

INF 1620 – P2 - 14/10/05	Questão 1
Nome:	
Matrícula:	Turma

Considere um cadastro de produtos de um estoque, com as seguintes informações para cada produto:

- Código de identificação do produto: representado por um valor inteiro
 - Nome do produto: com até 50 caracteres
 - Quantidade disponível no estoque: representado por um número inteiro
 - Preço de venda: representado por um valor real
- (a) Defina uma estrutura em C, denominada `produto`, que tenha os campos apropriados para guardar as informações de um produto, conforme descrito acima.
- (b) Escreva uma função que receba os dados de um produto (código, nome, quantidade e preço) e retorne o endereço de um `struct produto` criado dinamicamente e inicializado com os valores recebidos como parâmetros pela função. Essa função deve ter o seguinte protótipo:

```
struct produto* cria (int cod, char* nome, int quant, float preco);
```

INF 1620 – P2 - 14/10/05	Questão 2
Nome:	
Matrícula:	Turma

Considere um tipo que representa um funcionário de uma empresa, definido pela estrutura a seguir:

```
struct funcionario {
    char nome[81]; /* nome do funcionário */
    float valor_hora; /* valor da hora de trabalho em Reais */
    int horas_mes; /* horas trabalhadas em um mês */
};
typedef struct funcionario Funcionario;
```

Escreva uma função em C que preencha um vetor de ponteiros para `Funcionario` com os dados lidos de um arquivo texto. Essa função deve receber como parâmetros o vetor de ponteiros para `Funcionario` (representado pelo seu comprimento e pelo endereço de seu primeiro elemento) e o nome do arquivo de entrada. Nesse arquivo de entrada, os dados de cada funcionário são armazenados em duas linhas: uma com o seu nome (cadeia com até 80 caracteres), e outra com o valor de sua hora de trabalho e com o número de horas trabalhadas em um mês (nessa ordem). Um exemplo desse formato é mostrado abaixo.

João da Silva 15.0 160 Manuel Santos 15.0 80 Fulana de Tal 23.5 40

A função implementada deve ter o seguinte protótipo:

```
void carrega (int n, Funcionario** vet, char* arquivo);
```

Considere que: não existem linhas em branco no arquivo; o parâmetro `vet` já vem com todas as suas posições inicializadas com `NULL`; o comprimento do vetor (parâmetro `n`) é no mínimo igual à quantidade de registros de funcionários no arquivo de entrada; se não for possível abrir o arquivo, a função deve imprimir a mensagem “ERRO” e terminar a execução do programa.

INF 1620 – P2 - 14/10/05	Questão 3
Nome:	
Matrícula:	Turma

Considerando o tipo `Funcionario` definido na questão anterior, escreva uma função que imprima os nomes dos 5 funcionários com maiores salários, onde o salário de um funcionário é igual ao número de horas trabalhadas no mês vezes o valor de sua hora de trabalho. Assuma que não existem dois funcionários com o mesmo salário. A função deve receber como parâmetros o número total de funcionários e um vetor que armazena ponteiros para estruturas do tipo `Funcionario`, de acordo com o protótipo definido a seguir:

```
void imprime_marajas (int n, Funcionario** vet);
```

Dica: sua função pode primeiro ordenar o vetor em ordem decrescente de salários, usando alguma das técnicas de ordenação vistas no curso.

INF 1620 – P2 - 14/10/05	Questão 4
Nome:	
Matrícula:	Turma

Considerando novamente o tipo `Funcionario` definido na questão 2, escreva uma função que faça uma *busca binária* em um vetor de ponteiros para o tipo `Funcionario`, cujos elementos estão em ordem alfabética dos nomes dos funcionários. Essa função deve receber como parâmetros o número de funcionários, o vetor e o nome do funcionário que se deseja buscar, e deve ter como valor de retorno um ponteiro para o registro do funcionário procurado. Se não houver um funcionário com o nome procurado, a função deve retornar `NULL`. Sua função **não** deve utilizar a função `bsearch` da biblioteca padrão da linguagem C, e deve ter o seguinte cabeçalho:

```
Funcionario* busca (int n, Funcionario** v, char* nome);
```

INF 1620 – P2 - 14/10/05	Questão 5
Nome:	
Matrícula:	Turma

Considerando novamente o tipo `Funcionario` definido na questão 2, escreva uma função que salve em um arquivo texto o nome e salário de cada funcionário de um cadastro. Um exemplo de arquivo gerado por essa função é mostrado abaixo.

João da Silva	2400.00
Manuel Santos	1200.00
Fulana de Tal	940.00

Para isso, a função deve receber como parâmetros o número de funcionários, o cadastro de funcionários na forma de um vetor de ponteiros para o tipo `Funcionario`, e o nome do arquivo onde a função deve escrever. Se ocorrer erro na abertura do arquivo, a função deve retornar 0, caso contrário a função deve salvar o conteúdo do vetor segundo o formato acima e retornar 1. A função implementada deve ter o seguinte protótipo:

```
int salva (int n, Funcionario** vet, char* arquivo);
```

RASCUNHO

Respostas nesta folha não serão consideradas.

Protótipos de funções que podem ser úteis:

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, ...);
```

math.h:

```
double sqrt (double x);
double pow (double x, double exp);
```

string.h:

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

stdlib.h:

```
void* malloc (int nbytes);
void free (void* p);
void qsort (void *vet, int n, int tam, int (*comp) (const void*, const void*));
```

Não separe as folhas deste caderno.