Trabalho de Programação 1

Processador RAMSES

1. Descrição Geral

Você deverá desenvolver um programa em assembler do RAMSES para criptografar uma frase, usando a cifragem de Affine (http://en.wikipedia.org/wiki/Affine_cipher).

A frase a ser criptografada é chamada de "texto claro" e a frase criptografada é chamada de "texto cifrado".

2. Especificação do Trabalho

O texto claro conterá somente 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 cifrado deverá ser utilizado o seguinte procedimento:

1) Para cada caractere do texto claro, converte-se as letras 'A', 'B', ... 'Z' em números 0, 1, ..., 25. Esses números serão colocados em um vetor chamado de "X".

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

$$y_i = (a.x_i + b) \ mod \ 26$$

"*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 "*Y*" (valores que estão entre 0 e 25) em letras 'A', 'B', ... 'Z'. O resultado dessa operação será o texto cifrado.

O programa deverá usar como texto claro o conteúdo da memória existente a partir do endereço DAH (218) até o final da memória. O final do texto será indicado por um byte com valor 00H.

O programa deverá calcular os caracteres do texto cifrado e colocá-los na mesma área onde estava o texto claro, sobrescrevendo-o. O texto cifrado também deverá ser terminado por um byte com 00H.

Os valores de "*a*" e "*b*", que aparecem na expressão de criptografia, serão valores entre 0 e 25 (inclusive). Eles estarão disponíveis nos endereços D8H (216) e D9H (217) respectivamente.

3. Casos de Teste

Na tabela abaixo estão apresentados os casos de teste para os quais os programas devem estar funcionando corretamente, para serem considerados "minimamente operacionais". Cada linha da tabela representa um caso de teste. Nas colunas "*a*" e "*b*" estão indicadas as constantes usadas na cifragem (vide expressão de cifragem de Affine); o "texto claro" é sempre "INFORMATICA" (notar que estão sendo utilizados apenas os valores ASCII entre 'A' e 'Z'); finalmente na coluna "Texto Cifrado" está o resultado que o programa deve fornecer.

Caso	*a*	*b*	Texto Claro	Texto Cifrado
1	1	0	**INFORMATICA**	**INFORMATICA**
2	1	10	**INFORMATICA**	**SXPYBWKDSMK**
3	10	0	**INFORMATICA**	**CAYKOQAICUA**
4	10	10	**INFORMATICA**	**MKIUYAKSMEK**

Por exemplo, o caso 2 de teste seria colocado na memória para o processamento de seu programa, conforme tabela (a) abaixo; o resultado correto do processamento estará na memória conforme a tabela (b) abaixo.

Endereço	Conteúdo	Comentário
D8H (216)	01H (1)	Valor de *a*
D9H (217)	0AH (10)	Valor de *b*
DAH (218)	49H ('I')	Texto a ser cifrado
DBH (219)	4EH ('N')	Texto a ser cifrado
DCH (220)	46H ('F')	Texto a ser cifrado
DDH (221)	4FH ('O')	Texto a ser cifrado
DEH (222)	52H ('R')	Texto a ser cifrado
DFH (223)	4DH ('M')	Texto a ser cifrado
E0H (224)	41H ('A')	Texto a ser cifrado
E1H (225)	54H ('T')	Texto a ser cifrado
E2H (226)	49H ('I')	Texto a ser cifrado
E3H (227)	43H ('C')	Texto a ser cifrado
E4H (228)	41H ('A')	Texto a ser cifrado
E5H (229)	00H (0)	Delimitador de final do texto

Tabela (a)

Endereço	Conteúdo
D8H (216)	01H (1)
D9H (217)	0AH (10)
DAH (218)	53H ('S')
DBH (219)	58H ('X')
DCH (220)	50H ('P')
DDH (221)	59H ('Y')
DEH (222)	42H ('B')
DFH (223)	57H ('W')
E0H (224)	4BH ('K')
E1H (225)	44H ('D')
E2H (226)	53H ('S')
E3H (227)	4DH ('M')
E4H (228)	4BH ('K')
E5H (229)	00H (0)

Tabela (b)

4. Entregáveis: o que deve ser entregue?

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

A implementação entregue no arquivo .RAD deve seguir, rigorosamente, a especificação do trabalho no que diz respeito à localização na memória das variáveis de entrada e saída.

No relatório da implementação devem estar presentes os seguintes elementos:

- Identificação do aluno;
- Principais dificuldades encontradas.
- Descrição do programa implementado (NÃO COLOCAR O FONTE NO RELATÓRIO).

O trabalho só será considerado entregue se forem enviados os arquivos acima descritos e o programa estiver **minimamente operacional**, ou seja, deve fornecer os resultados corretos para os casos de teste descritos nessa especificação.

O trabalho deverá ser entregue até a **data prevista**. Admite-se a entrega do trabalho com até uma semana de atraso. Nesse caso a nota final do trabalho será reduzida de 20,0 pontos (do total de 100,00). Não serão aceitos trabalhos entregues além dos prazos estabelecidos.

Os programas considerados funcionais serão corrigidos de forma automática pela aplicação de novos casos de teste e a nota final do trabalho dependerá do número de casos em que o programa produziu a resposta correta.

Dentre os programas que fornecerem os resultados corretos para todos os casos de teste, aqueles mais rápidos (que utilizarem o menor número de acessos a memória) concorrerão a um "bônus" de 10,0 pontos na nota.

5. Observações

Recomenda-se a troca de ideias entre os alunos. Entretanto, a identificação de cópias de trabalhos acarretará na aplicação do Código Disciplina 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.