

1) a) Escreva uma função com a seguinte interface:

```
void maiuscula (char cadeia[])
```

que receba uma cadeia de caracteres (terminada com um caractere nulo) contendo caracteres arbitrários e substitua os caracteres que são letras minúsculas nessa cadeia por letras maiúsculas. Use **cadeia** apenas como vetor, juntamente com os índices necessários.

b) Escreva uma função com a seguinte interface:

```
void maiuscula (char *cadeia)
```

que receba uma cadeia de caracteres (terminada com um caractere nulo) contendo caracteres arbitrários e substitua os caracteres que são letras minúsculas nessa cadeia por letras maiúsculas. Use apenas ponteiros e aritmética com ponteiros.

2) Escreva um programa com nome soma.cpp que soma todos os argumentos informados na linha de comandos, considerando que todos eles são números inteiros. Por exemplo, executando o programa da seguinte forma:

```
prompt$ ./soma 81 25 2
```

deve produzir a seguinte saída: 108

3) Dizemos que uma sequência de  $n$  elementos, com  $n$  par, é balanceada se as seguintes somas são todas iguais:

*a soma do maior elemento com o menor elemento;*

*a soma do segundo maior elemento com o segundo menor elemento;*

*a soma do terceiro maior elemento com o terceiro menor elemento;*

*e assim por diante . . .*

Exemplo: 2 12 3 6 16 15 é uma sequência balanceada, pois  $16 + 2 = 15 + 3 = 12 + 6$ .

Dados  $n$  ( $n$  par e  $n > 0$ ) e uma sequência de  $n$  números inteiros, verificar se essa sequência é balanceada. Use alocação **dinâmica** de memória.

Se achar necessário, suponha a existência de uma função de ordenação crescente com o protótipo a seguir e use-a no seu código.

```
void bubbleSort(int n, int *vetor)
```

4)

c) Na opção 3 o usuário deve digitar o número de um aluno e o programa deve mostrar o nome dele e a média aritmética das notas desse aluno. As mensagens disponíveis nesta opção são: Aluno não cadastrado. Aluno sem notas e Média do aluno = valor calculado.

```
int temCadastro(int n, tipoAluno *vetorA);
```

```
void listaMedia(int nAl, tipoAluno *vetorA, int mat, int numNotas, tipoNota *vetorN)
```

```
float listaMedia(int nAl, tipoAluno *vetorA, int mat, int numNotas, tipoNota *vetorN)
```

d) Na opção 4 os nomes dos alunos que não possuem notas devem ser listados. As mensagens disponíveis nesta opção são: A listagem dos nomes e Todos os alunos possuem notas.

```
int temNota(int mat, tipoNota *vetorNotas);
```

```
void listaAlunosNota(int nAl, tipoAluno *vetorA, int numNotas, tipoNota *vetorN);
```

