
Raciocínio Algorítmico e Fundamentos da Computação

— Danilo Rios —
14/03/2025

Chegamos nessa parte

- Na aula de hoje começaremos a programar

Linguagem

- Nessa disciplina iremos aprender a programar em C

Linguagem

- Por que em C?
 - Alguém já escutou falar dessa linguagem?
- Não tem várias outras que são mais famosas ou populares?

Linguagem C

- Não foi a primeira linguagem criada
- Surgiu nos anos 70
- É a base / inspiração de muitas linguagens de programação
 - Ao conhecer ela tem conhecimento para trabalhar com outras linguagens

Linguagem C

- Com certeza não existem tantas vagas abertas para programadores C
 - Diferentemente de vagas para programadores JAVA, .NET, e diversos javascript ...
- Mas as outras linguagens são para sistemas/aplicativos web ou mobile que alguém tem a ideia e quer produzir
 - E isso não é bem a realidade da linguagem que surgiu nos anos 70

Linguagem C

- É uma linguagem de baixo nível
 - Do mesmo modo que assembly, mas não tão baixo assim
 - Tem maior controle sobre os recursos do sistema
- Pode ser executada em qualquer plataforma que possuir um compilador C
- É eficiente e utilizada em ambientes que precisam de desempenho
 - Ex.: Sistemas embarcados e software de tempo real

Linguagem C

- O sistema operacional Unix/Linux é em C
 - Não inteiro, boa parte
 - Minix
 - Sistema operacional educativo
 - Tanenbaum

Linguagem C

- Linguagem compilada
- Fortemente tipada

Linguagem compilada?

- Utiliza um compilador
 - Programa que transforma o código escrito em um código com instruções para o computador
- Na etapa de compilação o código é verificado e erros são detectados
 - Todos os erros?
 - Não, é validada a estrutura do código
 - Alguns erros só ocorrem durante a execução

Linguagem fortemente tipada?

- A variável tem um tipo definido e isso não muda mais
 - Se eu falar que num lugar eu guardo número então só posso guardar número
 - Não é possível mudar e começar a guardar texto

Linguagem C

- Estrutura básica do código
 - Inclui a biblioteca stdio
 - Declaração da função main

```
1  #include <stdio.h>
2
3  int main() {
4      //código fonte
5      return 0;
6  }
```

Função

- O que é uma função?

Função

- O que é uma função?
 - Tem vários sentidos/significados para função dependendo do contexto
 - Função do professor da sala?
 - Função de 1º grau?
- O que é uma função na programação?

Função

- O que é uma função na programação?
 - É um bloco de código que realiza determinada funcionalidade
- main
 - É a função principal (tradução de main) do nosso código
 - Todo programa começa a ser executado a partir da função main

Biblioteca

- O que é uma biblioteca?

Biblioteca

- O que é uma biblioteca?
 - É o local onde tem/está o conhecimento
 - Precisamos desse conhecimento para não ter que criar tudo do zero
 - Alguém já fez, alguém já criou e então nós utilizamos isso que está pronto

Biblioteca

- O que é uma biblioteca?
 - É um arquivo que tem funções que foram programadas por alguém
 - Obs.: também podemos programar nossas próprias bibliotecas

Biblioteca

- `#include <stdio.h>`
- Standard Input Output
 - Padrão de entrada e saída
- Biblioteca que tem funções para realizar a entrada de informações e a saída de informações do programa

Biblioteca

- `#include <stdlib.h>`
- Standard Library
 - Biblioteca padrão
- Biblioteca com algumas funções primordiais
 - Gerar número aleatório
 - Funções para trabalhar com memória
 - Em estrutura de dados

Biblioteca

- `#include <math.h>`
- Math (Mathematic)
 - Matemática
- Biblioteca com funções matemáticas
 - Não são as 4 operações básicas
 - Calcular ângulo (seno, cosseno)
 - Calcular logaritmo
 - Calcular raiz
 - ...

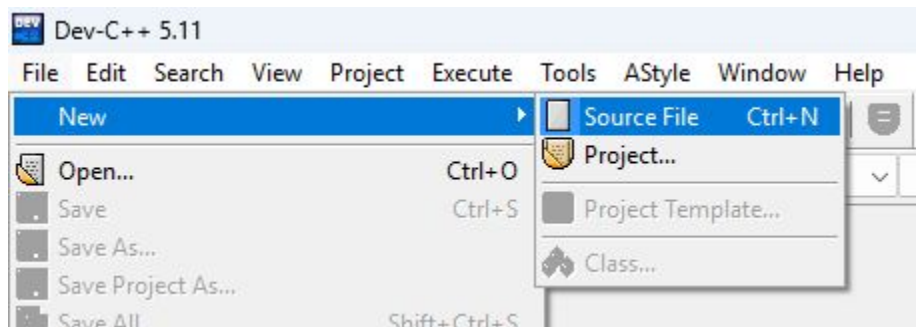
Como vamos programar?

- Vamos utilizar o software Dev C++
 - Com ele também é possível programar em C++, mas vamos programar em C
- É a única opção?
 - Não, mas é o mais leve e super simples de utilizar
 - Existem outros softwares que dá para programar em C
 - Também dá para programar com o bloco de notas, compilar e executar na linha de comando

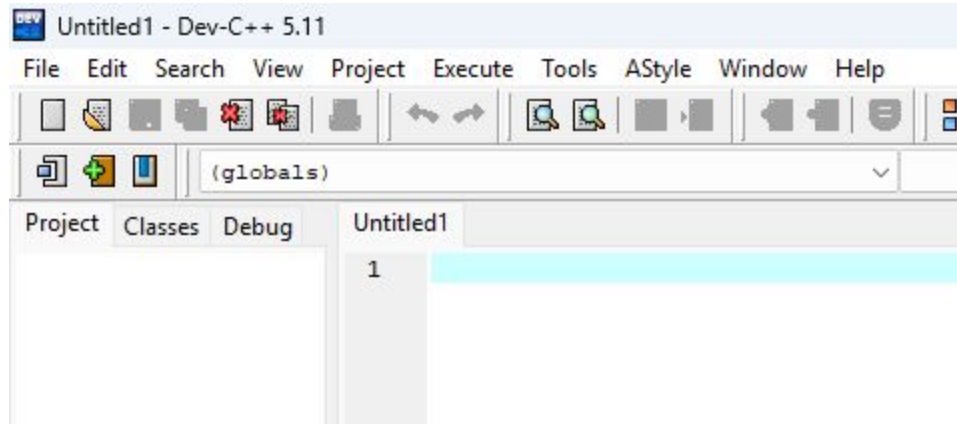


Dev C++

- Abrir o Dev C++
- Como criar um novo arquivo
 - File -> New -> Source File

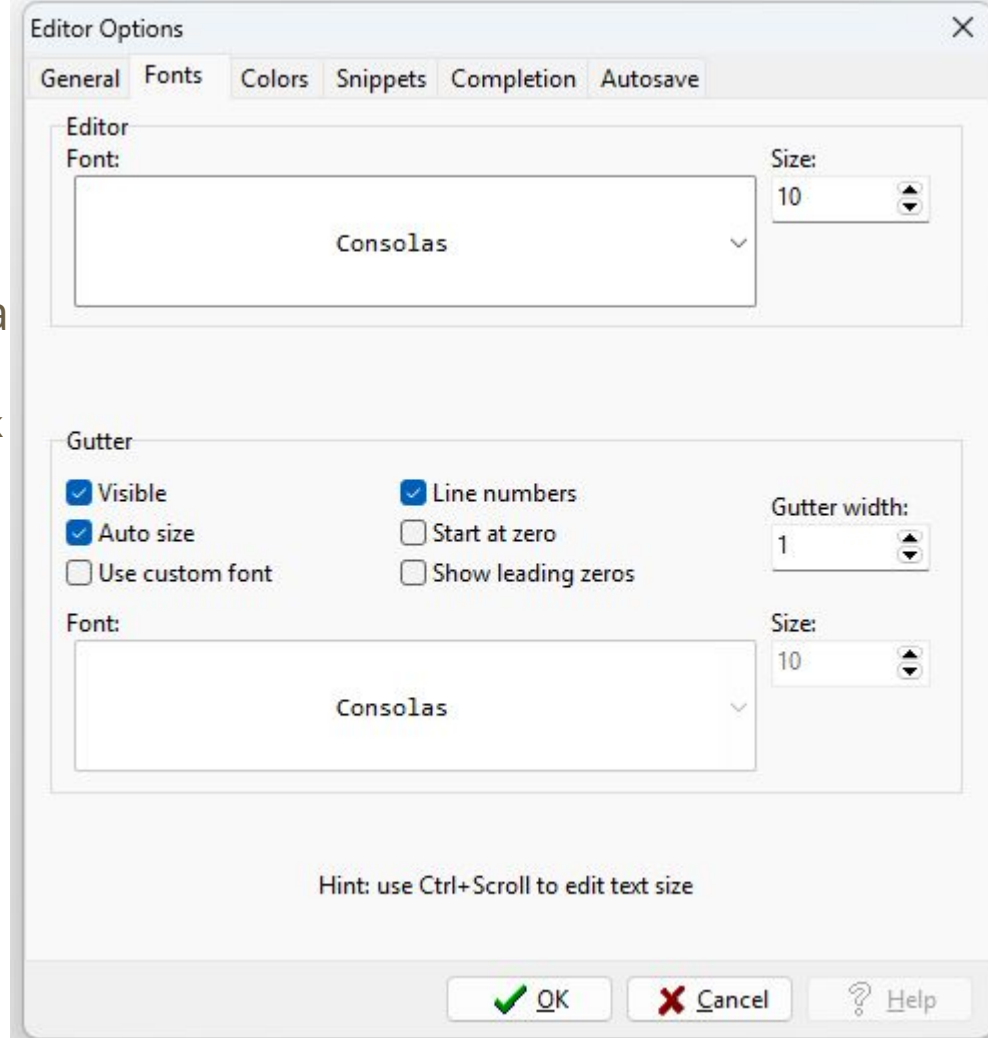
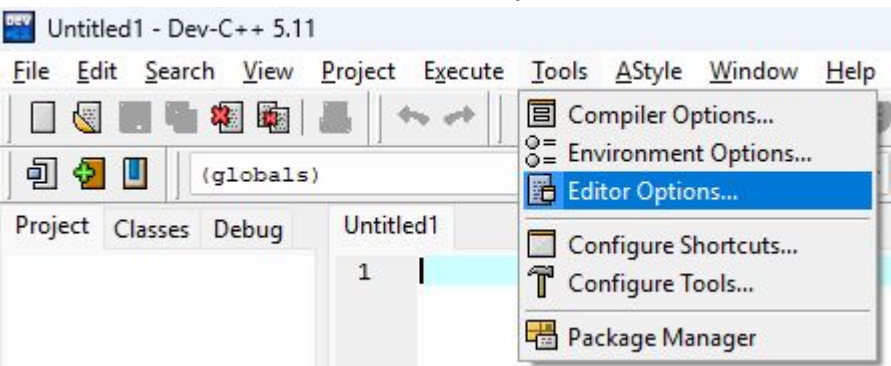


Dev C++



Dev C++

- Como habilitar o número da linha
 - Tools -> Editor Options
 - Fonts-> Marque "Line numbers" -> Ok
 - Varia dependendo da versão



O primeiro programa

- Sempre que começamos a aprender uma linguagem de programação começamos criando o mesmo programa
- O famoso “Hello World!”
 - Olá mundo!

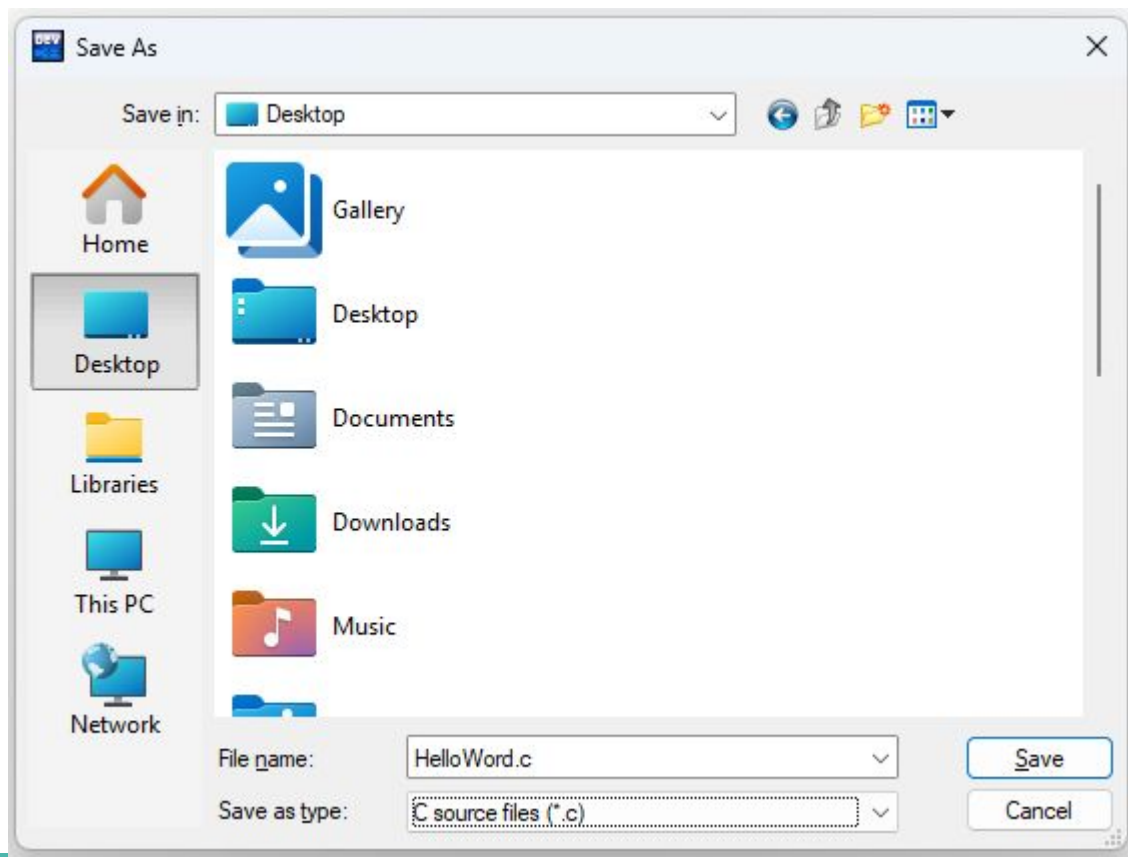
0 primeiro programa

- Incluir biblioteca stdio
- Incluir a função main
- Função printf, que existe na biblioteca stdio, para saída de informação do programa
 - Texto entre aspas
- Retorno da função

```
1  #include <stdio.h>
2
3  int main() {
4      printf("Hello World!");
5      return 0;
6  }
```

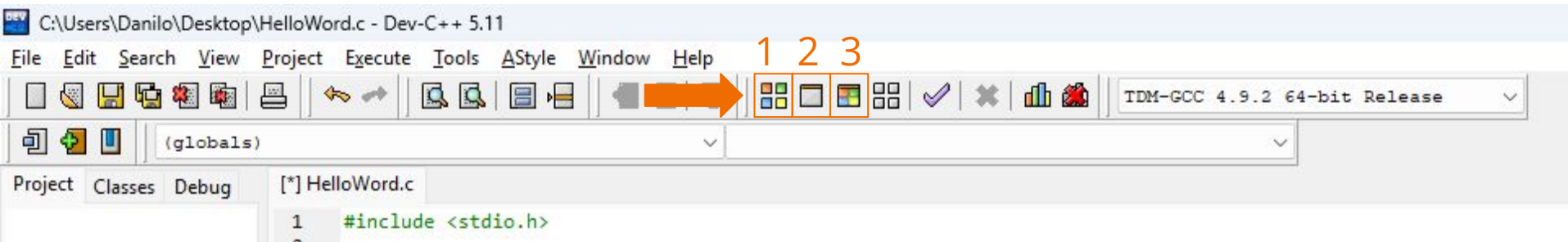
0 primeiro programa

- Salvar
- Escolher o tipo
 - C source files (*.c)



0 primeiro programa

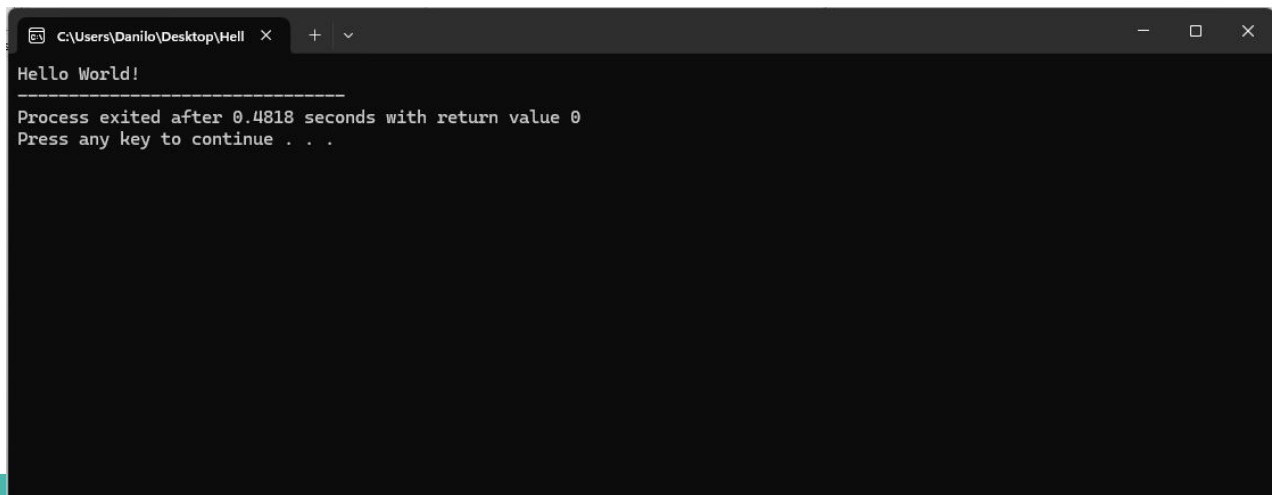
- Com o código pronto precisamos compilar e executar



- 1 - Compilar o código
- 2 - Executar o código compilado
- 3 - Compilar e executar o código
- Clicamos no 1 e depois o 2 OU só no 3

0 primeiro programa

- Abre um terminal com a saída do programa
 - Hello World!
 - Existem outras informações
 - Aperte qualquer tecla para continuar
 - Vai fechar o terminal



A screenshot of a Windows terminal window. The title bar shows the file path 'C:\Users\Danilo\Desktop\Hell' and standard window controls. The terminal output is as follows:

```
Hello World!  
-----  
Process exited after 0.4818 seconds with return value 0  
Press any key to continue . . .
```

printf

- printf é a nossa primeira função, além da main, que estamos aprendendo
- É utilizado para a saída da informação do sistema
 - Exibir essa informação na tela

printf

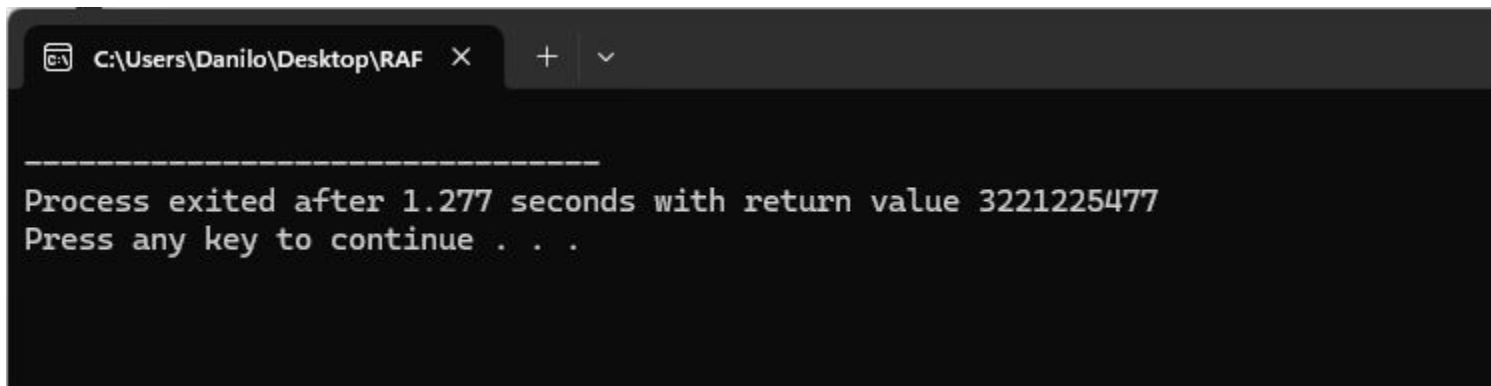
- O que esperamos de saída do seguinte código?

```
1  #include <stdio.h>
2
3  int main() {
4      printf(123);
5      return 0;
6  }
```


printf

- O que esperamos de saída do seguinte código?
 - Por que não apareceu 123 escrito no terminal?

```
1  #include <stdio.h>
2
3  int main() {
4      printf(123);
5      return 0;
6  }
```



```
C:\Users\Danilo\Desktop\RAF X + v

-----
Process exited after 1.277 seconds with return value 3221225477
Press any key to continue . . .
```

printf

- Por que não apareceu 123 escrito no terminal?
 - O printf espera exibir um texto e não um número
 - O que representa algo ser um texto?
 - Estar entre aspas

```
1  #include <stdio.h>
2
3  int main() {
4      printf("123");
5      return 0;
6  }
```

123

Process exited after 0.5137 seconds with return value 0
Press any key to continue . . .

printf

- Mas se eu fizer uma conta $2+2$ o resultado é um número e não um texto
- Não tem como exibir o resultado da conta?
 - Calma... vamos chegar neste ponto

printf

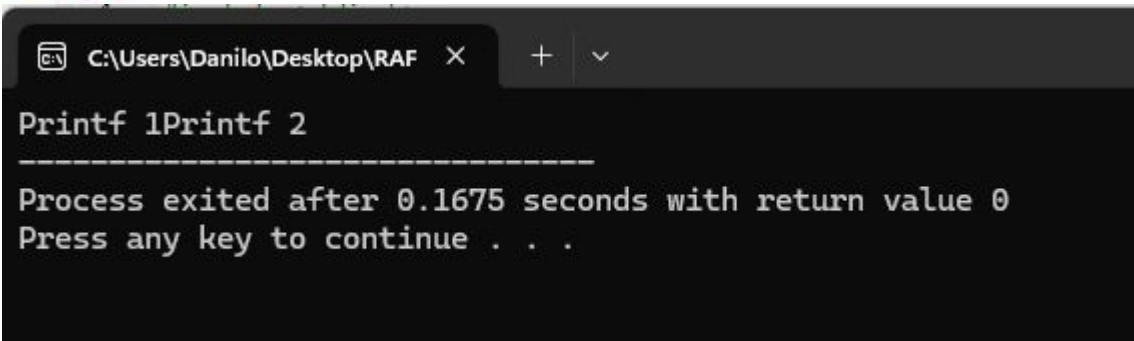
- O que esperamos de saída do seguinte código?
 - Cada texto vai sair em uma linha diferente?

```
1  #include <stdio.h>
2
3  int main() {
4      printf("Printf 1");
5      printf("Printf 2");
6      return 0;
7  }
```

printf

- O que esperamos de saída do seguinte código?
 - Cada texto vai sair em uma linha diferente?

```
1  #include <stdio.h>
2
3  int main() {
4      printf("Printf 1");
5      printf("Printf 2");
6      return 0;
7  }
```



The screenshot shows a Windows command prompt window with the title bar 'C:\Users\Danilo\Desktop\RAF'. The window displays the output of the program: 'Printf 1Printf 2' on the first line, followed by a horizontal separator line. Below the line, it says 'Process exited after 0.1675 seconds with return value 0' and 'Press any key to continue . . .'. The window has standard Windows controls (minimize, maximize, close) and a tab bar with a '+' icon and a dropdown arrow.

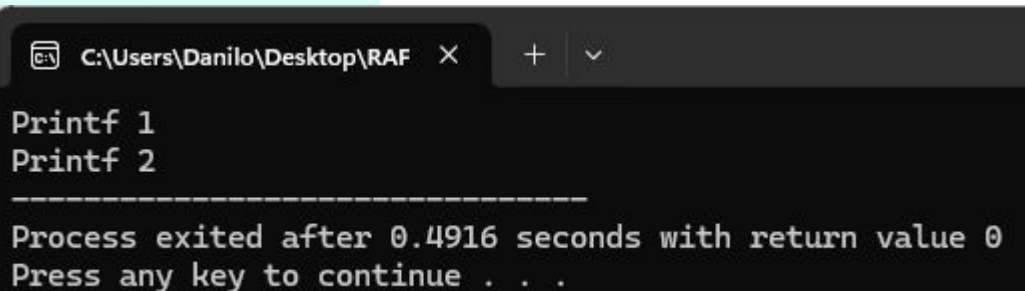
```
C:\Users\Danilo\Desktop\RAF X + v
Printf 1Printf 2
-----
Process exited after 0.1675 seconds with return value 0
Press any key to continue . . .
```

printf

- Colocar cada texto em um printf não faz pular de linha
- Para pular de linha precisamos utilizar o `\n` dentro das aspas

```
1  #include <stdio.h>
2
3  int main() {
4      printf("Printf 1\n");
5      printf("Printf 2");
6      return 0;
7  }
```

```
1  #include <stdio.h>
2
3  int main() {
4      printf("Printf 1\nPrintf 2");
5      return 0;
6  }
```



The screenshot shows a Windows command prompt window with the title bar "C:\Users\Danilo\Desktop\RAF". The window displays the output of the program: "Printf 1" followed by "Printf 2" on the next line. Below the output, a horizontal line is shown, followed by the text "Process exited after 0.4916 seconds with return value 0" and "Press any key to continue . . .".

```
C:\Users\Danilo\Desktop\RAF >
Printf 1
Printf 2
-----
Process exited after 0.4916 seconds with return value 0
Press any key to continue . . .
```

printf

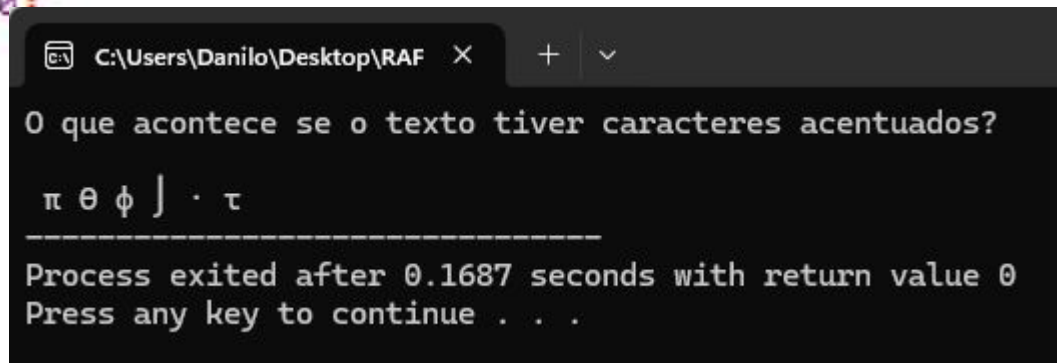
- O que esperamos de saída do seguinte código?

```
1  #include <stdio.h>
2
3  int main() {
4      printf("O que acontece se o texto tiver caracteres acentuados?\n\n");
5      printf(" ã é í ã ú ç ");
6
7      return 0;
8  }
```

printf

- O que esperamos de saída do seguinte código?

```
1  #include <stdio.h>
2
3  int main() {
4      printf("O que acontece se o texto tiver caracteres acentuados?\n\n");
5      printf(" ã é í ã ú ç ");
6
7      return 0;
8  }
```



A screenshot of a Windows command prompt window. The title bar shows the file path "C:\Users\Danilo\Desktop\RAF" and standard window controls. The output of the program is displayed in a monospaced font. It shows the first printf statement's output followed by a blank line, then the second printf statement's output. Below this, there is a horizontal line and a status message indicating the process has exited.

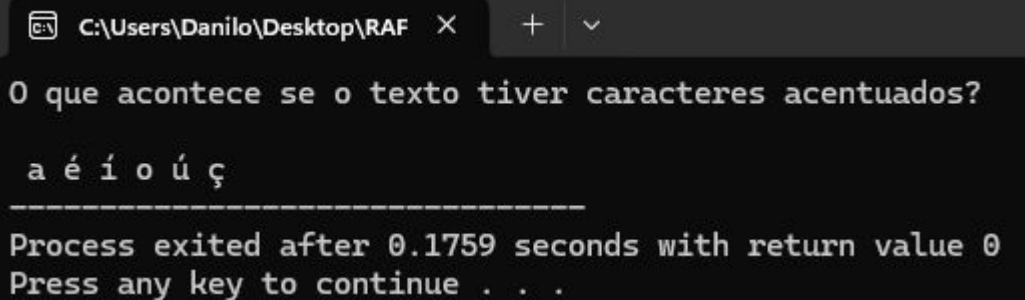
```
C:\Users\Danilo\Desktop\RAF >
O que acontece se o texto tiver caracteres acentuados?

 ã é í ã ú ç
-----
Process exited after 0.1687 seconds with return value 0
Press any key to continue . . .
```


printf

- É possível tratar a saída para mostrar a acentuação
 - Mas não vai funcionar para a entrada

```
1  #include <stdio.h>
2  #include <locale.h>
3
4  int main() {
5      setlocale(LC_ALL, "Portuguese");
6
7      printf("O que acontece se o texto tiver caracteres acentuados?\n\n");
8      printf("ã é í õ ú ç ");
9
10     return 0;
11 }
```



C:\Users\Danilo\Desktop\RAF x + v

O que acontece se o texto tiver caracteres acentuados?

ã é í õ ú ç

Process exited after 0.1759 seconds with return value 0

Press any key to continue . . .

Vamos combinar

- Vamos trabalhar sem utilizar acentuação
 - Nada de palavra com acentuação ou ç

Exercício 12

- Criar um código que a saída no console é:

```
*****  
* Raciocinio Algoritmico e Fundamentos da Computacao *  
*****  
* Nome: <coloque o seu nome> *  
* Hobby: <coloque o seu hobby> *  
* O Exercicio 12 eh o primeiro programa da disciplina *  
*****
```

- Obs.: Os asteriscos do lado direito tem que estar alinhados
- Enviar o exercício pelo Moodle
 - Conteúdo: arquivo .c

Sabemos escrever! Qual o próximo passo?

- Onde eu consigo armazenar/guardar uma informação?

Sabemos escrever! Qual o próximo passo?

- Onde eu consigo armazenar/guardar uma informação?
 - Em uma variável
 - A linguagem é fortemente tipada e então a variável é de algum tipo específico

Tipo de variável

- **int**
 - Números inteiros
 - 4 bytes = 32 bits = 2^{32} = 4294967296 números
 - 1 bit sinal (+, -) e 2^{31} de valor (-2147483648 até 2147483647)
- **float**
 - Números com casas decimais
 - 4 bytes, mas existe um padrão de representação e a conta não é igual ao int
- **double**
 - Números com casas decimais também, mas números maiores
 - 8 bytes
 - Não é o dobro do float $2^{32} \times 2 = 2^{33}$
 - 8 bytes = $2^{64} = 2^{32} \times 2^{32}$
- **char**
 - Caractere
 - 1 byte

Tipo de variável

```
int variavelInteira = 2147483647;
```

```
float variavelComCasaDecimal = 3.14; //ponto e não virgula
```

```
double variavelComCasaDecimalNumeroMaior = 3.14; //ponto e não virgula
```

```
char caractere = 'a'; //apóstrofo e não aspas simples
```

Operações matemáticas básicas

```
int numero1 = 3;
```

```
int numero2 = 2;
```

```
int soma = numero1 + numero2;
```

```
int subtracao = numero1 - numero2;
```

```
int multiplicacao = numero1 * numero2;
```

```
int divisao = numero1 / numero2;
```

```
int resto = numero1 % numero2;
```

- % (mod)
 - resto só existe para inteiro
 - float e double são números com casa decimal e não existe resto na divisão

Como imprimir o valor da variável?

```
1  #include <stdio.h>
2
3  int main() {
4
5      int variavelInteira = 2147483647;
6
7      float variavelComCasaDecimal = 3.14; //ponto e não virgula
8
9      double variavelComCasaDecimalNumeroMaior = 3.14; //ponto e não virgula
10
11     char caractere = 'a'; //apóstrofo e não aspas simples
12
13     printf("Variavel inteira = %d\n", variavelInteira);
14     printf("Variavel float = %f\n", variavelComCasaDecimal);
15     printf("Variavel double = %f\n", variavelComCasaDecimalNumeroMaior);
16     printf("Variavel char = %c\n", caractere);
17
18     printf("\n\n\n");
19
20     printf("No mesmo printf:\nint = %d\nfloat = %f\n", variavelInteira, variavelComCasaDecimal);
21
22     return 0;
23 }
```

Como imprimir o valor da variável?

- Utilizando o printf
 - %d
 - imprimir número inteiro
 - %f
 - imprimir número com casa decimal (número com ponto flutuante)
 - %c
 - imprimir caractere
- Para cada %d, %f, %c no texto deve existir 1 variável como parâmetro do printf

Exercício 13

- Criar um código
 - 1 variável com o seu ano de nascimento
 - 1 variável com o ano atual
 - 1 variável com o resultado da conta da sua idade no final do ano
 - Imprimir o texto, e substituir o que está entre <> por uma variável do seu código
 - No final de <ano atual> eu vou ter <idade> anos.
- Enviar o exercício pelo Moodle
 - Conteúdo: arquivo .c

Perguntas?

Obrigado!

Até a próxima aula!