

Para todas as funções pedidas abaixo, não se esqueça de acrescentar um teste numa função *main*.

1) Escreva uma função para eliminar todas as ocorrências de um caractere numa string, retornando uma nova string, alocada dinamicamente, com a mesma sequência de caracteres da string original, a menos das ocorrências do caractere especificado. Por exemplo, se for passado a string “PUC-Rio” e o caractere ‘-’, deve-se retornar a string “PUCRio”. Note que o caractere pode aparecer múltiplas vezes na string original.

```
char* elimina_ocorrencias (char* s, char c);
```

2) Escreva uma função **recursiva** para calcular o valor máximo de um vetor de valores reais. O protótipo da função deve ser:

```
float maximo (int n, float* v);
```

Que alterações seriam necessárias se fosse para retornar o índice (e não o valor) do maior elemento?

3) Considere os tipos definidos abaixo para representar um cadastro de alunos de uma turma:

```
#define MAX 100
struct aluno {
    int mat;          /* matricula */
    char nome[81];   /* nome */
    float nota;      /* nota final */
};
typedef struct aluno Aluno;

struct turma {
    int n;           /* número de alunos efetivamente alocados */
    Aluno* vet[MAX]; /* vetor de ponteiros para alunos */
};
typedef struct turma Turma;
```

(a) Implemente uma função para criar uma nova turma, inicialmente vazia, alocada dinamicamente:

```
Turma* cria_turma (void);
```

(b) Implemente uma função para adicionar um novo aluno na turma, dadas suas informações:

```
void adiciona_aluno (Turma* t, int mat, char* nome, float nota);
```

(c) Implemente uma função para calcular a média das notas dos alunos de uma turma:

```
float notamedia (Turma* t);
```

(c) Implemente uma função para liberar a memória alocada para uma turma, e seus respectivos alunos:

```
float libera_turma (Turma* t);
```