

IFPB - Instituto Federal de Educação, Ciência e Tecnologia da Paraíba

Curso de Engenharia de Computação

Disciplina: Algoritmos e Programação

Semestre Letivo: 2016.2

Professor: Marcelo Siqueira / Henrique Cunha

Nome:

---

### **PROVA 4**

#### **Instruções:**

1. Não é permitido consultar livros, anotações, Internet etc.
2. Não é permitido conversar durante o horário da prova
3. Não é permitido usar pendrive
4. Todas as questões têm o mesmo valor.
5. Salve seus arquivos em alguma pasta que não seja a área de trabalho.
6. Para cada problema você tem um conjunto de dados de entrada e de saída que servirão como uma referência na hora de testar seu código.
7. **A interpretação faz parte da prova.**

DURAÇÃO: 3 Horários

**(13h - 15h30)**

Campina Grande, PB  
05 de Abril de 2017

### Questão 1:

Leet é uma maneira de escrever palavras na qual algumas letras, sílabas ou palavras são substituídas por números ou símbolos. Isso é muito usado no contexto da informática para confundir leitores "não iniciados" ou escrever de forma resumida (por exemplo, na escrita de tweets).

Dentro desse contexto, você vai escrever um programa que transforma palavras na grafia comum em palavras na escrita Leet de forma invertida.

Ao final, o programa deve informar a quantidade de substituições realizadas.

Use a seguinte codificação:

a, A: @  
e, E: 3  
i, I: 1  
o, O: 0  
t, T: 7  
s, S: 5

Não haverão palavras acentuadas.

Se o usuário informar uma string vazia, o programa emitirá uma mensagem de erro e a quantidade 0 (zero).

Se o usuário informar uma string que contém pelo menos um número, o programa escreverá uma mensagem de erro e a quantidade 0 (zero).

O código deve fazer uso de **funções** sempre que necessário, observando os princípios da modularização e separação de objetivos.

#### Entrada

>> marcelo

>> 1999

>> Turing

#### Saída

0l3cr@m  
3

erro  
0

gn1ru7  
2

## Questão 2: Conversão para binário

Faça um programa para converter de decimal, octal ou hexadecimal para **binário**. O seu programa deve primeiro perguntar qual é a base da qual se deseja converter (8, 10 ou 16), em seguida, o número que se quer converter.

Seu programa deve verificar se o usuário digitou um número possível de representar naquela base. Por exemplo, o número 739 não pode estar na **base 8**, por conter um dígito que não pode ser representado na base 8, nesse caso o dígito 9.

Uma vez, digitado um número corretamente, o programa deve imprimir na tela o número convertido para a base 2.

Atente para o valor de cada *feature* que você implementar:

- Valida a entrada de dados corretamente. Ou seja, faz o programa aceitar apenas as bases 8, 10 e 16 (10 pontos)
- Valida um número octal corretamente (15 pontos)
- Valida um número hexa corretamente (15 pontos)
- Valida um número decimal corretamente (15 pontos)
- Converte um número octal corretamente (15 pontos)
- Converte um número hexa corretamente (15 pontos)
- Converte um número decimal corretamente (15 pontos)

Cada *feature* só terá pontuação máxima se você usar funções sempre que necessário, observando os princípios da modularização e separação de objetivos.

**Dica:** Para realizar a conversão de decimal para binário, basta usar a técnica de divisões sucessivas e guardar o resto da divisão.

### Entrada

Base: 8

Num: 839

Base: 8

Num: 777

Base: 5

Base: 10

Num: 1AB3

Base: 16

Num: 1AB3

Base: 10

### Saída

O número 839 não pode ser representado na base 8. Digite outro número.

11111111(2)

Base inválida. Apenas as bases 8, 10 e 16 são válidas. Entre novamente com a base.

O número 1AB3 não pode ser representado na base 10. Digite outro número.

1101010110011(2)

Num: 31442	111101011010010(2)
------------	--------------------