

| | |
|--------------------------|-----------|
| INF 1620 – P4 - 27/06/02 | Questão 1 |
| Nome: | |
| Matrícula: | Turma |

Considere o tipo que representa uma grandeza vetorial (com três componentes) dado por:

```
struct vetor {
    float x;
    float y;
    float z;
};
typedef vetor Vetor;
```

Pede-se:

- (a) Escreva uma função, que dados dois vetores, calcule e retorne o valor do produto escalar desses vetores. Sabe-se que o produto escalar de dois vetores a e b é dado por: $a_x b_x + a_y b_y + a_z b_z$. O protótipo dessa função deve ser:

```
float prodesc (Vetor* a, Vetor* b);
```

- (b) Escreva um programa completo que capture os valores das componentes de dois vetores, entrados pelo usuário via teclado, e imprima o valor do produto escalar desses vetores usando a função do item anterior.

| | |
|--------------------------|-----------|
| INF 1620 – P4 - 27/06/02 | Questão 2 |
| Nome: | |
| Matrícula: | Turma |

- (c) Escreva uma função em C para concatenar duas cadeias de caracteres. A função deve receber duas cadeias de caracteres como parâmetros de entrada e ter como valor de retorno uma nova cadeia, cujo espaço de memória deve ser alocado pela função, contendo a concatenação da primeira com a segunda. Por exemplo, se forem passadas as cadeias “PUC” e “Rio”, a função deve retornar a cadeia “PUCRio”. A assinatura da função deve ser:

```
char* concatena (char* a, char* b);
```

Protótipos de funções que podem ser úteis:

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

Não separe as folhas deste caderno. Todas as folhas devem ter seu nome. Responda cada questão na folha correspondente. Use o verso se necessário.

| | |
|--------------------------|-----------|
| INF 1620 – P4 - 27/06/02 | Questão 3 |
| Nome: | |
| Matrícula: | Turma |

Considere uma lista encadeada que representa um conjunto de alunos. O tipo que representa cada nó da lista é dado por:

```
struct lista {
    int mat;           /* número de matrícula */
    char nome[81];    /* nome do aluno */
    struct lista* prox; /* ponteiro para próximo elemento */
};
typedef struct lista Lista;
```

Assumindo que os elementos da lista estejam ordenados (ordem crescente) pelo número de matrícula, escreva uma função que insira um novo aluno na lista. Esse novo aluno deve ser inserido de forma que a ordenação da lista seja mantida. A função deve receber como parâmetro o ponteiro para o primeiro nó da lista e os dados do novo aluno (número de matrícula e nome), e deve ter como valor de retorno o ponteiro para o primeiro nó da lista após a inserção. O protótipo dessa função é dado por:

```
Lista* insere (Lista* l, int m, char* n);
```

| | |
|--------------------------|-----------|
| INF 1620 – P4 - 27/06/02 | Questão 4 |
| Nome: | |
| Matrícula: | Turma |

Considere uma árvore binária que armazena valores inteiros. O tipo que representa um nó da árvore é dado por:

```
struct arv {
    int val;
    struct arv* esq;
    struct arv* dir;
};
typedef struct arv Arv;
```

Pede-se

- (a) Escreva uma função que, dado o ponteiro para o nó raiz, calcule e retorne a altura da árvore. O protótipo dessa função deve ser:

```
int altura (Arv* a);
```

- (b) Considerando que essa árvore representa uma *árvore binária de busca*, onde os valores associados aos nós da sub-árvore à esquerda são menores que o valor associado à raiz e que os valores dos nós da sub-árvore à direita são maiores, escreva uma função para imprimir os valores associados aos nós da árvore em ordem *decrecente*. O protótipo dessa função é dado por:

```
void imprime (Arv* a);
```

| | |
|--------------------------|-----------|
| INF 1620 – P4 - 27/06/02 | Questão 5 |
| Nome: | |
| Matrícula: | Turma |

Considere a representação de matrizes quadradas de números reais de dimensão $N \times N$. O tipo que representa a matriz é definido a seguir:

```
#define N 4
typedef double MAT[N][N];
```

Pede-se

- (a) Escreva uma função para salvar os elementos da matriz num arquivo cujo nome é passado como parâmetro. Os elementos da primeira linha da matriz devem ser salvos na primeira linha do arquivo (separados por brancos), os elementos da segunda linha da matriz na segunda linha do arquivo, e assim por diante. Se os elementos forem salvos com sucesso, a função deve retornar o valor 1; caso contrário, se o arquivo de saída não puder ser aberto, a função deve retornar o valor 0. O protótipo da função é dado por:

```
int salva(char *nomearq, MAT m);
```

- (b) Escreva uma função para ler, de um arquivo, os elementos de uma matriz supostamente salvos pela função do item anterior. O nome do arquivo é passado como parâmetro. Se a leitura dos elementos for bem sucedida, a função deve retornar o valor 1; caso contrário, se esse arquivo não puder ser aberto para leitura, a função deve retornar o valor 0. O protótipo da função é dado por:

```
int carrega(char *nomearq, MAT m);
```

Protótipos de funções que podem ser úteis:

```
FILE* fopen (char* nome, char* modo);
int fclose (FILE* fp);
int fscanf (FILE* fp, char* formato, ...);
int fprintf (FILE* fp, char* formato, ...);
```

Não separe as folhas deste caderno. Todas as folhas devem ter seu nome. Responda cada questão na folha correspondente. Use o verso se necessário.

RASCUNHO

Respostas nesta folha não serão consideradas.

Não separe as folhas deste caderno.