

Prova 3
Algoritmos e Estruturas de Dados I
Professor: Pedro O.S. Vaz de Melo
07 de julho de 2017

Nome: _____
escrevendo o meu nome eu juro que seguirei o código de honra

Código de Honra para este exame:

- Não darei ajuda a outros colegas durante os exames, nem lhes pedirei ajuda;
- não copiarei nem deixarei que um colega copie de mim;
- não usarei no exame elementos de consulta não autorizados.

Referências:

Função/Operador	Descrição	Exemplo
<code>void* malloc (size_t size);</code>	aloca um bloco de memória de tamanho <code>size</code> , retornando um ponteiro para o início do bloco.	<code>int *p1 = (int*)malloc(sizeof(int));</code>
<code>FILE* fopen(const char *filename, const char *mode)</code>	abre o arquivo <code>filename</code> no modo <code>mode</code>	<code>FILE *temp = fopen("temp.txt", "w");</code>
<code>int fscanf(FILE *arq, const char *format, &variáveis);</code>	lê dados numéricos do arquivo <code>arq</code>	<code>fscanf(arq, "%f", &notal);</code>
<code>int fclose (FILE * arq)</code>	fecha o arquivo <code>arq</code>	<code>fclose(arq);</code>
<code>char* fgets (char *str, int num, FILE *arq)</code>	Lê uma linha do arquivo apontado por <code>arq</code> ou no máximo <code>num</code> caracteres	<code>fgets(buffer, 1000, arq);</code>
<code>char *strtok (char *str, const char *delimiters)</code>	Retorna um campo da <i>string</i> <code>str</code> separado por um dos caracteres contidos em <code>delimiters</code> . Se <code>str</code> é NULL, busca o campo da <i>string</i> usada na chamada anterior.	<code>char *nome = strtok(buffer, ",");</code>
<code>int atoi (const char *str);</code>	converte a <i>string</i> <code>str</code> para inteiro	<code>int num_cem = atoi("100");</code>

1. (5 points) Escreva uma função RECURSIVA de nome **somaNaturais** que retorna a soma de todos os números naturais de 1 até n . O procedimento não deve fazer uso de estruturas de repetição (while, for etc) nem de variáveis globais. O protótipo dessa função deve ser:

```
int somaNaturais(int n);
```

2. (5 points) Escreva uma função RECURSIVA que retorna 1 se um número n for uma potência de 2 ou 0 caso contrário. Um número n é uma potência de 2 se pode ser representado por 2^k , $k \geq 0$. Exemplo: **ehPotencia2(16)** retorna 1, pois $16 = 2 * 2 * 2 * 2 = 2^4$. O protótipo dessa função deve ser:

```
int ehPotencia2(int n);
```

3. (5 points) Escreva uma função RECURSIVA que recebe um apontador para uma *string* como parâmetro e retorna o número de espaços contidos na *string*. Ex: para a *string* `ola, tudo bem?`, a sua função deve retornar 2. A função deve ter o seguinte protótipo:

```
int numEspacos(char *str);
```

4. (6 points)

Escreva um procedimento que recebe um ponteiro para *string* como **parâmetro por referência** e dois índices inteiros *ini* e *fim*. O seu procedimento deve compactar (ou cortar) a *string* original, mantendo na memória somente os caracteres de *ini* a *fim* (inclusive eles). Dica: use alocação dinâmica de memória para armazenar uma área nova para a *string* compactada e depois manipule os ponteiros de forma que o ponteiro passado como parâmetro aponte, no final, para essa nova área. Não esqueça de desalocar memória e do \0.

5. (9 points) Os dados dos funcionários de uma empresa são armazenados em um arquivo de nome `func.dat`. Cada linha deste arquivo armazena o registro de um funcionário, que contém os seguintes campos: matrícula, nome e data de admissão. Um exemplo deste arquivo pode ser visto abaixo:

```
839|Carlos Alberto Silva|1/11/1987
263|Carlos Alberto Parreira|12/7/1990
13|Mario Jorge Lobo Zagallo|8/6/1996
666|Dunga|2/8/2006
```

Preencha o código abaixo, que descreve um programa que lê este arquivo e cria um outro, de nome `out.dat`, contendo em cada linha a matrícula do funcionário e o número de palavras que seu nome tem. Considere que no arquivo há apenas um espaço entre os nomes. Assim, para contar o número de palavras use a função `numEspacos`. Assuma também que as bibliotecas estão todas incluídas.

```
void main() {
    FILE *arq, *arqw;
    int matricula, numNomes;
    char buf[500], *nome;
    //abre arquivo para leitura:

    arq = _____;
    //abre arquivo para escrita:

    arqw = _____;
    while(!feof(arq)) {

        fgets(_____);

        matricula = _____;

        nome = _____;

        numNomes = _____;

        fprintf(_____);
    }

    _____;

    _____;
}
```

Exemplo de arquivo de saída:

```
839|3
263|3
13|4
666|1
```