Prova 3

Algoritmos e Estruturas de Dados I

Professor: Pedro O.S. Vaz de Melo

30 de junho de 2016

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
`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));`
`FILE* fopen(const char *filename, const char *mode)`	abre o arquivo `filename` no modo `mode`	`FILE *temp = fopen("temp.txt", "w");`
`int fscanf(FILE *arq, const char *format, &variáveis);`	lê dados númericos do arquivo `arq`	`fscanf(arq, "%f", &nota1);`
`int fclose ( FILE * arq )`	fecha o arquivo `arq`	`fclose(arq);`
`char* fgets (char *str, int num, FILE *arq)`	Lê uma linha do arquivo apontado por `arq` ou no máximo `num` caracteres	`fgets(buffer, 1000, arq);`
`char *strtok (char *str, const char *delimiters)`	Retorna um campo da *string* `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, ",”);
`int atoi (const char *str);`	converte a *string* `str` para inteiro	`int num_cem = atoi("100");`

1. (*5 points*) Escreva um procedimento RECURSIVA que imprime uma linha com n asteriscos e uma quebra de linha no final (`\n`). A função deve ter o seguinte protótipo:

```
void impAst(int n);
```

2. (*5 points*)
Escreva um procedimento RECURSIVO de protótipo `void impTri(int i, int n)` que imprime um triângulo lateral na tela formado por asteriscos (com cortes nos lados, se for o caso). O triângulo deve começar imprimindo i asteriscos na primeira linha, $i+1$ na segunda, até n asteriscos na $(n-i+1)$ésima linha. Depois deve imprimir $n-1$ asteriscos, $n-2$, até i asteriscos na última linha. Dica: use o procedimento `impAst`. Exemplo: impTri(3,5) imprime

```
***
****
*****
****
***
```

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. (*8 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
```