

Prova 3

Algoritmos e Estruturas de Dados I

Professor: Pedro O.S. Vaz de Melo

25 de novembro de 2014 (valor: 30 pontos)

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.

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.
- Vocês podem utilizar qualquer função pedida na prova em suas questões. Considere que a implementação da função que você está usando está correta.

Referências:

Função/Operador	Descrição	Exemplo
FILE* fopen(const char *filename, const char *mode)	abre o arquivo filename no modo mode	FILE *temp = fopen("temp.txt", "w");
int fclose (FILE * arq)	fecha o arquivo arq	fclose(arq);
int feof (FILE * arq)	verificar se o arquivo arq chegou ao fim	int fim_arq = feof(arq);
void* malloc (size_t size);	aloca um bloco de memória de tamanho size, retornando um ponteiro para o início do bloco.	int *p1 = (int*)malloc(sizeof(int));
char *strtok (char *str, const char *delimiters)	Retorna um campo da <i>string</i> str separado por um dos caracteres contidos em delimiters. Se str é NULL, busca o campo da string usada na chamada anterior.	char *nome = strtok(buffer, ",");
void free (void *p);	Desaloca o bloco de memória apontado por p.	free(p);

1. (6 points) Escreva uma função RECURSIVA que recebe uma *string* como parâmetro e a imprime invertido. Exemplo: se a função receber a palavra “bola”, deve imprimir “alob”. A sua função deve ter o seguinte protótipo:

```
void strInvertida(char str[]);
```

2. (6 points) A função `hotpo(n)` (*Half Or Triple Plus One*) consiste no seguinte: pegue o número n e o divida por 2 se ele for par, ou multiplique por 3 e some 1 caso n seja ímpar. Repita o processo até que n seja 1. A conjectura de Collatz estipula que esse processo sempre converge para 1. Escreva uma função RECURSIVA que retorna o número de elementos gerados pela função `hotpo` até que n se torne 1. Imprima cada um dos elementos gerados. Exemplo: `hotpo(5)` retorna 6 e imprime a sequência 5, 16, 8, 4, 2, 1. A sua função deve ter o seguinte protótipo:

```
int hotpo(int n);
```

3. (8 points) Escreva uma função de nome `substring` que recebe como parâmetros uma *string* `str` por referência, um inteiro `ini` e um inteiro `fim`. A sua função deve remover de `str` todos os caracteres que estão em posições inferiores a `ini` e superiores a `fim`. Em suma, você deve transferir a *string* `str` somente com os caracteres entre `ini` e `fim` para uma nova área de memória (*heap*) e, após isso, fazer com que o parâmetro `str` aponte para essa nova área. Exemplo: se `str` for "abracadabra", `ini` for 2 e `fim` for 5, você deve transformar a *string* `str` em "raca". Não se esqueça de desalocar o espaço da *string* antiga e usar o terminador `\0`. A função deve ter o seguinte protótipo:

```
void substring(char **str, int ini, int fim);
```

4. (10 points) Escreva uma função que recebe o nome de um arquivo texto como parâmetro e retorna o número de palavras contidas nele. Considere que o único conector de palavras é o hífen, ou seja, o caractere "-". A sua função deve ter o seguinte protótipo:

```
int contaPalavras(char nomeArquivo[]);
```

Para o exemplo de arquivo abaixo, o seu programa deve retornar 26.

```
Afoot and light-hearted I take to the open road,  
Healthy, free, the world before me,  
The long brown path before me leading wherever I choose.
```