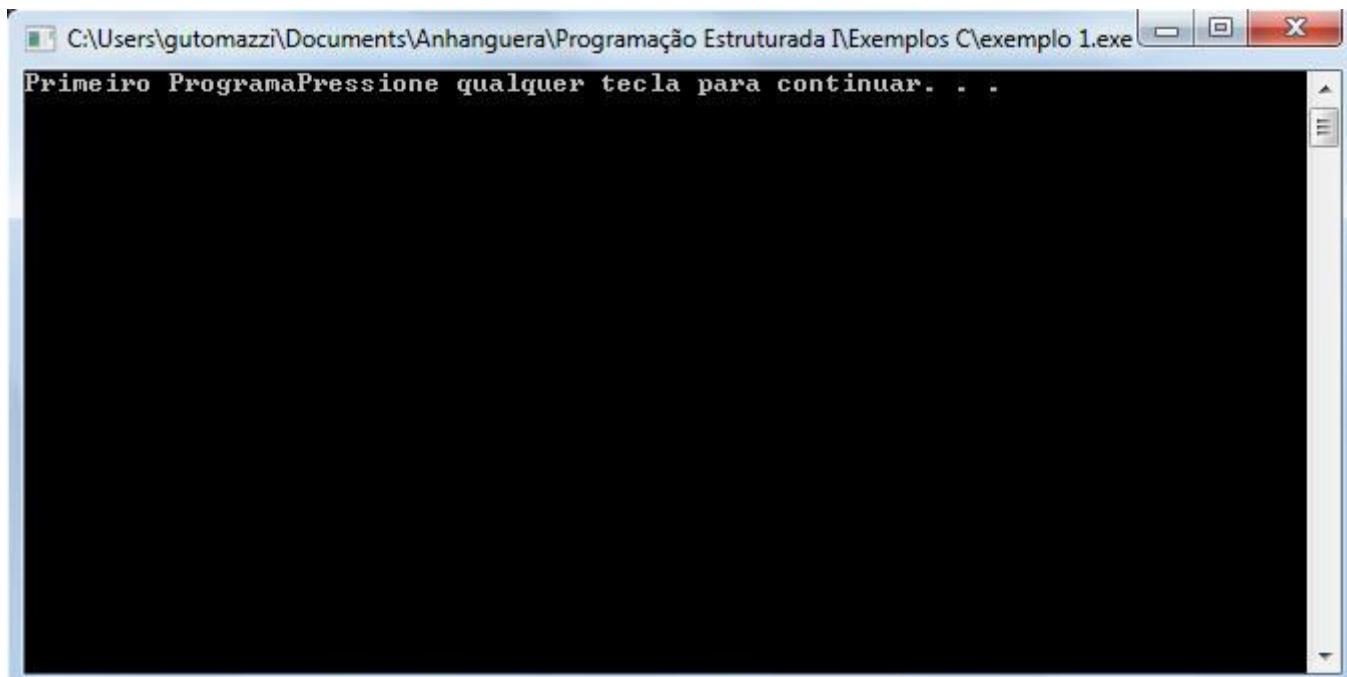
“Uma das funções de E/S (entrada e saída) que podem ser usadas em C. Quando o programa encontra essa linha, passa o controle para a função printf() que imprime na tela do seu computador.”

Exemplo1:

```
#include<stdio.h>
#include <stdlib.h>

main()
{
printf ("Primeiro Programa");
system("pause");
}
```

O resultado da execução desse programa



Exemplo2:

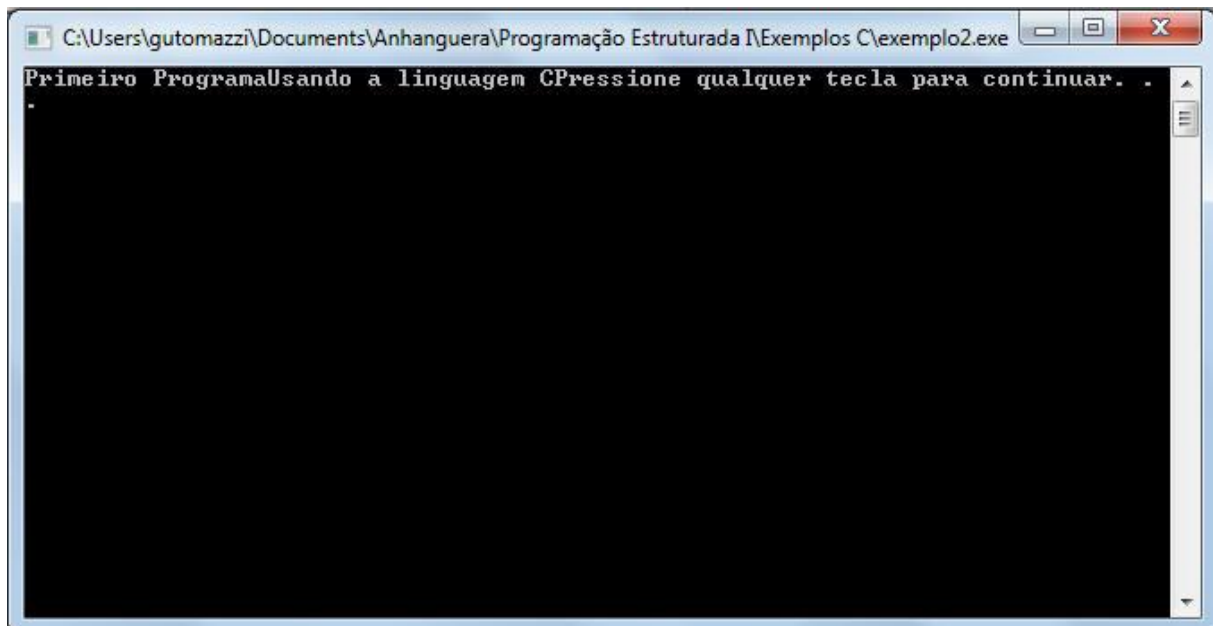
```
#include<stdio.h>
#include <stdlib.h>

main()
{
printf ("Primeiro Programa");
printf ("usando a linguagem C");
system("pause");
}
```

Linguagem C

Ausanello

O resultado da execução desse programa adiciona as duas mensagens na mesma linha. Vamos alterar esse mesmo código.

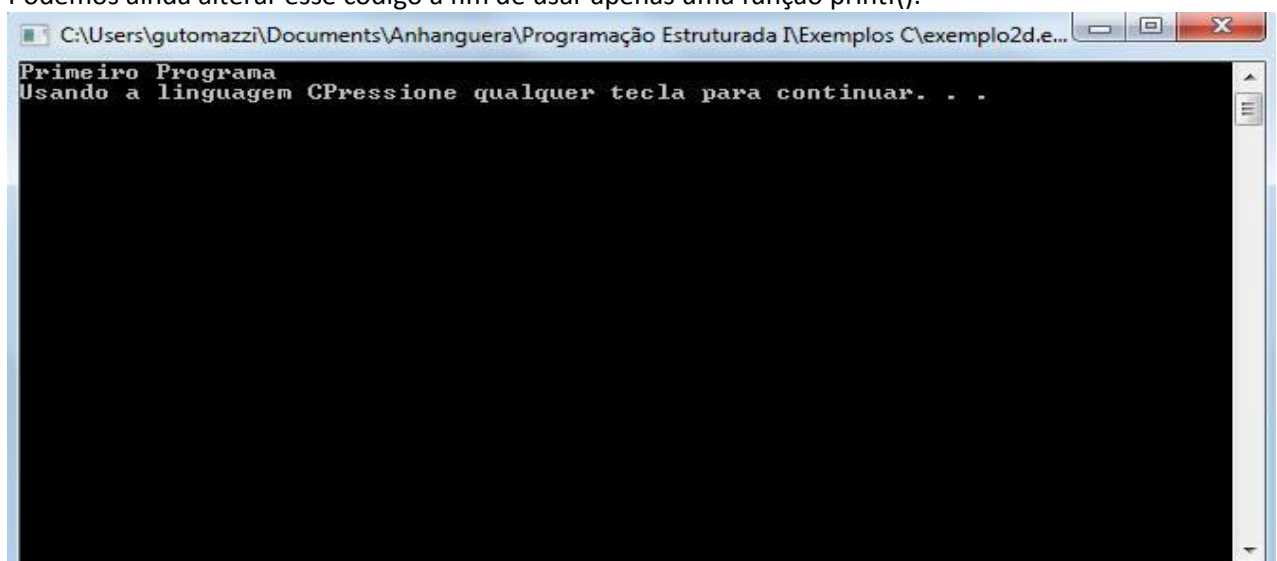


```
#include<stdio.h>
#include <stdlib.h>
```

```
main()
{
    printf ("Primeiro Programa \n");
    printf ("usando a linguagem C");
    system("pause");
}
```

O \n é um código especial que indica a função printf() que o restante da impressão deve ser feito em uma nova linha.

Podemos ainda alterar esse código a fim de usar apenas uma função printf().



```
#include<stdio.h>
#include <stdlib.h>

main()
{
printf ("Primeiro Programa \n usando a linguagem C") ;
system("pause");
}
```

Outro comando que muitas vezes temos que colocar para que seja possível visualizar com tempo o resultado do programa é o `system("pause");`. Ele deve ser colocado antes de fechar a função principal.

```
#include<stdio.h>
#include <stdlib.h>

main()
{
printf ("Primeiro Programa \n usando a linguagem C") ;
    system("pause");
}
```

Diretiva #include.

A diretiva `#include` diz ao compilador que ele deve incluir um arquivo externo na hora da compilação, portanto essa diretiva causa a inclusão de um programa fonte no seu programa.

Em alguns ambientes de programação o uso da diretiva `#include` é necessário para que os comandos sejam localizados e executados.

Um exemplo disso é o arquivo `stdio.h`, que contém as principais funções de entrada e saída.

Por isso para utilizar a função `printf()` é necessário utilizar a diretiva `#include <stdio.h>` e para o comando `system()` a `#include <stdlib.h>`

Exemplo:

```
#include <stdio.h>
#include <stdlib.h>
main()
{
printf ("Primeiro Programa \n usando a linguagem C") ;
/* Comando para pausar o programa */
    system("pause");
}
```

A tabela a seguir apresenta alguns dos principais .h da linguagem C:

Arquivo	Descrição
stdio.h	Funções de entrada e saída (I/O)
string.h	Funções de tratamento de strings
math.h	Funções matemáticas
ctype.h	Funções de teste e tratamento de caracteres
stdlib.h	Funções de uso genérico

Variáveis

Dados e Variáveis:

Variável: “Um dado é classificado como uma variável, quando tem a possibilidade de ser alterado em algum instante no decorrer do tempo, ou seja, durante a execução do programa”.

Exemplo: Cotação do dólar, o peso de uma pessoa, o índice da inflação.

Definição técnica de variável:

“Uma variável em C é um espaço de memória reservado para armazenar um certo tipo de dado e tendo um nome para referenciar o seu conteúdo”.

Nomes de Variáveis:

No desenvolvimento de um programa é necessário inserir ações que serão executadas pelo computador. Essas ações irão manipular os dados, sendo eles constantes e/ou variáveis. Para representar as variáveis nas linguagens de programação, devemos criar identificadores, nomes que representarão esses valores.

Para a formação de um nome de variável devemos acompanhar as seguintes regras:

1. Devem começar por um caractere alfabético ou por _;
2. Podem ser seguidos por mais caracteres alfabéticos, numéricos ou _;
3. Não devem ser usados caracteres especiais (!@#\$%^&*()+,etc);
4. Deve associar o nome com o que ele armazena.
5. Não utilizar palavras-chaves da linguagem

Exemplos:

a. Identificadores válidos:

Alpha, X, BJ153, K7, notas, MEDIA, ABC, INPS, FGTS, Dolar, Peso, Inflacao

b. Identificadores inválidos:

5X, E(13), A:B, X-Y, Nota/2, AWQ*, P&AA

Tipos de Variáveis

“O tipo de uma variável informa a quantidade de memória, que está irá ocupar e a forma como seu conteúdo será armazenada”.

Principais tipos de variáveis da linguagem C

Tipo	Característica
char	Um caractere
int	Números inteiros
float	Números com ponto flutuante
double	Números com ponto flutuante de precisão dupla
Void	Sem valor

Declarando Variáveis na linguagem C

As variáveis na linguagem C devem ser declaradas antes de serem usadas, para que possa ser reservado espaço na memória do computador.

Sintaxe geral da declaração de variáveis:

Tipo_da_variável Lista_de_variáveis

Exemplo:

int Idade, n_filhos

double salario

No exemplo acima foi declarada as variáveis: Idade e n_filhos do tipo int e salario do tipo double

Inicializando variáveis

Podemos inicializar variáveis no momento de sua declaração. Isto é importante, pois quando o C cria uma variável ela não é inicializada.

abaixo:

int Idade =18, n_filhos = 0;

double salário = 100.00;

Utilizando a função printf() para mostrar o conteúdo de uma variável.

A função printf() tem a seguinte sintaxe:

Printf("expressão de controle",lista de argumentos);

A expressão de controle contém a mensagem que será mostrada na tela. Já vimos essa função nos exemplos anteriores.

Além da mensagem, a expressão de controle mostra também o conteúdo das variáveis. Isso é feito utilizando a lista de argumentos e utilizando a notação %

Vamos analisar o resultado do programa abaixo sem o uso da notação % e sem a lista de argumentos.


```
#include <stdio.h>
#include <stdlib.h>

main() {

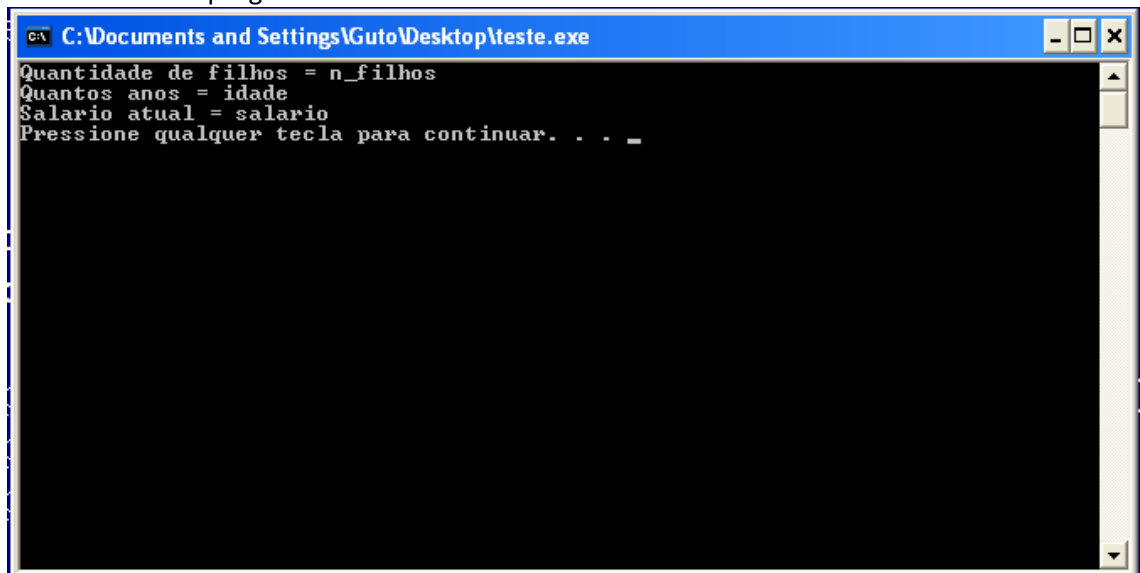
    int idade = 18, n_filhos = 0;
    double salario = 100.00;

    printf("Quantidade de filhos = n_filhos\n");
    printf("Quantos anos = idade\n");
    printf("Salario atual = salario\n");

    system("pause");

}
```

Resultado desse programa



Para que o conteúdo das variáveis possa ser mostrada devemos utilizar a notação % e a lista de argumentos.

Além do símbolo % devemos informar para cada tipo de variável um código de %.

Código	Siginifica
%d	Inteiro
%f	Float
%c	Caractere
%%	Coloca na Tela um %
%s	String

Vamos alterar o programa utilizando agora a lista de argumentos.

```
#include <stdio.h>
#include <stdlib.h>

main(){

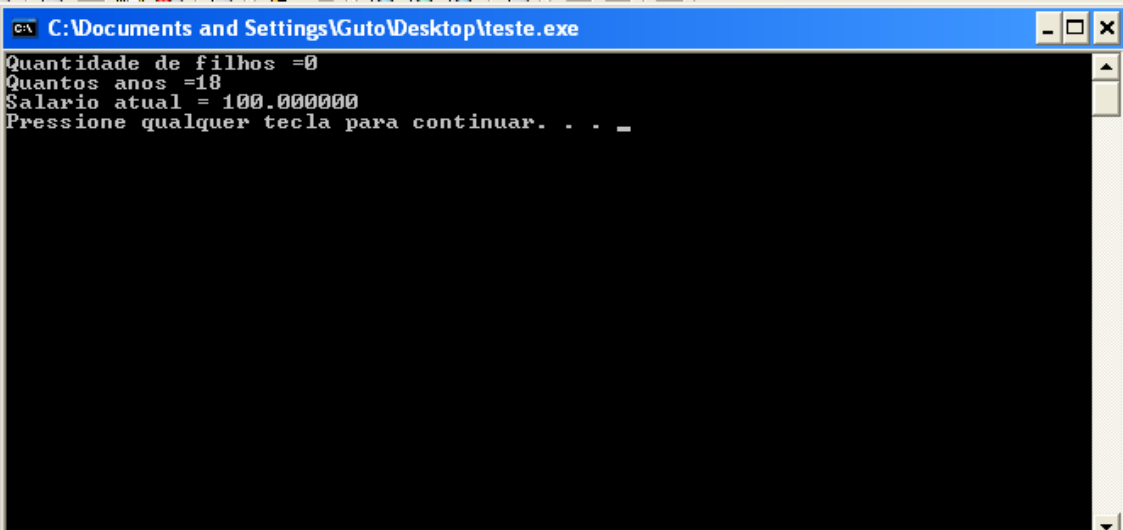
    int idade = 18, n_filhos = 0;
    double salario = 100.00;

    printf("Quantidade de filhos =%d\n", n_filhos);
    printf("Quantos anos =%d\n", idade);
    printf("Salario atual = %f\n",salario);

    system("pause");

}
```

Resultado desse programa



The screenshot shows a Windows command prompt window titled "C:\Documents and Settings\Guto\Desktop\teste.exe". The output of the program is displayed as follows:

```
Quantidade de filhos =0
Quantos anos =18
Salario atual = 100.000000
Pressione qualquer tecla para continuar. . . _
```

Exemplo 2

```
#include <stdio.h>
#include <stdlib.h>

main() {

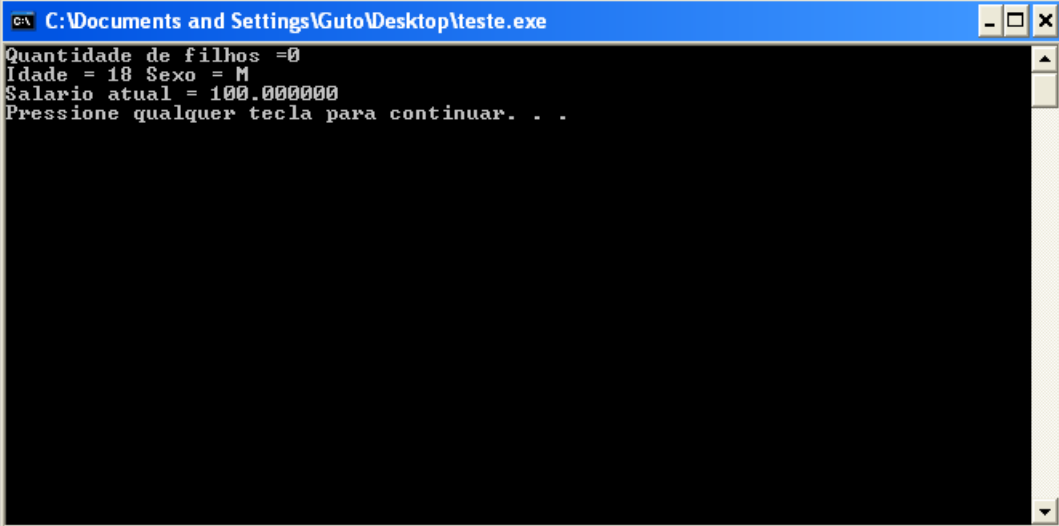
    int idade = 18, n_filhos = 0;
    double salario = 100.00;
    char sexo = 'M';

    printf("Quantidade de filhos =%d \n", n_filhos);
    printf("Idade = %d Sexo = %c \n", idade, sexo);
    printf("Salario atual = %f \n", salario);

    system("pause");

}
```

Resultado desse programa

A screenshot of a Windows command prompt window. The title bar reads "C:\Documents and Settings\Guto\Desktop\teste.exe". The window contains the following text: "Quantidade de filhos =0", "Idade = 18 Sexo = M", "Salario atual = 100.000000", and "Pressione qualquer tecla para continuar. . .". The text is displayed in a monospaced font on a black background.

Comando de Entrada de Dados

Função scanf():

A função scanf() é responsável por ler dados formatados da entrada padrão (teclado). Esses dados serão digitados pelo usuário e armazenados em cada uma das variáveis do programa.

Sintaxe:

Scanf(expressão_de_controle, lista_de_argumentos)

Usando a função scanf() podemos pedir dados ao usuário. Devemos ficar atentos em colocar o mesmo número de argumentos que o código de controle na expressão de controle. Outra coisa importante é colocar o & antes das variáveis da lista de argumentos.

Exemplo 1 :

```
#include <stdio.h>
#include <stdlib.h>

main() {

    double salario = 0;

    printf("Digite se Salario\n");

    scanf("%lf",&salario);

    printf("Voce digitou %f \n",salario);

    system("pause");

}
```

Para cada tipo de informação devemos utilizar um código específico.

No exemplo anterior a linha

```
scanf("%lf",&salario);
```

,define que o valor digitado é do tipo Double “%lf” e que deve ser armazenada na variável salario. O símbolo & é chamado de operador de endereço e devemos utilizá-lo junto com a variável.

Códigos de Formatação da Função scanf()

Código	Função
%c	Leia um Caractere
%d	Leia um inteiro
%f	Leia um número em ponto flutuante
%lf	Leia um dobre

Operadores aritméticos

Para efetuar cálculos com a linguagem C utilizamos os seguintes símbolos:

Símbolo	Função
=	Atribuição
+	Soma
-	Subtração
*	Multipliação
/	Divisão
%	Módulo(resto da divisão)

Exemplo 2

```
#include <stdio.h>
#include <stdlib.h>

main() {
    float anos, dias;

    printf("Digite sua idade em anos:\n");
    scanf("%f", &anos);

    dias = anos*365;

    printf("Sua idade em dias e %f\n", dias);

    system("pause");
}
```

A linha `dias = anos*365;` indica que será atribuído (=) a variável dias o cálculo `anos*365`, onde a variável anos contém o valor que o usuário digitou, multiplicado (*) por 365 que é o total de dias de um ano. O resultado dessa conta será mostrado para o usuário.

Atividade

Como atividade final, faça os exercícios abaixo.

Exercício 1

Quais dos seguintes nomes são válidos para variáveis em C?

- a) 3ab
- b) _sim
- c) N_a_o
- d) 00FIM
- e) int
- f) A123
- g) x**x
- h) ____A
- i) Y-2
- j) OOFIM

k) \meu

l) *y2

Exercício 2

Faça um algoritmo que receba quatro números inteiros, calcule e mostre a soma desses números.

Exercício 3

Faça um programa que receba três notas, calcule e mostre a Média aritmética entre elas.

Exercício 4

Faça um programa que calcule e mostre a Área de um triângulo
Sabe-se que : $\text{Área} = (\text{base} * \text{altura}) / 2$

Exercício 5

Faça um programa que calcule e mostre a Área de um trapézio.
Sabe-se que $\text{Área} = ((\text{base maior} + \text{base menor}) * \text{altura}) / 2$