Instituto Federal de Educação Ciência e Tecnologia do RN
Diretoria Acadêmica de Tecnologia da Informação
Curso de Tecnologia em Redes de Computadores
Disciplina de Programação de Computadores
Prof: Galileu Batista e Charles Freitas

Operações com bits e bytes

1) Faça a conversão de bases dos seguintes números para a base 10:
    a) 340 na base 8 é igual a _____.
    b) 123 na base 16 é igual a _____.
    c) 1001 na base 2 é igual a _____.
    d) ABC na base 16 é igual a _____.

2) Um elemento da base dez tem dez possíveis valores, do zero ao nove.
    a) Cada elemento na base dezesseis tem quantos possíveis valores?
    b) A base 62 tem 62 elementos diferentes. Suponha que os elementos naquela base são: 0, 1, 2, …., 9, a, b, c, ….z, A, B, C, D, …. Z. Nesse cenário a letra **a** representa o valor 10, **b** o 11, …. , **z** o 35, **A** o 36, **B** o 37, até finalmente **Z** que representa 61. Qual o valor na base 10, dos seguintes elementos na base 62:
        - a3
        - A3
        - 51z
        - 100

3) Uma base ainda mais diferente é a base 256. Nessa deve haver 256 possíveis símbolos para representar cada um dos seus elementos. Como pode ser difícil encontrar essa quantidade de símbolos, adote o seguinte critério:

    00 – representa o zero na base 256
    01 – representa o 1 na base 256
    02 – representa o 2 na base 256
    09 – representa o 9 na base 256
    0A – representa o 10 …
    0B – representa o 11
    …
    0F – representa o 15
    10 – representa o 16
    11 – representa o 17
    …
    1A – representa o 26
    …
    1F – representa o 31
    20 – representa o 32
    …
    32 – representa 0 50
    A0 – representa o 160
    ….
    FF – representa o 255

a) Seguindo esse padrão tente descobrir qual o elemento da base 256 é representado por cada um dos seguintes pares:
   - o par 34 – representa o elemento 52
   - o par 72 – representa o _____
   - o par 4C – representa o _____
   - o par C2 – representa o _____
   - o par F2 – representa o _____

b) Considerando que cada um dos pares representa um elemento na base 256, determine o valor na base 10 dos seguintes elementos na base 256 (não esqueça de seguir o padrão: o elemento (no caso, o par) mais à direita é multiplicado por 256^0, o seguinte por 256^1, o próximo por 256^2 e assim por diante:
   - 4C 32 – representa o _____
   - 72 F2 – representa o _____
   - 36 8A 25 – representa o _____
   - 33 78 54 D5 – representa o _____
   - 08 00 00 00 – representa o _____

4) Lembra do algoritmo que faz mudança de base? Faça um programa que recebe um número inteiro decimal e mostra esse número na base 6.

5) Considere o conjunto de bytes a seguir que representa o cabeçalho de um pacote IPv4. Usando como referência o formato do pacote em https://en.wikipedia.org/wiki/Internet_Protocol_version_4 (e considerando que os dados agrupados são *big-endian*), responda indicando o trecho de código em Python que foi usado para obter cada resposta.

```
IpPacket = b"\x45\x00\x00\x38\x04\x88\x40\x00\x80\x11\x69\
x34\xc0\xa8\x01\x69\xac\xd9\x1e\x0e"
#Se preferir, crie um pacote usando struct
```

- Qual o TTL desse datagrama?
- Quais os endereços de origem e destinho desse datagrama?
- Qual o tamanho do header e o tamanho total desse datagrama?
- Indique o valor em cada um dos (três) flags relacionados ao datagrama.

6) Faça um programa que leia os primeiros 6 bytes da imagem JPEG em anexo. Nas posições 4 e 5 há um valor que especifica o tamanho dos metadados presentes nessa imagem. Obtenha esse número (chame-o `app1DataSize`). Feche o arquivo.

- Abra o arquivo novamente, leia 4 bytes e os ignore. Agora leia o número de bytes em `app1DataSize` para `app1Data`. Na posição 16 de `app1Data` há 2 bytes que indicam quantos metadados essa imagem tem. Descubra-o e informe.

- A partir da posição 18 de `app1Data` há efetivamente os metadados. Cada metadado tem o formato:
  - 2bytes - qual o metadado. Você pode obter toda a lista aqui: https://exiftool.org/TagNames/EXIF.html

- 2bytes - o tipo do metadado. Valores possíveis são (1- unsigned byte; 2 – string; 3 – unsigned short; 4 – unsigned long, …)
- 4 bytes - o número de repetições que esse metadados tem. Exemplo: tem tipo inteiro, mas se repete 5 vezes.
- 4 bytes - o valor do metadado. Se são necessários mais de 4 bytes, indica o *offset* no arquivo onde o valor está, contado a partir da posição 12 (ou seja, deve somar 12 para chegar na posição real).

Identifique a partir dos metadados, qual a altura e a largura dessa imagem. O metadado para altura é 0x0101 e para largura 0x0100.