

## 2ª Lista de Exercícios

### Linguagem de Programação I

Turma	Data de Entrega
LP- Noturno	21/10/2018

**Natureza do Trabalho:** em dupla.

- Entregar a lista através do e-mail [palomar.cris@gmail.com](mailto:palomar.cris@gmail.com) até às 23:59 do dia 21/10. Enviar apenas os programas fonte (.c). Os nomes dos arquivos fontes deverão ter no máximo 8 caracteres (além da extensão) e não poderão conter caracteres especiais ou acentuação.
- Cada programa deverá conter um comentário com o Ra e nome do(s) aluno(s).
- **Não serão aceitas listas entregues fora do prazo determinado**
- A lista merecerá uma nota de 0 a 10, sendo o critério de avaliação definido pelo professor.
- Exercícios que sejam considerados como fruto de algum tipo de fraude por parte da equipe (como cópia total ou parcial, por exemplo) atribuirão nota zero na atividade para TODOS os envolvidos.
- **Os programas serão compilados e executados no ambiente Cygwin ou CodeBlocks.**
- IDENTAR todos os programas.

## EXERCÍCIOS

1) (2,5) **USAR NOTAÇÃO PONTEIRO.** Cada livro que é publicado atualmente possui um código identificador específico, que não se repete, chamado ISBN (International Standard Book Number). Esse código é uma sequência de 10 dígitos decimais, mas em alguns casos a letra X (em maiúsculas) pode também aparecer como sendo o 10º dígito. A apresentação desse código geralmente é feita intercalando os dígitos com alguns hífen, mas a posição desses separadores pode variar bastante entre um caso e outro. A fórmula de validação é relativamente simples: duas somas, que chamaremos de s1 e s2 são calculadas a partir dos 10 dígitos do ISBN (os possíveis hífen são desconsiderados), com s2 sendo a soma dos valores parciais de s1 após o valor de cada dígito do ISBN ter sido a ele adicionado. Caso o 10º caractere seja um X, considera-se que seu valor é 10. O ISBN é entendido como correto se o valor final de s2 for divisível por 11. A título de exemplo, consideremos o ISBN 0-13-162959-X, que corresponde ao livro "Computers Network" do Tanenbaum.

Dígitos considerados	0	1	3	1	6	2	9	5	9	X
Acumulador s1	0	1	4	5	11	13	22	27	36	46
Acumulador s2	0	1	5	10	21	34	56	83	119	165

Como o valor final de s2 é 165, um número divisível por 11, o ISBN está correto. Sua tarefa é fazer um programa que receba diversos ISBN e determine se ele é válido ou não. Se for válido, imprimir a mensagem 'ISBN 999999999-9 é válido', senão, imprimir 'ISBN inválido: 99' onde o 99 indica o resto da divisão do valor final de s2 por 11.

Assumir que todos os ISBN serão informados no formato '999999999-9' (com um único hífen, portanto, separando os nove dígitos iniciais do dígito final) e que o programa deve encerrar quando o usuário informar a string 'FIM' para o novo ISBN a ser processado. A seguir são dados alguns exemplos de ISBN válidos e inválidos.

Válidos	Inválidos
089237010-6	123456789-0
013152447-X	292020202-0
156884030-6	013152447-x
081252030-0	111110111-1

2) (2,5) **USAR NOTAÇÃO PONTEIRO.** Implemente o programa C abaixo, definindo também a rotina Cripto, conforme as especificações apresentadas a seguir:

```
#include <stdio.h>
int main()
{
    char texto1[9], texto2[9], texto3[9] ;
    printf( "Informe um texto de até 8 caracteres:" ) ;
    scanf( "%[^\n]", texto1 ) ;
    Cripto( texto1, texto2 ) ;
    Cripto( texto2, texto3 ) ;
    printf( "O texto criptografado eh: \n%s\n e descriptografado
eh: \n%s\n", texto2, texto3 ) ;
    return 0 ; }
```

### Rotina: Cripto

Interface da rotina:

Recebe 2 parâmetros: Variável contendo a string a ser (des)criptografada e variável onde o resultado deverá ser armazenado.

Comportamento:

Com base no conteúdo de duas strings de controle definidas dentro da subrotina, esta rotina gera a criptografia de um dado. Sua lógica interna consiste em obter um a um os caracteres da string original e, após pesquisá-los nas cadeias de controle, substituí-los conforme o seguinte princípio:

- a) Se o caracter existir na primeira string de controle, colocar em seu lugar o caracter equivalente na segunda string de controle;
- b) Senão, se o caracter existir na segunda string de controle, colocar em seu lugar o caracter equivalente na primeira string de controle;
- c) Senão, se o caracter não existir em nenhuma das duas strings de controle, deixá-lo inalterado.

Dicas:

Montar as duas strings de controle com o mesmo tamanho, mas sem nenhum tipo de repetição de caracteres. Considerar diferentes as versões maiúscula e minúscula de uma mesma letra. Utilizar, além das letras, também números e caracteres especiais.

3) (2,5) **USAR NOTAÇÃO PONTEIRO.** Desenvolva um programa de acordo com as especificações abaixo:

- a. Crie uma função que receba duas strings como parâmetro, bem como um valor inteiro representando uma posição.
- b. insira a segunda string na primeira, na posição indicada pelo valor.
- c. Fazer um programa que receba duas strings do usuário, o valor da posição, chame a função anteriormente implementada e exiba o resultado ao usuário.
- d. Repetir o programa enquanto a posição indicada seja maior que zero.

4) (2,5) **USAR NOTAÇÃO PONTEIRO.** Em uma dada string devemos encontrar o CPF do cliente e o valor de suas faturas pendentes, mas esses valores estão embaralhados em caracteres não numéricos. Sua tarefa é encontrar o CPF e somar o valor total das faturas pendentes de cada cliente. (1898

### **Entrada:**

A entrada é composta por um número que indica a quantidade de clientes. Cada cliente possui uma string contendo entre 1 e 60 caracteres ('0'-'9' e outros caracteres não numéricos). Os 11 primeiros dígitos numéricos correspondem ao CPF do cliente. Os demais números são separados pela casa decimal, que sempre vem acompanhada de um "." com dois dígitos na sequência indicando os centavos.

### **Saída:**

Imprima o CPF do cliente, conforme apresentado abaixo e a soma dos valores das faturas em aberto, sempre com duas casas decimais.

Exemplo de Entrada:

```
4
7a5a6adfg4a4adsfgsd9fg6aa904aa91#$$%11.2342**&774667
11#$$%11111*&`1111&*(1198-=&8789."34"8%(&567.22%99
123ccc456dd875tt.q10120.891000.40
```

Exemplo de Saída

```
CPF: 75644969049 Soma: 42774779.23
CPF: 11111111111 Soma: 11997455,56
CPF: 12345687510 Soma: 1121.29
```