

Algoritmos e Estruturas de Dados 1

Primeiro Trabalho Prático

Entrega Sugerida: 13/Set/2019

2º Semestre 2019 - DC-UFSCar

1 Introdução

No primeiro Trabalho Prático (TP01) vamos exercitar o uso de **recursão**. O programa entregue deve seguir os itens abaixo:

- O TP01 deverá ser feito individualmente e plágio não será tolerado;
- Cada função pedida deve seguir a assinatura indicada na sua descrição para fins de correção;
- Outras funções auxiliares podem ser criadas para facilitar o desenvolvimento da recursão, desde que as funções pedidas no enunciado estejam presentes e se comportem de acordo com o que foi pedido;
- O estudante deverá usar um arquivo **tp01_recursao.c** como base para o desenvolvimento. Este arquivo possui algumas funções implementadas, o esqueleto de outras que devem ser implementadas pelo estudante, bem como testes no corpo da função main.
- O TP01 deve ser entregue no run codes (<https://run.codes>) em um arquivo **tp01.c** com as funções pedidas e um cabeçalho com as informações do estudante (nome, curso, RA);
- Cada estudante deve se cadastrar no run codes (<https://run.codes>) informando Nome Completo, escolhendo “UFSCar - Universidade Federal de São Carlos” no campo Universidade e colocando seu RA no campo Núm. Matrícula. Depois de cadastrado, basta logar no run codes e se matricular na disciplina “1001502 - Algoritmos e Estruturas de Dados 1” usando o Código de Matrícula WAE8.
- Compile o seu TP usando o compilador GCC;
- Observação: Não iremos utilizar flags de compilação nesse TP.

2 Tarefas

2.1 Número Palíndromo

Um número é dito palíndromo se ele é igual quando lido nos dois sentidos (do início para o final e do final para o início). Alguns exemplos: 77, 505, 2112, 369963.

Desenvolva uma função **recursiva** que verifica se um número é palíndromo. No arquivo **tp01_recursao.c** existe uma função

```
void palindromo(char *s)
```

que chama a função recursiva

```
int palindromoR(int i, int f, char *s)
```

que recebe uma string s, com início na posição i e final na posição f, e devolve 1 se a string contém um palíndromo e 0 caso contrário.

É a função recursiva palindromoR que você deve implementar para que o programa funcione corretamente.

2.2 Inversão de palavras

Desenvolva uma função **recursiva** que imprime a string recebida no sentido inverso. No arquivo **tp01_recursao.c** existe uma função

```
void inversa(char *s)
```

que chama a função recursiva

```
void inversaR(char *str)
```

que recebe uma string s e imprime a inversa de s.

É a função recursiva inversaR que você deve implementar para que o programa funcione corretamente.

2.3 Números de Stirling

Números de Stirling sem sinal estão relacionados com os coeficientes $\begin{bmatrix} n \\ k \end{bmatrix}$ da seguinte expansão:

$$(x)_n = x(x+1)(x+2)\dots(x+n-1) = \sum_{k=0}^n \begin{bmatrix} n \\ k \end{bmatrix} x^k$$

Tais números pode ser computados pela seguinte relação de recorrência, válida para $k > 0$.

$$\begin{bmatrix} n+1 \\ k \end{bmatrix} = n \begin{bmatrix} n \\ k \end{bmatrix} + \begin{bmatrix} n \\ k-1 \end{bmatrix}$$

$$\text{com, } \begin{bmatrix} 0 \\ 0 \end{bmatrix} = 1 \text{ e } \begin{bmatrix} n \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ k \end{bmatrix} = 0$$

Dados n e k , não negativos. Desenvolva uma função **recursiva** que imprime o coeficiente $\begin{bmatrix} n \\ k \end{bmatrix}$. No arquivo **tp01_recursao.c** existe uma função

```
void stirling(unsigned long n, unsigned long k)
```

que chama a função recursiva

```
unsigned long stirlingR(unsigned long n, unsigned long k)
```

que recebe n e k e retorna o valor do coeficiente $\begin{bmatrix} n \\ k \end{bmatrix}$.

Você deve implementar a função `stirlingR` para que o programa funcione corretamente.

2.4 Imprimindo uma padrão

Desenvolva uma função **recursiva** que, dado n não negativo, imprime o seguinte padrão:

$$n(n-1)(n-2)\dots 210012\dots (n-2)(n-1)n$$

Por exemplo, se $n = 10$. Deve ser impresso o padrão:

```
109876543210012345678910
```

No arquivo **tp01_recursao.c** existe uma função

```
void padrao(unsigned int)
```

que chama a função recursiva

```
void padraoR(unsigned int)
```

Você deve implementar a função `padraoR` para que o programa funcione corretamente.