

Lista de Exercícios

Ponteiros (parte2)

[Ex01]. Escreva uma função que crie dinamicamente um vetor de inteiros. A função deve inicializar o vetor com um valor informado por parâmetro e devolver seu endereço.

```
int* criaVetor(int n, int valor)
```

[Ex02]. Escreva uma função que crie dinamicamente um vetor de inteiros contendo números aleatórios. A função deve preencher o vetor com números aleatórios entre 0 e (3 * n) e devolver seu endereço.

```
int* criaVetorAleatorio(int n)
```

[Ex03]. Escreva uma função que crie dinamicamente uma matriz de inteiros. A função deve inicializar a matriz com um valor informado por parâmetro e devolver seu endereço.

[Ex04]. Escreva uma função que faça uma cópia de string. A função deve criar dinamicamente a nova string ajustando o tamanho do vetor à quantidade necessária de caracteres e devolver seu endereço. Não use as funções da biblioteca string.h.

```
char* clone(char* s)
```

[Ex05]. Escreva uma função que extraia uma substring de uma string. A função deve criar dinamicamente o vetor que acomodará a substring e devolver seu endereço. O tamanho do vetor será definido pelo intervalo informado. A função deve validar o intervalo para certificar-se que ele existe, caso seja inválido a função deve devolver NULL.

```
char* sub1(char* s, int inicio, int fim)
```

[Ex06]. Escreva uma função que extraia uma substring de uma string. A função deve criar dinamicamente o vetor que acomodará a substring e devolver seu endereço. O tamanho do vetor será definido pela distância entre o índice informado e o fim da string. A substring inicia no índice informado e termina no último caractere.

O índice pode ser positivo ou negativo. Quando negativo, o índice deve ser considerado do fim para o início. Por exemplo, considerando uma string com N caracteres, o índice -1 é N-1, -2 é N-2, -3 é N-3 e assim por diante. A função deve validar as entradas e devolver NULL se não for possível extrair a substring.

```
char* sub2(char* s, int indice)
```

[Ex07]. Baseado no protótipo apresentado, escreva três versões da função. A primeira versão compara números inteiros, a segunda compara números reais e a terceira compara strings. As funções devem comparar dois elementos passados por parâmetro e devolver 1 se o primeiro elemento for maior que o segundo, -1 se for menor ou 0 se forem iguais.

```
int compara_int(void* a, void* b)  
int compara_double(void* a, void* b)  
int compara_string(void* a, void* b)
```

[Ex08]. Escreva uma função que intercale dois vetores já ordenados em um único vetor. A função deve MANTER a ordem crescente entre os elementos do vetor resultante.

```
int* intercala1(int* v1, int tamV1, int* v2, int tamV2)
```

Escreva a mesma função, porém, considerando um único vetor. A separação dos dois vetores é baseada em um índice. Por exemplo, da posição 0-3 é um vetor e da 4-9 outro vetor. As alterações são realizadas no próprio vetor passado por parâmetro.

```
void intercala2(int* vetor, int delimitador, int tamVetor)
```

[Ex09]. Escreva uma função recursiva para imprimir uma string.

```
void imprime_string(char* s)
```

[Ex10]. Escreva uma função recursiva que devolva o tamanho de uma string

```
int comprimento_string(char* s)
```

[Ex11]. Escreva uma função recursiva que faça cópia de uma string para outra.

```
void copia_string(char* destino, char* origem)
```

[Ex12]. Explique a diferença entre as duas funções apresentadas a seguir e se existe algum problema em alguma das versões.

a)

```
int* quadrado(int a){  
    int valor = a * a;  
    return &valor;  
}
```

b)

```
int* quadrado(int a){  
    int* valor = (int*)calloc(1, sizeof(int));  
    *valor = a * a;  
    return valor;  
}
```

Modelo de cabeçalho dos programas

```
/******  
Algoritmos e Estrutura de Dados 1  
Prof. Rafael Liberato  
-----  
Exercício: <Lista>.<Exercício>.<Item>  
Aluno: <Nome do aluno>  
Data: <data>  
*****/  

```

Exemplo:

```
/******  
Algoritmos e Estrutura de Dados 1  
Prof. Rafael Liberato  
-----  
Exercício: L01.Ex01.a  
Aluno: João da Silva  
Data: 10/12/12  
*****/  

```