

INF 1007 – EXERCICIO PARA P1 – 23/04/2013	
Nome:	
Matrícula:	Turma:

Instruções: Leia as instruções até o final antes de realizar qualquer ação.

ATENÇÃO: Este exercício NÃO É UM “SIMULADO”, é apenas para o aluno ter alguma noção da estrutura da prova. O número e o tipo de questões não são necessariamente os mesmos.

1. Esta prova deverá ser resolvida em até 1:15 minutos
2. A prova é sem consulta e sem perguntas. A interpretação do enunciado faz parte da prova.
3. Seguindo as instruções do fiscal, o aluno deve **CRIAR** o arquivo **TURMA_MATRICULA_NOME_EXERCICIO.c** (onde deverá ser colocada a solução da prova) substituindo *TURMA* pela turma do aluno (com 3 dígitos) , *MATRICULA* pela matrícula do aluno (com 7 dígitos) e *NOME* pelo último nome do aluno. Por exemplo, “33B_0720870_MARTINS_EXERCICIO.c” seria o arquivo contendo a solução da prova de um aluno da turma 33B, cuja matrícula é 0720870 , chamado, por exemplo Pedro da Silva Martins. Deve ser incluído no início desse arquivo, como comentário, o cabeçalho abaixo preenchido CORRETAMENTE com nome completo, matrícula e turma do aluno. Informações incorretas ou omitidas neste cabeçalho invalidam a prova.

/*

NOME:

MATRICULA:

TURMA:

*/

Antes da solução de cada questão deve vir a identificação da questão, também como comentário:

Exemplo: /* Solucao da Questao 1.a */

4. Por segurança, este arquivo .c deve ser salvo a cada 15 minutos, pelo próprio aluno. Somente esse arquivo será considerado a resposta da prova. EVITAR MANTER MAIS DE UMA CÓPIA, PARA NÃO ALTERAR UM ARQUIVO E ENTREGAR O ERRADO.
5. Esta é uma prova prática onde o aluno é suposto desenvolver as funções pedidas de maneira que compilem e funcionem corretamente conforme as especificações. Portanto, recomendamos que o aluno crie o arquivo .c em um projeto no MS Visual Studio. Mas apenas as funções pedidas serão consideradas e o único arquivo considerado resposta é o arquivo .c do item 3. (Não entregar projeto ou mais de um arquivo em nenhuma hipótese).

IMPORTANTE:

As funções que você escrever irão compor um módulo .c que será colocado no projeto do professor para fins de correção. Portanto, respeite a ordem dos argumentos da função conforme o enunciado. Por exemplo, o texto “a função *f* recebe um vetor de inteiros e o tamanho desse vetor e retorna a soma dos elementos desse vetor” implica no seguinte protótipo **int f(int * v,int n);** e não **int f(int n,int * v);**. Em outras palavras, você tem a liberdade de nomear suas variáveis, mas não pode mudar a ordem com que elas aparecem como argumentos. Isto é necessário para que o seu módulo possa ser usado pelos módulos de teste do professor.

6. Se em alguma questão o aluno criar e usar uma função auxiliar, essa função deve também ser apresentada no arquivo de respostas.

Questão 1:

No Departamento de Transporte, para implementar o cadastro dos carros existe um vetor do tipo estruturado Carro definido como se segue:.

```
struct carro
{
    int registro;
    char placa[8];
    char dono[51];
    int ano;
    int numMultas;
};
```

```
typedef struct carro Carro;
```

1) Escreva a função *exibeVetorDeCarros* , que recebe um vetor de carros, o número de carros e exibe os dados de todos os carros do vetor.

2) Escreva a função *obtemRegistrosDeUmAno* que recebe os seguintes dados através de seus argumentos de entrada:

- um vetor (de Carro) com carros da concessionária,
- o número de carros,
- um ano , e
- o endereço de uma variável inteira onde a função deverá disponibilizar o número de carros desse ano;

e retorna um novo vetor de inteiros, ou seja, o endereço do primeiro elemento de um novo vetor, com os registros apenas dos carros daquele ano. Esse novo vetor deve ser alocado dinamicamente com o tamanho exato necessário e deverá conter os registros desejados, na mesma ordem em que eles aparecem no vetor original. A função deve também disponibilizar o número de carros desse ano na variável inteira, cujo endereço foi fornecido como um dos argumentos na chamada da função. **Caso não existam carros desse ano ou caso não seja possível criar o novo vetor, a função deve retornar NULL.**

OBS: No caso de você querer testar sua função aqui vai um exemplo de como criar facilmente e inicializar um vetor de Carro a ser usado no seu teste

```
Carro vcarro[6] = { {4444, "LAN4532", "ana", 2008, 2},
                  {1111, "KCD1531", "leo", 2012, 6},
                  {6666, "LOB4172", "rui", 2010, 3},
                  {8888, "JKD5539", "bia", 2002, 2},
                  {3333, "LDV4511", "edu", 2008, 8},
                  {2222, "KWX2382", "lia", 2012, 7} };
```

INF1007- DI- EXERCICIO PARA P1 – 23/04/2013

Questão 2:

2.A) Escreva uma função que leia do teclado uma cadeia e crie com alocação dinâmica uma nova cadeia, do tamanho exato necessário, com cópia da cadeia lida. A função retorna a nova cadeia (ou seja o endereço do primeiro caractere da nova cadeia). Caso não seja possível criar a nova cadeia, a função retorna NULL.

2.B) Escreva agora a função `mostraMultiplosDe3`, que recebe um vetor de ponteiros para cadeias que representam números, o número de cadeias e exibe os números que são múltiplos de 3. Por exemplo, uma cadeia válida seria a cadeia de 6 caracteres “456120” correspondente ao número inteiro 456120. Para isso, cada cadeia terá que ser primeiro convertida no número inteiro que representa. Isso é feito utilizando-se a função `atoi()` do C, da biblioteca `stdlib.h`, que recebe uma cadeia que representa um número inteiro e retorna o valor inteiro a que corresponde essa cadeia. Por exemplo:

EX1: `char cad1[10] = "3451"; int num1 = atoi(cad1); => num1 tem o valor inteiro 3451`

EX2: `num2 = atoi("13008"); => num2 tem o valor inteiro 13008`

Obs: se desejar testar suas funções você pode, por exemplo, criar (ou incluir em uma já existente) uma main em que seja criado um vetor de 10 ponteiros para cadeias. Em seguida, preencha o vetor com 10 novas cadeias, onde cada cadeia corresponde a um número inteiro, ou seja, à representação de um número inteiro como uma cadeia de dígitos de ‘0’ a ‘9’, sem nenhum outro caractere. Por exemplo, uma cadeia válida seria a cadeia de 6 caracteres “256122” correspondente ao número inteiro 256122. Para ler e criar cada cadeia, pode ser utilizada a função do item 2.A. Assuma que cada cadeia lida é sempre válida, ou seja, não contém qualquer outro caractere que não esteja entre ‘0’ a ‘9’. Chame a função do item 2.B. E não esqueça de que, no caso de cadeias criadas com alocação dinâmica, elas devem ser liberadas quando não forem mais utilizadas.

Protótipos de funções que podem ser úteis nesta prova:

stdio.h:

```
int scanf (char* formato, ...);
int printf (char* formato, ...);
FILE* fopen (char* nome, char* modo);
int fclose (FILE* fp);
int fscanf (FILE* fp, char* formato, ...);
int fprintf (FILE* fp, char* formato, ...);
char* fgets(char* str, int size, FILE* fp);
int sscanf(char* str, char* formato, ...);
```

stdlib.h:

```
void* malloc (int nbytes);
void* realloc (void *p, int nbytes);
void free (void* p);
int atoi(char *s);
```

math.h:

```
double sqrt (double x);
double pow (double x, double exp);
double cos (double radianos);
double sin (double radianos);
```

string.h:

```
int strlen (char* s);
int strcmp (char* s, char *t);
char* strcpy (char* destino, char* fonte);
char* strcat (char* destino, char* fonte);
```

ctype.h:

```
int isdigit ( int c );
int toupper ( int c );
int tolower ( int c );
```

Não separe as folhas deste caderno.