

Prova 3

Algoritmos e Estruturas de Dados I - turma TM2

Professor: Pedro O.S. Vaz de Melo

26 de novembro de 2013 (valor: 30 pontos)

Nome: _____

escrevendo o meu nome eu juro que seguirei o código de honra

Código de Honra para este exame (baseado no *Honor Code* da Universidade de Stanford):

- 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.

Informações importantes:

- Considere que todos os procedimentos e funções pedidas nesta prova serão implementados no módulo `prova3.h`.
- Em questões que pede um **programa**, este deve ser completo, com bibliotecas (incluindo o módulo `prova3.h` quando necessário), função `main`, etc. Se deve ser feita uma **função**, somente a função é suficiente. Se deve ser feito um **procedimento**, somente o procedimento é suficiente.
- A interpretação das questões da prova faz parte do critério de avaliação. Caso tenha dúvida sobre a sua interpretação de uma determinada questão, escreva as suas suposições na resolução da mesma.

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 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 <code>NULL</code> , busca o campo da <i>string</i> usada na chamada anterior.	<code>char *nome = strtok(buffer, ",");</code>

1. (8 points) Escreva uma função RECURSIVA que receba por parâmetro dois valores inteiros x e y e retorne o resultado de x^y . Sua função não pode usar *loops* (`for`, `while`, etc) nem a função `pow`. A função deve ter o seguinte protótipo:

```
int potencia(int x, int y);
```

2. (2 points) Escreva uma função de nome `tamString` que recebe uma *string* como parâmetro e retorna o seu tamanho. Lembre que o último caractere de uma *string* é sempre `'\0'`. A função deve ter o seguinte protótipo:

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

3. (10 points) A função `strcat` da biblioteca `string.h` recebe duas *strings* como parâmetros, `str1` e `str2`, e concatena a *string* `str2` à *string* `str1`. No entanto, se o espaço alocado à *string* `str1` (ex: `char str1[10] = "mas que"`) não for suficiente para receber os caracteres de `str2` (ex: `char str2[10] = "loucura!"`), a função falha (ex: falhou porque `str1` só tem 10 posições alocadas, não suficientes para receber todos os caracteres de `str2`). Assim, escreva uma função de nome `concatena` que retorna a concatenação de duas *strings* e que garante que a concatenação ocorrerá. Em suma, essa função recebe dois ponteiros para `char` como parâmetros, cada qual contendo uma *string* completa, e retorna outro, contendo a *string* resultante da concatenação de `str1` e `str2`. Exemplo: se `str1` for "mas que" e `str2` for " loucura!", a função deve retornar "mas que loucura!". A função deve ter o seguinte protótipo:

```
char* concatena(char *str1, char *str2);
```

4. (10 points) Um professor deseja comparar quais foram os seus melhores alunos em um determinado semestre, os cruzeirenses ou os atleticanos. Para isso, pediu para você implementar um programa que leia o seu arquivo "notas.txt" contendo as informações dos alunos e compare a média das notas dos cruzeirenses com a dos atleticanos. O seu programa deve informar quais torcedores obtiveram a maior média (ex: maior média: cruzeirenses). O formato do arquivo é o seguinte:

```
nome|time (C = cruzeirense, A=atleticano, O=outro)|nota (0 a 100)
```

Exemplo de arquivo:

```
Tyrion Lannister|C|98
Davos Seaworth|A|75
Theon Greyjoy|A|82
Sandor Clegane|O|59
Khal Drogo|C|65
```