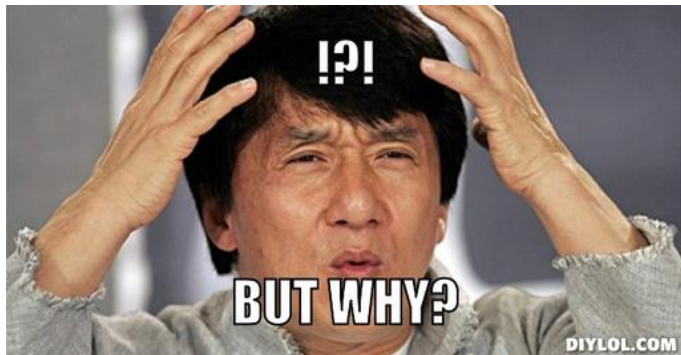


Algoritmos

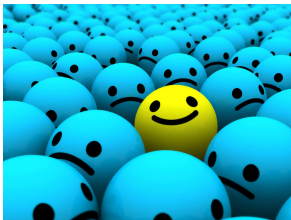
Introdução

Guilherme Meira

Primeiramente...



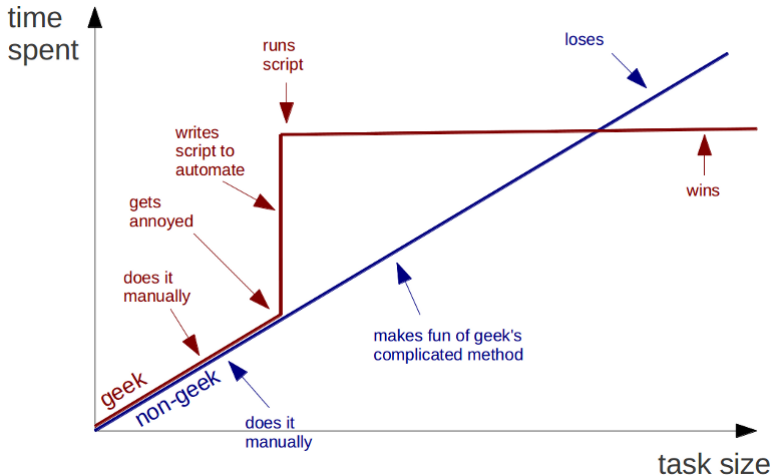
Primeiramente...



Primeiramente...

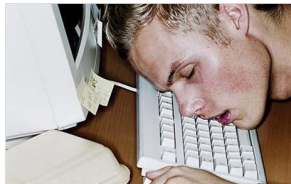


Primeiramento...



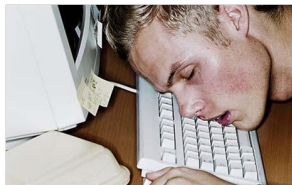
Primeiramente...

- Tarefas repetitivas



Primeiramente...

- Tarefas repetitivas
- Matemática



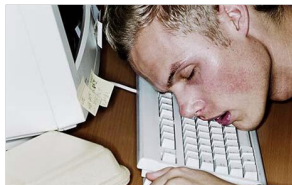
$$\int_0^{\infty} e^{-ax^2} (1+x^2)^{z-1/2} \cos(2ax + (2z-1)\tan^{-1}(x)) dx$$

Primeiramente...

- Tarefas repetitivas
- Matemática

$$\int_0^{\infty} e^{-ax^2} (1+x^2)^{z-1/2} \cos(2ax + (2z-1)\tan^{-1}(x)) dx$$

- Estatística

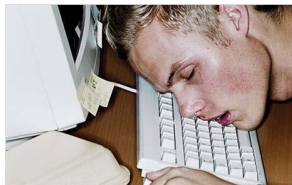


Primeiramente...

- Tarefas repetitivas
- Matemática

$$\int_0^{\infty} e^{-ax^2} (1+x^2)^{z-1/2} \cos(2ax + (2z-1)\tan^{-1}(x)) dx$$

- Estatística
- Formatura



Agenda

1. Apresentação da disciplina
2. Introdução
3. Variáveis
4. Operações
5. Entrada de dados
6. Exercícios

Apresentação da disciplina

Disciplina Algoritmos

Professor Guilherme Meira

Horário Segundas-feiras, das 18:50h às 22:00h

- Intervalo de 10 minutos por volta das 20:20h
- Chamada ao final da aula

Avaliação Duas provas + listas de exercício

- 7 pontos de prova (P_1 e P_2)
- 3 pontos de listas de exercício (T_1 e T_2)
- $M_P = \frac{P_1 + T_1 + P_2 + T_2}{2}$
- $M_P \geq 7$: aprovado
- $M_P < 7$: prova final (P_F)
- $M_F = \frac{M_P + P_F}{2}$
- $M_F \geq 5$: aprovado
- $M_F < 5$: gostou tanto da matéria que vai fazer de novo

Apresentação da disciplina

Trabalhos Exercícios práticos de programação

- Individual
- Aproximadamente a cada 2 semanas
- Correção automática

Livros Há muito material disponível na internet

- Livros da bibliografia do curso
- C Completo e Total (7 exemplares na biblioteca)

Outros

- Honestidade acadêmica
- Comportamento em sala
- Feedback!

Agenda

1. Apresentação da disciplina
- 2. Introdução**
3. Variáveis
4. Operações
5. Entrada de dados
6. Exercícios

Introdução

O que é um Algoritmo?

Introdução

O que é um Algoritmo?

Definição

Um **algoritmo** é uma especificação não-ambígua de como resolver uma classe de problemas.

Introdução

O que é um Algoritmo?



Definição

Um **algoritmo** é uma especificação não-ambígua de como resolver uma classe de problemas.

Introdução

O que é um Algoritmo?

Exemplo:

Problema Fazer bolo

- Algoritmo**
- Bata as claras em neve e reserve
 - Misture as gemas, a margarina e o açúcar até obter uma massa homogênea
 - Acrescente o leite e a farinha de trigo aos poucos, sem parar de bater
 - Por último, adicione as claras em neve e o fermento
 - Despeje a massa em uma forma grande de furo central untada e enfarinhada
 - Asse em forno médio 180 °C, preaquecido, por aproximadamente 40 minutos ou ao furar o bolo com um garfo, este saia limpo

Introdução

O que é um Algoritmo?

Exemplo:

Problema Calcular as raízes da equação $a \cdot x^2 + b \cdot x + c = 0$

Algoritmo

- Calcular o valor de $\Delta = b^2 - 4 \cdot a \cdot c$
- Usar o valor de delta para calcular as raízes:

$$x = -\frac{b \pm \sqrt{\Delta}}{2 \cdot a}$$

Introdução

O que é um Algoritmo?

Exemplo:

Problema Calcular as raízes da equação $a \cdot x^2 + b \cdot x + c = 0$

Algoritmo

- Calcular o valor de $\Delta = b^2 - 4 \cdot a \cdot c$
- Usar o valor de delta para calcular as raízes:

$$x = -\frac{b \pm \sqrt{\Delta}}{2 \cdot a}$$

Queremos ensinar ao computador como resolver tarefas por meio de algoritmos.

Introdução

Como ensinar um computador?

Computadores são capazes de executar pequenas tarefas chamadas de **instruções**.

Devemos descrever algoritmos por meio de uma sequência de instruções que o computador é capaz de executar.

```
; Hello World for Intel Assembler (MSDOS)
```

```
mov ax,cs  
mov ds,ax  
mov ah,9  
mov dx, offset Hello  
int 21h  
xor ax,ax  
int 21h
```

```
Hello:
```

```
    db "Hello World!",13,10,"$"
```

Introdução

Como ensinar um computador?

Computadores são capazes de executar pequenas tarefas chamadas de **instruções**.

Devemos descrever algoritmos por meio de uma sequência de instruções que o computador é capaz de executar.



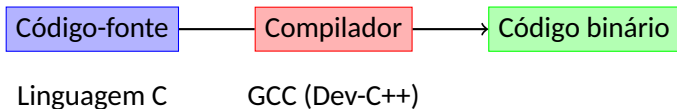
Introdução

Como ensinar um computador?



Introdução

Como ensinar um computador?



Introdução

Linguagem C

- Desenvolvida por Dennis Ritchie entre 1969 e 1973
- Presente em **muitas** plataformas
- Baixo nível
- Utilizam C:
 - Sistemas operacionais (Windows, Linux)
 - Jogos (Doom, Quake)
 - Bancos de dados (MySQL, Oracle)
 - Linguagens de programação (Python, Ruby)
 - Muitos outros...



Introdução

Nosso compilador

- Dev-C++
 - Ambiente de Desenvolvimento Integrado (IDE)
 - Contém o compilador GCC
 - Suporta C e C++
 - Gratuito e fácil de instalar
 - <http://www.bloodshed.net/devcpp.html>
- Outras opções:
 - Code::Blocks (<http://www.codeblocks.org/>)
 - Microsoft Visual Studio (<https://www.visualstudio.com/>)
 - GCC via Cygwin (<http://www.cygwin.com/> - avançado)

Introdução

Olá, mundo!

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

```
int main() {  
    printf("Olá, mundo!\n");  
    return 0;  
}
```

Introdução

Olá, mundo!

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

```
int main() {  
    printf("Olá, mundo!\n");  
    return 0;  
}
```

Inclusão de `stdio.h` da biblioteca padrão.

Introdução

Olá, mundo!

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

```
int main() {  
    printf("Olá, mundo!\n");  
    return 0;  
}
```

O programa começa a executar daqui.

Introdução

Olá, mundo!

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

```
int main() {  
    printf("Olá, mundo!\n");  
    return 0;  
}
```

Textos entre aspas são chamados de **strings**. (O que é \n?)

Introdução

Olá, mundo!

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

```
int main() {  
    printf("Olá, mundo!\n");  
    return 0;  
}
```

Terminamos a execução. Zero indica que tudo correu bem.

Introdução

Caracteres especiais

Alguns caracteres em uma string tem representação especial chamada **sequência de escape**.

Errado

```
printf("Olá, mundo!  
Bom dia!");
```

Correto

```
printf("Olá, mundo!\nBom dia!\n");
```


Introdução

Caracteres especiais

Principais sequências de escape em C:

Caractere	Sequência de escape
Quebra de linha	\n
Retorno de carro	\r
Tabulação	\t
Contra-barra	\\
Aspas duplas	\"

Introdução

Caracteres especiais

```
printf("Alice disse:\n\t\"Hoje é 19\\02!\"\\n");
```

Saída

Alice disse:

 "Hoje é 19\02!"

Introdução

Comentários

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

```
/* Esta função imprime Olá, mundo.
```

```
Isto é um comentário ocupando várias linhas. */
```

```
int main() {
```

```
//Este comentário só vai até o fim da linha
```

```
printf("Olá, mundo!\n");
```

```
return 0;
```

```
}
```

Introdução

Comentários

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

```
/* Esta função imprime Olá, mundo.
```

```
    Isto é um comentário ocupando várias linhas. */
```

```
int main() {
```

```
    //Este comentário só vai até o fim da linha
```

```
    printf("Olá, mundo!\n");
```

```
    return 0;
```

```
}
```

Introdução

Comentários

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

```
/* Esta função imprime Olá, mundo.
```

```
Isto é um comentário ocupando várias linhas. */
```

```
int main() {
```

```
//Este comentário só vai até o fim da linha
```

```
printf("Olá, mundo!\n");
```

```
return 0;
```

```
}
```

Agenda

1. Apresentação da disciplina
2. Introdução
- 3. Variáveis**
4. Operações
5. Entrada de dados
6. Exercícios

Variáveis

Tipos

```
int main() {  
    int idade = 27;  
    return 0;  
}
```



Variáveis

Tipos

Diversos tamanhos, chamados de **tipos**:

int Números inteiros. Em geral, pode armazenar valores de -2.147.483.648 até 2.147.483.647.

char Geralmente utilizado para armazenar caracteres. Valores de -128 a 127.

float Números decimais. Valores entre $\pm 3.4 \cdot 10^{\pm 38}$ com cerca de 7 casas decimais.

double Números decimais de precisão dupla. Valores entre $\pm 1.7 \cdot 10^{\pm 308}$ com cerca de 15 casas decimais.

Variáveis

Tipos

Outros tipos:

short Números inteiros, valores entre -32.768 e 32.767.

long Números inteiros, valores entre -9.223.372.036.854.775.808 e 9.223.372.036.854.775.807.

O modificador **unsigned** cria uma variável que só armazena números positivos (somente tipos inteiros).

char: -128 a 127 \Rightarrow unsigned char: 0 a 255

Todos esses valores podem variar dependendo da plataforma.

Variáveis

Tipos

Outro

short

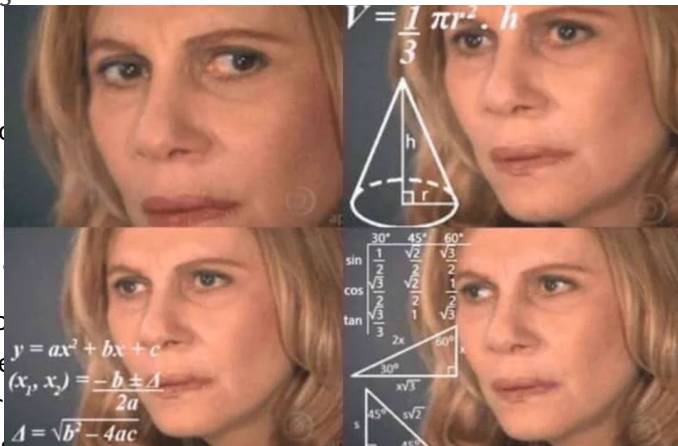
long

O mo

núme

char

Todos esses valores podem variar dependendo da plataforma.



08 e

Variáveis

Tipos

Regra geral:

- Use `int` para números inteiros.
- Use `char` para caracteres.
- Use `double` para números decimais.

Variáveis

Tipos

Regra geral:

- Use `int` para números inteiros.
- Use `char` para caracteres.
- Use `double` para números decimais.

Por que não `double` para tudo?

Variáveis

Tipos

Regra geral:

- Use `int` para números inteiros.
- Use `char` para caracteres.
- Use `double` para números decimais.

Por que não `double` para tudo?

- Operações com números decimais são mais lentas
- Erros de arredondamento

Variáveis

Identificadores

Os nomes dados às variáveis são chamados de **identificadores**.

Regras para os identificadores:

- Começar com uma letra ou underscore (`_`)
- Os próximos caracteres podem ser letras, números ou underscores
- Não devem ser palavras reservadas (`return`, `int`, ...)

Use **nomes bem descritivos** para suas variáveis (prefira **altura** em vez de **a**);

Variáveis

Exibição

```
//Include, main, etc...
```

```
int i = 27;
```

```
char l = 'J';
```

```
double a = 1.73;
```

```
printf("Idade: %d Letra: %c Altura: %lf.\n", i, l, a);
```

Saída

```
Idade: 27 Letra: J Altura: 1.730000.
```

Variáveis

Exibição

Tipo	Exibição
int	%d
unsigned int	%u
float	%f
double	%lf
char	%c
long	%ld
unsigned long	%lu

Variáveis

Exibição

Tipo	Exibição
int	%d
unsigned int	%u
float	%f
double	%lf
char	%c
long	%ld
unsigned long	%lu

E o caractere %?

Variáveis

Exibição

Tipo	Exibição
int	%d
unsigned int	%u
float	%f
double	%lf
char	%c
long	%ld
unsigned long	%lu

E o caractere %? Usamos %%.

Variáveis

Exibição

Saída

Idade: 27 Letra: J Altura: 1.730000.

Como esconder esses zeros?

Variáveis

Exibição

```
double a = 1.73;  
printf("Normal: %lf\n", a);  
printf("2 casas: %.2lf\n", a);  
printf("Com zeros: %05.2lf\n", a);  
printf("Com espaços: %5.2lf\n", a);
```

Saída

```
Normal: 1.730000  
2 casas: 1.73  
Com zeros: 01.73  
Com espaços:  1.73
```

Variáveis

Exibição

```
double a = 1.73;  
printf("Normal: %lf\n", a);  
printf("2 casas: %.2lf\n", a);  
printf("Com zeros: %05.2lf\n", a);  
printf("Com espaços: %5.2lf\n", a);
```

Saída

```
Normal: 1.730000  
2 casas: 1.73  
Com zeros: 01.73  
Com espaços:  1.73
```

Variáveis

Exibição

```
double a = 1.73;  
printf("Normal: %lf\n", a);  
printf("2 casas: %.2lf\n", a);  
printf("Com zeros: %05.2lf\n", a);  
printf("Com espaços: %5.2lf\n", a);
```

Saída

```
Normal: 1.730000  
2 casas: 1.73  
Com zeros: 01.73  
Com espaços:  1.73
```

Variáveis

Exibição

```
double a = 1.73;  
printf("Normal: %lf\n", a);  
printf("2 casas: %.2lf\n", a);  
printf("Com zeros: %05.2lf\n", a);  
printf("Com espaços: %5.2lf\n", a);
```

Saída

```
Normal: 1.730000  
2 casas: 1.73  
Com zeros: 01.73  
Com espaços:  1.73
```

Variáveis

Exibição

```
double a = 1.73;  
printf("Normal: %lf\n", a);  
printf("2 casas: %.2lf\n", a);  
printf("Com zeros: %05.2lf\n", a);  
printf("Com espaços: %5.2lf\n", a);
```

Saída

```
Normal: 1.730000  
2 casas: 1.73  
Com zeros: 01.73  
Com espaços:  1.73
```


Agenda

1. Apresentação da disciplina
2. Introdução
3. Variáveis
- 4. Operações**
5. Entrada de dados
6. Exercícios

Operações

As principais operações matemáticas estão disponíveis:

- **Adição:** +
- **Subtração:** -
- **Multiplicação:** *
- **Divisão:** /

Valem as regras de precedência da matemática:

1. Multiplicações e divisões da esquerda para a direita
2. Somas e subtrações da esquerda para a direita

Operações

```
//Include, main, etc...
```

```
int a = 12+3*4;
```

```
printf("a = %d\n", a);
```

Operações

```
//Include, main, etc...
```

```
int a = 12+3*4;  
printf("a = %d\n", a);
```

Saída

a = 24

Operações

Podemos alterar a ordem das operações usando parênteses:

```
int a = (12+3)*4;
```

```
int b = (12+(3-2)*5)*4;
```

```
printf("a = %d b = %d\n", a, b);
```

Saída

```
a = 60 b = 68
```

Operações

A divisão entre dois inteiros será também inteira:

```
int resultInt = 5/2;
```

```
double resultDouble = 5.0/2.0;
```

```
printf("int = %d double = %lf\n",  
      resultInt, resultDouble);
```

Saída

```
int = 2 double = 2.500000
```

Operações

O operador % calcula o resto da divisão:

```
int r = 10%6;
```

```
printf("r = %d\n", r);
```

Saída

```
r = 4
```

Operações

Este operador pode ser usado em operações de “aritmética de relógio”:

```
int agora = 20;
```

```
int horas = 9;
```

```
int depois = (20+9)%24;
```

```
printf("Agora são %d horas\n", agora);
```

```
printf("Daqui %d horas serão %d horas\n", horas, depois);
```

Saída

Agora são 20 horas

Daqui 9 horas serão 5 horas

Operações

A biblioteca `math.h` disponibiliza operações matemáticas:

- **Potenciação:** `pow`
- **Seno:** `sin`
- **Cosseno:** `cos`
- **Tangente:** `tan`
- **Raiz quadrada:** `sqrt`
- **Valor absoluto (inteiro):** `abs`
- **Valor absoluto (double):** `fabs`

Operações

A biblioteca `math.h` disponibiliza operações matemáticas:

- **Logaritmo na base 10:** `log10`
- **Logaritmo natural:** `log`
- **Exponencial (e^x):** `exp`
- **Arredondamento para cima:** `ceil`
- **Arredondamento para baixo:** `floor`
- Dentre várias outras

Operações

```
#include <stdio.h>
#include <math.h>
int main() {
    double c1 = 3;
    double c2 = 4;
    double hip = sqrt(pow(c1, 2) + pow(c2, 2));
    printf("Catetos: %lf e %lf Hipotenusa: %lf\n",
           c1, c2, hip);
    return 0;
}
```

Saída

Catetos: 3.000000 e 4.000000 Hipotenusa: 5.000000

Operações

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

```
#include <math.h>
```

```
int main() {  
    double c1 = 3;  
    double c2 = 4;  
    double hip = sqrt(pow(c1, 2) + pow(c2, 2));  
    printf("Catetos: %lf e %lf Hipotenusa: %lf\n",  
           c1, c2, hip);  
    return 0;  
}
```

Saída

```
Catetos: 3.000000 e 4.000000 Hipotenusa: 5.000000
```

Agenda

1. Apresentação da disciplina
2. Introdução
3. Variáveis
4. Operações
- 5. Entrada de dados**
6. Exercícios

Entrada de dados

A principal função que usaremos é a `scanf`.

```
int numero;  
printf("Digite um número: ");  
scanf("%d",&numero);  
int dobro = numero*2;  
printf("O dobro de %d é %d\n", numero, dobro);
```

Saída

```
Digite um número: 10  
O dobro de 10 é 20
```

Entrada de dados

A principal função que usaremos é a `scanf`.

```
int numero;  
printf("Digite um número: ");  
scanf("%d",&numero);  
int dobro = numero*2;  
printf("O dobro de %d é %d\n", numero, dobro);
```

Saída

```
Digite um número: 10  
O dobro de 10 é 20
```

Entrada de dados

Importante lembrar sobre a `scanf`:

- Utiliza os mesmos especificadores que vimos na `printf`
- Lembrar sempre de colocar o `&` antes do nome da variável (entenderemos o motivo mais adiante)

Agenda

1. Apresentação da disciplina
2. Introdução
3. Variáveis
4. Operações
5. Entrada de dados
- 6. Exercícios**

Exercícios

Exercício 1

Escreva um programa que imprima seu nome na tela. Cada nome deve estar em uma linha separada.

Exercícios

Exercício 1

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

```
int main() {  
    printf("Guilherme\nMeira\n");  
    return 0;  
}
```

Exercícios

Exercício 2

Escreva um programa que calcule o valor de:

$$\frac{26 + 2 \cdot 5}{(7 + 2) \cdot 4}$$

(1)

Exercícios

Exercício 2

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

```
int main() {  
    double valor = (26+2*5)/((7+2)*4);  
    printf("O valor é %lf\n", valor);  
    return 0;  
}
```

Exercícios

Exercício 3

Escreva um programa que receba um número pelo teclado e eleve o número ao quadrado.

Exercícios

Exercício 3

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

```
#include <math.h>
```

```
int main() {  
    int numero;  
    printf("Digite um número: ");  
    scanf("%d", &numero);  
    int quadrado = pow(numero, 2);  
    printf("O quadrado de %d é %d\n", numero, quadrado);  
    return 0;  
}
```

Exercícios

Exercício 4

Marina acaba de comprar um carro. O pagamento será feito da seguinte maneira:

- 40% do valor a vista
- O restante dividido em 10 parcelas

Escreva um programa que receba o valor do carro pelo teclado e imprima o valor a ser pago à vista e o valor de cada parcela.

Exercícios

Exercício 4

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

```
int main() {  
    double valor;  
    printf("Digite o valor do carro: ");  
    scanf("%lf", &valor);  
  
    double entrada = valor*0.4;  
    double restante = valor-entrada;  
    double parcela = restante/10;  
  
    printf("Valor da entrada: R$ %.2f\n", entrada);  
    printf("Valor das parcelas: R$ %.2f\n", parcela);  
    return 0;  
}
```