Trabalho de Programação 3

Processador 8086 da Intel

1. Descrição Geral

Você deverá desenvolver um programa em assembler do 8086 da Intel para decriptar um conjunto de frases, usando a cifragem de Affine (http://en.wikipedia.org/wiki/Affine_cipher).

Cada frase a ser decriptada é chamada de "texto cifrado" e o resultado da decriptação é o "texto claro" (ou texto decifrado). As frases a serem decifradas serão fornecidas através de um arquivo texto.

Para realizar a decifragem o programa deve procurar e descobrir (por "força bruta") os valores para os parâmetros "*a*" e "*b*" que permitem a decifragem de cada frase.

Para realizar a criptoanálise (processo de quebra de código), o programa atribui valores iniciais aos dois parâmetros e realiza a decifragem. Em seguida, compara o resultado com um dicionário de palavras válidas. Será considerado como o resultado da decifragem a primeira frase que contenha uma das palavras do dicionário.

Então, o programa deve colocar no visor o resultado encontrado, indicando os parâmetros usados no processo de criptoanálise.

O algoritmo a ser empregado é o de "força bruta", onde todas as combinações possíveis são tentadas, até que se encontre aquela que forneça um texto claro com significado. Uma diferença importante em relação ao trabalho do CESAR é que, agora, a verificação do resultado será feito automaticamente e não mais pelo usuário.

2. Especificação do Trabalho (idêntico ao do trabalho do CESAR)

O texto cifrado será formado apenas pelas letras maiúsculas, representadas pelos seus códigos ASCII ('A'=41H, 'B'=42H, ..., 'Z'=5AH). Existem 26 letras maiúsculas na tabela ASCII.

Para determinar o valor de cada caractere do texto claro deverá ser utilizado o seguinte procedimento:

1) Converter o código ASCII das letras 'A', 'B', ... 'Z' do texto cifrado em números 0, 1, ..., 25. Esses números serão colocados em um vetor chamado de "*Y*".

2) Considerando-se "*i*" como a posição do número "y_i" no vetor "*Y*", calcula-se o número "x_i", a ser colocado na posição "*i*" de um vetor "*X*". O cálculo a ser feito é o seguinte (conforme a decifragem de Affine):

$$x_i = a^{-1}(y_i - b) \ mod \ 26$$

> "*mod* 26" representa o resto da divisão por 26.
> Assim, "*n mod* 26" é o resto da divisão de "*n*" por 26.

3) Converter os números do vetor "*X*" (valores que estão entre 0 e 25) para o código ASCII das letras 'A', 'B', ... 'Z'. O resultado dessa operação será o texto claro.

Detalhamento da Decifragem

A expressão de cálculo da decifragem é obtida pela inversão da expressão de cifragem, conforme abaixo:

$$\begin{aligned} y_i &= (a.x_i + b) \ mod \ 26 \\ (y_i - b) &= (a\ x_i) \ mod \ 26 \\ a^{-1}(y_i - b) &= a^{-1}(a\ x_i) \ mod \ 26 \\ a^{-1}(y_i - b) &= a^{-1}\ a\ (x_i) \ mod \ 26 \end{aligned}$$

Nesse ponto percebe-se que só é possível isolar o valor de "x_i" se "$1 = a.a^{-1}$", com essa multiplicação sendo realizada em módulo 26. Dessa forma, se "$1 = a.a^{-1}$ *mod* 26", a expressão final torna-se a expressão de decifragem de Affine:

$$x_i = a^{-1}(y_i - b) \ mod \ 26$$

Sendo assim, apesar do programa receber e apresentar o valor do parâmetro *a*, a expressão de decodificação utiliza a^{-1}. Logo, o programa deve determinar o valor de "a^{-1}" a partir do valor de "*a*" escolhido, satisfazendo a seguinte expressão:

$$1 = a.a^{-1} \ mod \ 26.$$

Entretanto, a análise dessa expressão revela que ela só pode ser satisfeita para alguns valores de *a*. Ou seja, existem valores de "*a*" para os quais não é possível encontrar um valor para "a^{-1}" que tornem verdadeira a expressão (para

verificar isso, por exemplo, tente encontrar o valor de "a^{-1}" para "a=2"). Os valores de "a" que admitem "a^{-1}", segundo a expressão, estão listados na Tabela 1 a seguir.

a	1	3	5	7	9	11	15	17	19	21	23	25
a^{-1}	1	9	21	15	3	19	7	23	11	5	17	25

Tabela 1 – Valores válidos de "a" e os correspondentes a^{-1}

3. Procedimentos do programa

A seguir é apresentada a sequência de procedimentos a serem implementados no programa:

[1] Ao iniciar, o programa deve apresentar na tela a identificação do aluno (nome e número do cartão).

[2] Em seguida, o programa deve solicitar ao usuário o nome do "Arquivo de Dicionário".
   a. Deve ser colocada uma mensagem na tela solicitando que o usuário digite o nome do arquivo.
   b. O programa deve permitir o uso do <BS> (backspace) para eventuais correções.
   c. Ao encerrar a digitação do texto cifrado o usuário deverá digitar <ENTER>.

[3] O "Arquivo de Dicionário" deve ser lido.
   a. O arquivo será formado por uma lista de palavras.
   b. Cada palavra estará em uma linha do arquivo, a partir da coluna zero.
   c. As palavras terão, todas, um mínimo de 5 caracteres e um máximo de 30 caracteres. Palavras com menos de 5 caracteres devem ser ignoradas.
   d. O final do arquivo indica o final da lista de palavras.
   e. O "Arquivo de Dicionário" poderá conter até um total de 1000 (mil) palavras.
   f. Caso ocorra algum erro ao abrir e ler o "Arquivo de Dicionário", o erro deverá ser informado em mensagem na tela e o programa deverá ser encerrado.
   g. Caso o "Arquivo de Dicionário" seja lido corretamente, deve ser apresentado no visor uma mensagem informado esse fato. Além disso, deve ser informado o número de palavras válidas lidas.

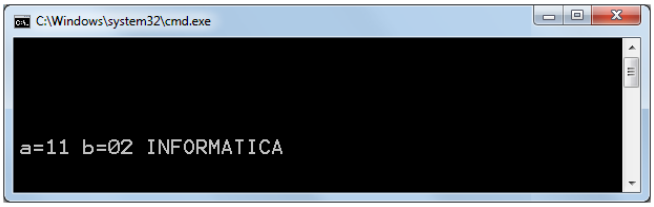
[4] Na sequência, o programa deve solicitar ao usuário o nome do "Arquivo com Texto Cifrado".
   a. Deve ser colocada uma mensagem na tela solicitando que o usuário digite o nome do arquivo.
   b. O programa deve permitir o uso do <BS> (backspace) para eventuais correções.
   c. Ao encerrar a digitação do texto cifrado o usuário deverá digitar <ENTER>.

[5] O "Arquivo com Texto Cifrado" deve ser lido.
   a. O arquivo será formado por até 20 frases (linhas de texto). Cada frase poderá estar cifrada com um par diferente de parâmetros.
   b. Cada frase poderá ter entre 5 e 60 caracteres. Linhas com menos de 5 caracteres devem ser ignoradas. Caracteres além dos 60 podem ser ignorados.
   c. Todas as frases iniciam na coluna zero.
   d. Os caracteres usados nas frases são apenas as letras entre "A" e "Z" (maiúsculas ou minúsculas).
   e. No caso de letras minúsculas, o programa deverá convertê-las para maiúsculas, antes de aplicar o algoritmo de decifragem.
   f. Caso ocorra algum erro ao abrir e ler o "Arquivo com Texto Cifrado", o erro deverá ser informado em mensagem na tela e o programa deverá ser encerrado.
   g. Caso o "Arquivo de Dicionário" seja lido corretamente, deve ser apresentado no visor uma mensagem informado esse fato. Além disso, deve ser informado o número de frases lidas.

[6] Com os dados lidos dos arquivos, o programa deve iniciar a busca pelo conjunto de parâmetros que permite decifrar cada frase fornecida.

a. Se o programa encontrar uma solução, esta deverá ser colocada na tela junto com os parâmetros, da seguinte forma:

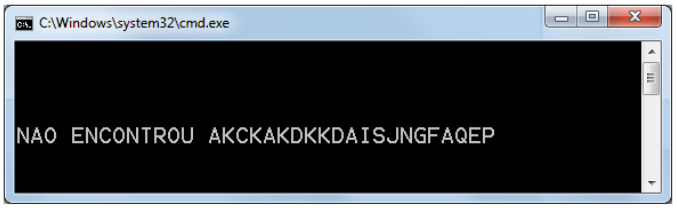
a=XX b=XX Frase Decifrada



Exemplo de saída, caso a=11, b=2 e a frase decifrada seja “INFORMATICA”

b. Se o programa não puder encontrar uma solução, esse fato deverá ser informado. Além disso, a frase cifrada deverá ser colocada no visor, da seguinte forma:

NAO ENCONTROU Frase Cifrada



Exemplo de saída, caso não tenha encontrado a solução

[7] Após informar o resultado da criptoanálise o programa deverá encerrar.

4. Programa “minimamente operacional”

O programa será considerado minimamente operacional se for capaz de realizar os passos correspondentes à leitura do “Arquivo de Dicionário” e “Arquivo com o Texto Cifrado” (passos [1] até [5] acima).

5. Entregáveis: o que deve ser entregue?

Devem ser entregues dois arquivos: o arquivo fonte (.ASM) e um relatório em formato PDF.

A implementação entregue no arquivo (.ASM) deve seguir, rigorosamente, a especificação do trabalho no que diz respeito aos procedimentos de entrada (arquivos e teclado) e saída (tela) de dados.

Para gerar arquivos PDF a partir de qualquer editor no Windows, pode-se usar o “primo pdf” (http://www.primopdf.com/) ou o “doPDF” (http://www.dopdf.com/br/).

No relatório da implementação devem estar presentes os seguintes elementos:

- Identificação do aluno;
- Principais dificuldades encontradas;
- Descrição do programa implementado (pode ser feita sobre uma implementação em “C”).

Observação: não colocar a listagem completa dos programas no relatório.

6. Avaliação

O trabalho só será considerado entregue se forem enviados os arquivos solicitados e o programa estiver **minimamente operacional**.

A implementação final do trabalho deverá ser entregue até 27 de novembro. Admite-se essa entrega com até uma semana de atraso. Nesse caso a nota final do trabalho será obtida pela diminuição de 20,0 pontos (de um total de 100,00) da nota alcançada na avaliação do mesmo. Não serão aceitos trabalhos entregues além dos prazos estabelecidos.

7. Observações

Recomenda-se a troca de ideias entre os alunos. Entretanto, a identificação de cópias de trabalhos acarretará na aplicação do Código Disciplinar Discente e a tomada das medidas cabíveis para essa situação.

O professor da disciplina reserva-se o direito, caso necessário, de solicitar uma demonstração do programa, onde o aluno será arguido sobre o trabalho como um todo. Nesse caso, a nota final do trabalho levará em consideração o resultado da demonstração.