Prova 3

Algoritmos e Estruturas de Dados I

Professor: Pedro O.S. Vaz de Melo

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. (*2 points*) Escreva uma função RECURSIVA que recebe um ponteiro para uma *string* como parâmetro e retorna o seu tamanho. Sua função não pode usar *loops* (`for, while, etc`) e deve ter o seguinte protótipo:

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

2. (*5 points*) Escreva uma função RECURSIVA que recebe um inteiro n por parâmetro e retorna a soma dos seus dígitos. Exemplo: se a função receber 54321 como parâmetro, ela deve retornar 15, que é o resultado da soma $5+4+3+2+1$. Sua função não pode usar *loops* (`for, while, etc`) e deve ter o seguinte protótipo:

```
int somaDigitos(int n);
```

3. (*5 points*) Escreva uma função RECURSIVA de nome `geraPalindromo` que recebe uma *string* `str` como parâmetro e imprime o palíndromo resultante da concatenação de `str` com o seu inverso. Exemplo: se a *string* recebida como parâmetro for `mus`, a sua função deve imprimir `mussum`. Sua função não pode usar *loops* (`for, while, etc`) e deve ter o seguinte protótipo:

```
int geraPalindromo(char str[]);
```

4. (*8 points*) Escreva uma função que recebe dois ponteiros para char `str1` e `str2` como parâmetros, cada qual contendo uma *string* completa, e retorna **outro**, diferente dos ponteiros de entrada, contendo a *string* resultante da concatenação das *strings* apontadas por `str1` e `str2`. Exemplo: se `str1` for `"mas que"` e `str2` for `" loucura!"`, a função deve retornar um ponteiro para a *string* `"mas que loucura!"`. Dica: você pode usar a função `tamString`. A função deve ter o seguinte protótipo:

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

5. (*10 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 CPF (sem o dígito verificador). Um exemplo deste arquivo pode ser visto abaixo:

```
839|Rorschach|570303063
263|Doctor Manhattan|290816052
13|Ozymandias|610342411
```

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 dos funcionários que estão com o CPF incorreto. Considere que um CPF é incorreto se a soma dos seus dígitos não for divisível por 9. Além disso, assuma que um tipo de dado `int` é capaz de armazenar corretamente qualquer CPF do arquivo. Para somar os dígitos, use a função `numEspacos`. Assuma também que as bibliotecas estão todas incluídas.

```
void main() {
   FILE *arq, *arqw;
   int matricula, CPF;
   char buf[1000], *nome;
   //abre arquivo para leitura:

   arq = ________________________________________________;
   //abre arquivo para escrita:

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

      fgets(_____________________________________________);

      matricula = ________________________________________;

      nome = _____________________________________________;

      CPF = ______________________________________________;

      if(_____________________________________________________________)

            fprintf(_____________________________________________);
   }

   ____________________________________________;

   ____________________________________________;
}
```

Para o arquivo acima, o arquivo de saída deverá conter:

```
2
666
```