

Podemos facilmente perdoar uma criança que tem medo do escuro;
a real tragédia da vida é quando os homens têm medo da luz.

Platão

Lista de Exercícios III - Problemas envolvendo manipulação de String

[1] No sistema de numeração romana as cifras escrevem-se com as letras I, V, X, L, C, D e M. Exemplo: 125 é representado por CXXV. Implemente um programa que leia um número inteiro, caso ele esteja dentro do intervalo de 1 a 999, mostre o número romano equivalente, caso contrário mostre “*valor fornecido fora dos limites operacionais*”.

DICA: Utilize um vetor de strings para representar as unidades em números romanos, outro para as dezenas e outro para as centenas. Separe, usando mod e div, o valor da unidade, da dezena e da centena do número fornecido e use estes valores como índices de consulta aos vetores em romanos.

Observação: tabela com as referências da numeração romana disponível em http://pt.wikipedia.org/wiki/Números_romanos

[2] Escreva um “*Mini-conjugador de verbos regulares*”, ou seja, um programa que receba como entrada um verbo regular no infinitivo e mostre a sua conjugação. Como limitações: (i) considerar apenas verbos regulares da primeira conjugação (terminação “ar”), segunda conjugação (terminação “er”) e terceira conjugação (terminação “ir”); (ii) mostrar apenas os tempos Presente do Indicativo, Pretérito Perfeito e Futuro do Presente.

DICA: Utilize um vetor de strings para representar os pronomes pessoais, e um vetor de desinências verbais para cada tempo/conjugação.

Observação: modelo de conjugação em http://pt.wikipedia.org/wiki/Modelos_de_conjugação_dos_verbos

[3] *Leet* (ou l33t) é uma forma de se escrever o alfabeto latino usando outros símbolos em lugar das letras, como números ou caracteres similares graficamente. O uso do leet reflete uma subcultura relacionada ao mundo dos jogos de computador e da Internet; veja explicação em: <http://pt.wikipedia.org/wiki/Leet>

Usando vetores de strings e a tabela do alfabeto leet disponível no link acima, desenvolva um pequeno tradutor de leet, cuja função é ler uma frase do usuário e depois mostrá-la em seu formato original e 3m \$3μ f0rm4t0 l33t. Como são várias as possibilidades de tradução para cada caracter, mostre três formatos, fácil (deixando algumas letras normais), médio (trocando a maior parte das letras, mas com equivalentes mais conhecidos) e difícil (trocando todas as letras por equivalentes elaborados).

DICA: Utilize um vetor de strings para armazenar cada uma das versões dos caracteres em leet (alfabetos recodificados em fácil, médio e difícil) e então use o valor decimal (ASCII) de cada caracter fornecido na frase original como índice de acesso aos vetores com o alfabeto leet.

[4] Escreva um programa que leia um número inteiro, caso ele esteja dentro do intervalo de 1 a 999, mostre seu valor por extenso, caso contrário mostre “*valor fornecido fora dos limites operacionais*”. Ou seja, se o usuário digitar 158, o programa mostra “158: cento e cinquenta e oito”

DICA: Utilize vetor de strings para representar as unidades (1..9), dezenas especiais (10..19), dezenas gerais (10, 20 etc) e centenas (100, 200 etc). Separe, usando mod e div, o valor da unidade, da dezena e da centena do número fornecido e use estes valores como índices de consulta aos vetores contendo as strings equivalentes. Serão necessários vários testes para verificar o uso da letra “e” entre os valores. Para testar sua implementação, faça uma repetição de 1 a 999, e veja se todos estão grafados corretamente.

[5] Desenvolva um programa para o “*Jogo da Forca*”. Para tal, crie um vetor de strings inicializado com um conjunto de palavras-chave (por exemplo: nomes de capitais do Brasil, ou times de futebol da Serie A ou Países da América do Sul, etc). Sorteie uma das palavras para ser o segredo e forneça seis vidas para o usuário acertar o segredo. A cada rodada informe o número de vidas disponíveis e a disposição das letras acertadas e ausentes na palavra segredo (lembre de quando brincava com este jogo em seu caderno escolar na infância), mostre também quais as letras que já foram usadas (e não compute acerto ou erro no caso do usuário repetir uma letra já fornecida).

Vale a pena ler para conhecer mais sobre conjuntos de caracteres:

- ASCII (<http://en.wikipedia.org/wiki/ASCII>),
- Unicode (<http://en.wikipedia.org/wiki/Unicode>),
- UTF-8 (<http://en.wikipedia.org/wiki/UTF-8>),
- ISO-8859 (http://en.wikipedia.org/wiki/ISO_8859).