

Aula Prática 6

Prazo de entrega: conferir no Moodle

Forma de Entrega: Enviar somente os arquivos `.c` e `.h` que você fez.

Exercício 1: Número de Euler

O valor de e (número de Euler) pode ser aproximado pelo somatório:

$$e = 1 + 1/1! + 1/2! + 1/3! + 1/4! + 1/5! \dots$$

Implemente um programa para aproximar o valor de e . Seu programa deve aumentar a precisão da aproximação calculando iterativamente (isto é, num laço) os termos do somatório acima. Seu programa deve parar a aproximação quando o i -ésimo termo ($1/i!$) for menor do que 10^{-6} .

Implemente esse programa fazendo uma função separada para calcular o fatorial de i .

Número de Euler na [Wikipedia](#).

Exercício 2: Calculador de conceito

O conceito de um aluno é calculado a partir de sua nota, seguindo a tabela abaixo. Faça um programa que lê a nota de um aluno do teclado e imprime seu conceito. Assuma que alunos só podem receber notas inteiras (isto é, sem decimais). Você deve usar o comando `switch` para verificar a nota lida.

Nota	Conceito
0 a 4	F
5	E
6	D
7	C
8	B
9+	A

Exercício 3: Triângulo de Floyd

Faça um programa que lê um número inteiro N e imprime as N primeiras linhas do triângulo de Floyd:

```
1
2 3
4 5 6
7 8 9 10
11 12 13 14 15
```

Desafio: Faça um programa para imprimir a N -ésima linha sem calcular as anteriores.

Exercício 4: Tamanho de uma string

Implemente um programa que leia um *string* e conte o número de caracteres. **Não** use a função `strlen`. Para ler uma *string* de até 127 caracteres do teclado, use o seguinte código:

```
char linha[128];
printf("digite uma linha:\n");
fgets(linha, 128, stdin);
```

Exercício 5: Conversão de caixa

Faça um programa que leia um *string* e modifique todos os caracteres minúsculos por caracteres maiúsculos.

Dica: os caracteres minúsculos tem o código ASCII entre 97 e 122 e os caracteres maiúsculos tem o código ASCII entre 65 e 90.

Exercício 6: Detector de palíndromos

Implemente código C para testar se uma *string* é um palíndromo. Um palíndromo é uma palavra idêntica quando lida de trás para frente, como "arara", "radar" e "reviver".

Exercício 7: Inversão de string

Escreva um programa para ler um texto do teclado e imprimir o inverso dele. Use o ponto final para indicar o término do texto, ou seja, o usuário deve terminar o texto sempre com um ponto final. Leia um caractere do texto por vez usando a função `getc` como a seguir:

```
char c = getc(stdin);
```

Exercício 8: Abreviação de nome

Escreva um programa para ler um nome completo do teclado terminado com um ponto final ('.'). Seu programa deve imprimir o mesmo na forma abreviada. Exemplo: o nome "Pedro Olmo Stancioli Vaz de Melo." deve ser abreviado para "P.O.S.V.M.". Note que o processo de abreviação deve ignorar palavras que começam com caracteres minúsculos. Considere que o usuário irá inserir apenas nomes válidos, sempre irá terminar o nome com o caractere '.' e o nome não conterá acentos.

Exercício 9: Desafio

Implementar um programa do **Exercício 8** considerando os possíveis erros do usuário:

- O número de espaços entre os nomes pode ser ilimitado. Ex: "Pedro Olmo".
- Considerar que o usuário pode trocar letras maiúsculas por minúsculas e vice-versa. Ex: "Pedro oLMo".
- Desconsiderar todas as palavras conectoras de nomes: de, da, do, das, dos.