

## Projeto Prático #1 (parte I) (Comandos sequencias, condicionais e de repetição)

Professora: Myriam / Leyza

Estudante: \_\_\_\_\_

**Questão 1 pj01-I**

Anna vive em Algol e Bob vive em Betelgeuse. Uma longa distância os separa, dado que as estrelas estão em diferentes constelações - Perseus e Orion. Eles encontraram uma forma de se comunicar via e-mail, mas perceberam que algumas letras podem ser alteradas durante a transmissão devido à longa distância.

Anna propôs uma forma simples para tentar identificar erros, supondo que as mensagens contém apenas letras (maiúsculas e minúsculas) e espaços, e que linha de comunicação pode corromper no máximo um bit em cada byte.

A proposta é transmitir esses símbolos usando o código ASCII - por exemplo, ao invés de transmitir 'A', é transmitido o valor 65. Anna e Bob observaram que, para a faixa de valores utilizada para as letras, os valores ocupam apenas um byte e que o seu bit mais significativo não precisa ser usado. Por exemplo, o valor 65 em binário é 01000001 - note que o bit mais à esquerda é zero. Eles decidiram explorar este bit para determinar se a letra transmitida foi corrompida ou não.

A proposta consiste em alterar este bit mais significativo de tal forma que a soma dos bits seja sempre par (2, 4, 6 or 8). Seguem alguns exemplos de codificação:

símbolo	código ASCII original	binário original	nro de bits com valor 1	binário codif. para que a soma dos bits seja par	decimal codif.
'A'	65	01000001	2	01000001	65
'B'	66	01000010	2	01000010	66
'C'	67	01000011	3	11000011	195
'.'	46	00101110	4	00101110	46
' '	32	00100000	1	10100000	160

Por exemplo, considere o caracter 'C': como a codificação do valor original (67) tem um número ímpar de bits com valor 1, a codificação altera o bit mais significativo de 0 para 1. Com isso, o valor 195 é transmitido.

Como a linha de comunicação pode corromper no máximo um bit em cada byte, aqueles cuja soma dos bits seja um número ímpar são considerados corrompidos e devem ser descartados. Por exemplo, ao receber o valor codificado de 103 (que em binário é 1100111), pode-se afirmar que esse símbolo foi corrompido durante a transmissão, por ter um número ímpar de bits.

Sua tarefa possui duas partes (crie um menu de forma a possibilitar que o usuário escolha o que deseja fazer):

**A** dada a mensagem de entrada: (a) codificar segundo a regra acima e (b) mostrar o resultado. Exemplo:

Mensagem de entrada:  
Anna.

Mensagem codificada:  
65 238 238 225 46

**B** dada a mensagem codificada: (a) checar cada letra e (b) mostrar a mensagem decodificada inserindo um \* em letras corrompidas.Exemplo:

Mensagem de entrada (codificada):  
65 238 236 225 46

Saída:  
An\*a.

Assuma que a mensagem pode conter apenas letras (maiúsculas e minúsculas) e espaços. O final da mensagem é sempre dada pelo símbolo '.' - para facilitar a implementação, você pode supor que ele jamais será corrompido.

## Questão 2 pj01-I

Faça o teste de mesa para o seguinte código. Além disso, faça um texto explicando detalhadamente o que os operadores `&` e `<<`.

```
1 #include <stdio.h>
2
3 #define SIZE (1 << 4)
4 int main()
5 {
6     int x, y, i;
7     for (y = SIZE - 1; y >= 0; y--, putchar('\n')) {
8         for (i = 0; i < y; i++)
9             putchar(' ');
10        for (x = 0; x + y < SIZE; x++)
11            printf((x & y) ? "  " : "* ");
12    }
13    return 0;
14 }
```

Além disso, discuta:

1. O que muda se o início do código for `#define SIZE (1 << 7)`?
2. Existe outra forma de solucionar esse problema?