



PROBLEMA 1: Cifrador Enigma (Visto 1) – v2: Rotores Corrigidos

A solução deste problema é individual, de forma a que o aluno comprove seu domínio sobre o assunto. Após resolvê-lo, o aluno deve fazer um pequeno vídeo (máximo de 2 minutos) de forma a comprovar sua solução. Cada aluno deverá enviar sua listagem e seu vídeo. Serão aceitos apenas dois arquivos por aluno. A listagem de sua solução deve ser autocontida, ou seja, ela simplesmente será copiada para o CCS para comprovar seu correto funcionamento. Será verificado o correto funcionamento de todo programa. **Caso o programa solução entregue não funcione, a nota será zerada.**

Importante: Serão aceitas apenas uma listagem e apenas um arquivo de vídeo por aluno.

**Regra para o nome dos arquivos:
Erros na nomeação serão penalizados em 20%**

Siga o formato: **TP1-xxxxxxx.ext**

T = sua turma (A, B, C, ...)

P1 = Indicação de que é o “Problema 1

xxxxx = seu número de matrícula, usando apenas números

ext = extensão que indica o tipo do arquivo (asm, c, avi, mp4, ...)

Escreva na primeira linha do seu arquivo de código (linha de número 1) sua matrícula e seu nome completo.

ÉTICA E HONESTIDADE ESTUDANTIL:

Será verificada a similaridade entre os programas entregues e os demais programas, incluindo os dos semestres anteriores. Caso a similaridade entre dois programas seja grande o suficiente para caracterizar a “cola”, as equipes estão **automaticamente reprovadas**.

Este problema trabalha com a ideia do cifrador Enigma, usado pelos alemães durante a segunda guerra mundial. Para facilitar o entendimento deste visto, faremos abaixo uma rápida apresentação deste cifrador, seguida por 5 experimentos simples (não devem ser entregues) que auxiliam na compreensão dos conceitos e ao final apresentamos o pedido de forma completa. Vamos usar um modelo simplificado do Enigma. Uma explicação mais completa está disponível no Apêndice E da fonte de consulta.



O Enigma original trabalha apenas com 26 letras maiúsculas: A, B, C, ..., Z. Não tem caractere de espaço, números ou qualquer outro símbolo. A cifragem do Enigma faz a troca entre essas 26 letras. O segredo está na grande possibilidade de trocas.



Foto do Enigma e detalhes do rotor.

Como pode ser visto na figura acima, o rotor é um cilindro com 26 contatos (26 letras) em cada extremidade e um padrão de permutação (conexão por fios) entre os contatos nessas extremidades. Diferentes padrões de contatos geram diferentes cilindros. Então, o que ele faz é uma permutação entre as letras (entre os contatos). A figura abaixo apresenta uma ilustração onde o cilindro é apresentado de forma linear. Nesta figura temos as trocas $A \rightarrow G$, $B \rightarrow E$, $C \rightarrow T$, ..., $Z \rightarrow J$.

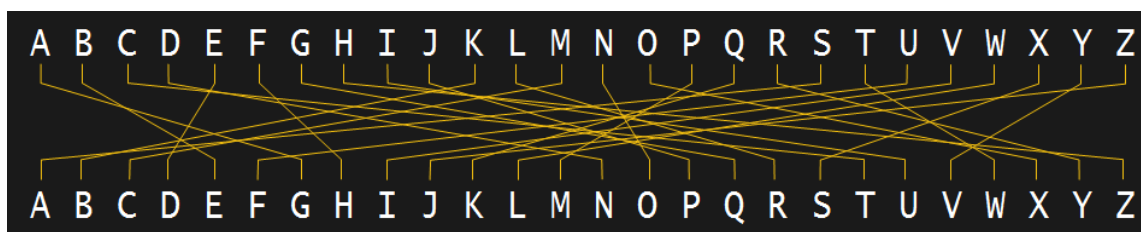


Ilustração de um rotor e a permuta entre as 26 letras realizada por uma série de condutores (linhas amarelas).

(Fonte: hackaday.com/2017/08/22/the-enigma-enigma-how-the-enigma-machine-worked)



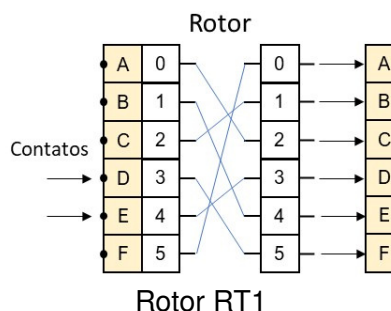
A partir deste ponto iniciamos os conceitos do Enigma simplificado, com o uso de pequenos experimentos que trabalham com um modelo hipotético de apenas 6 letras (A, B, ..., F). Usamos apenas 6 letras para facilitar a visualização das explicações.

Experimento 1: Entendendo um Rotor

Como ilustrado ao lado, propomos um rotor de 6 posições, numeradas de 0 até 5. É fácil de ver que estão programadas (pelos condutores internos) as trocas: A(0) → C(2), B(1) → E(4), ..., F (5) → A (0). Este rotor será denominado por RT1. Podemos representá-lo pelo vetor, que indica o final de cada “caminho”.

RT1 = { 2, 4, 1, 5, 3, 0 }

É importante que fique clara a ideia de que o rotor é caracterizado pelo seu padrão de contatos. Um padrão diferente origina um novo rotor.



Problema: Escreve a sub-rotina “ENIGMA1” que recebe a mensagem “msg” em claro e a armazena cifrada no vetor “gsm”, usando o rotor RT1.

```
EXP1:      MOV      #MSG,R5
           MOV      #GSM,R6
           CALL     #ENIGMA1
           JMP      $
           NOP

;
; Sua rotina ENIGMA (Exp 1)
ENIGMA1:   ...
;
; Dados para o enigma
           .data
MSG:       .byte    "CABECAFEFACAFAD",0      ;Mensagem em claro
GSM:       .byte    "XXXXXXXXXXXXXXXX",0      ;Mensagem cifrada
RT1:       .byte    2, 4, 1, 5, 3, 0          ;Trama do Rotor 1
```

Resposta: BCEDBCADACBCACF



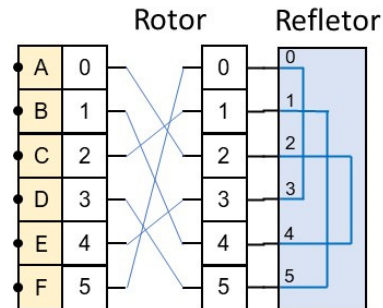
Experimento 2: Entendendo um Refletor

O Enigma é um cifrador simétrico, ou seja, a mensagem em claro gera a cifrada e a mensagem cifrada gera a mensagem em claro. Para isso, se faz necessário um recurso denominado Refletor, que está ilustrado ao lado. Como o nome diz, ele retorna o sinal. O refletor proposto, denominado de RF1 é representado pelo vetor:

RF1 = { 3, 5, 4, 0, 2, 1 }

O conjunto RT1 e RF1 faz as permutações:

$A \leftrightarrow B, C \leftrightarrow D, E \leftrightarrow F.$



Problema: Escreva a sub-rotina “ENIGMA2” que recebe a mensagem “msg” em claro e a armazena cifrada no vetor “gsm”, usando o rotor RT1 e o refletor RF1. Alerta, sua solução terá dois acessos ao rotor RT1, o primeiro é direto e o segundo é na forma inversa. Pense nisso! Após a cifragem, comprove sua solução apresentando a mensagem cifrada para ver se ela retorna a mensagem em claro.

```
EXP2:      MOV      #MSG, R5
           MOV      #GSM, R6
           CALL     #ENIGMA2      ;Cifrar
           ;
           MOV      #GSM, R5
           MOV      #DCF, R6
           CALL     #ENIGMA2      ;Decifrar
           JMP      $
           NOP

;
; Sua rotina ENIGMA (Exp 2)
ENIGMA2:   ...
;
; Dados para o enigma
           .data
MSG:       .byte    "CABECAFEFACAFAD", 0      ;Mensagem em claro
GSM:       .byte    "XXXXXXXXXXXXXXXX", 0      ;Mensagem cifrada
DCF:       .byte    "XXXXXXXXXXXXXXXX", 0      ;Mensagem decifrada
RT1:       .byte    2, 4, 1, 5, 3, 0          ;Trama do Rotor
RF1:       .byte    3, 5, 4, 0, 2, 1          ;Tabela do Refletor
```

Resposta: DBAFDBEFEBDBEBC



Experimento 3: Entendendo a Configuração de um Rotor

Para aumentar a quantidade de possibilidades do rotor, o anel com as letras pode ser girado sobre o padrão de contatos, resultando no que vamos chamar de **Configuração**. Na figura abaixo, temos a representação simplificada do rotor RT1 com suas 6 possíveis configurações. Note que estamos convencionando que a Configuração é dada pelo número que corresponde à letra A (número à direita da letra A).

<table><tr><td>A</td><td>0</td><td>2</td></tr><tr><td>B</td><td>1</td><td>4</td></tr><tr><td>C</td><td>2</td><td>1</td></tr><tr><td>D</td><td>3</td><td>5</td></tr><tr><td>E</td><td>4</td><td>3</td></tr><tr><td>F</td><td>5</td><td>0</td></tr></table> <p>Config = 0</p>	A	0	2	B	1	4	C	2	1	D	3	5	E	4	3	F	5	0	<table><tr><td>F</td><td>0</td><td>2</td></tr><tr><td>A</td><td>1</td><td>4</td></tr><tr><td>B</td><td>2</td><td>1</td></tr><tr><td>C</td><td>3</td><td>5</td></tr><tr><td>D</td><td>4</td><td>3</td></tr><tr><td>E</td><td>5</td><td>0</td></tr></table> <p>Config = 1</p>	F	0	2	A	1	4	B	2	1	C	3	5	D	4	3	E	5	0	<table><tr><td>E</td><td>0</td><td>2</td></tr><tr><td>F</td><td>1</td><td>4</td></tr><tr><td>A</td><td>2</td><td>1</td></tr><tr><td>B</td><td>3</td><td>5</td></tr><tr><td>C</td><td>4</td><td>3</td></tr><tr><td>D</td><td>5</td><td>0</td></tr></table> <p>Config = 2</p>	E	0	2	F	1	4	A	2	1	B	3	5	C	4	3	D	5	0	<table><tr><td>D</td><td>0</td><td>2</td></tr><tr><td>E</td><td>1</td><td>4</td></tr><tr><td>F</td><td>2</td><td>1</td></tr><tr><td>A</td><td>3</td><td>5</td></tr><tr><td>B</td><td>4</td><td>3</td></tr><tr><td>C</td><td>5</td><td>0</td></tr></table> <p>Config = 3</p>	D	0	2	E	1	4	F	2	1	A	3	5	B	4	3	C	5	0	<table><tr><td>C</td><td>0</td><td>2</td></tr><tr><td>D</td><td>1</td><td>4</td></tr><tr><td>E</td><td>2</td><td>1</td></tr><tr><td>F</td><td>3</td><td>5</td></tr><tr><td>A</td><td>4</td><td>3</td></tr><tr><td>B</td><td>5</td><td>0</td></tr></table> <p>Config = 4</p>	C	0	2	D	1	4	E	2	1	F	3	5	A	4	3	B	5	0	<table><tr><td>B</td><td>0</td><td>2</td></tr><tr><td>C</td><td>1</td><td>4</td></tr><tr><td>D</td><td>2</td><td>1</td></tr><tr><td>E</td><td>3</td><td>5</td></tr><tr><td>F</td><td>4</td><td>3</td></tr><tr><td>A</td><td>5</td><td>0</td></tr></table> <p>Config = 5</p>	B	0	2	C	1	4	D	2	1	E	3	5	F	4	3	A	5	0
A	0	2																																																																																																															
B	1	4																																																																																																															
C	2	1																																																																																																															
D	3	5																																																																																																															
E	4	3																																																																																																															
F	5	0																																																																																																															
F	0	2																																																																																																															
A	1	4																																																																																																															
B	2	1																																																																																																															
C	3	5																																																																																																															
D	4	3																																																																																																															
E	5	0																																																																																																															
E	0	2																																																																																																															
F	1	4																																																																																																															
A	2	1																																																																																																															
B	3	5																																																																																																															
C	4	3																																																																																																															
D	5	0																																																																																																															
D	0	2																																																																																																															
E	1	4																																																																																																															
F	2	1																																																																																																															
A	3	5																																																																																																															
B	4	3																																																																																																															
C	5	0																																																																																																															
C	0	2																																																																																																															
D	1	4																																																																																																															
E	2	1																																																																																																															
F	3	5																																																																																																															
A	4	3																																																																																																															
B	5	0																																																																																																															
B	0	2																																																																																																															
C	1	4																																																																																																															
D	2	1																																																																																																															
E	3	5																																																																																																															
F	4	3																																																																																																															
A	5	0																																																																																																															

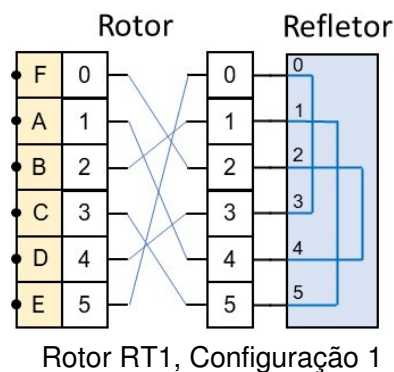
As 6 possíveis configurações do rotor RT1.

Para que a figura não ficasse grande, a trama interna do rotor foi omitida. Apresentam-se apenas a entrada e a saída. São os contatos 0-2, 1-4, ..., 5-0.

Problema: Escreva a sub-rotina “ENIGMA3” que recebe a mensagem “msg” em claro e a armazena cifrada no vetor “gsm”, usando o rotor RT1 na Configuração indicada pela constante CONF1 e o refletor RF1.

Na solução será necessário conhecer o tamanho do rotor, que é dado pela constante RT_TAM.

Para facilitar seu raciocínio, a figura ao lado apresenta com detalhes o rotor RT1 na Configuração = 1.



Comprove sua solução apresentando a mensagem cifrada para ver se ela retorna a mensagem em claro. Faça ensaios com diferentes configurações.

```
RT_TAM      .equ      6          ;Tamanho dos rotores
CONF1       .equ      1          ;Configuração do Rotor 1

EXP3:       MOV       #MSG,R5
            MOV       #GSM,R6
            CALL      #ENIGMA3
            ;
            ; Dependendo da solução, pode ser necessária uma
            ; sub-rot para restaurar posição original do rotor
            ;
```



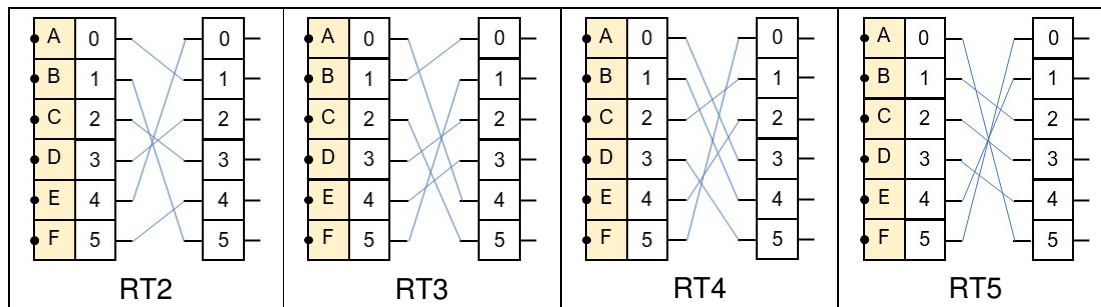
```
MOV      #GSM,R5
MOV      #DCF,R6
CALL     #ENIGMA3
JMP      $
NOP

;
; Sua rotina ENIGMA (Experimento 3)
ENIGMA3:  ...
;
; Dados para o enigma
.data
MSG:      .byte      "CABECAFEFACAFAD",0      ;Mensagem em claro
GSM:      .byte      "XXXXXXXXXXXXXXXXX",0      ;Mensagem cifrada
DCF:      .byte      "XXXXXXXXXXXXXXXXX",0      ;Mensagem decifrada
RT1:      .byte      2, 4, 1, 5, 3, 0          ;Trama do Rotor
RF1:      .byte      3, 5, 4, 0, 2, 1          ;Tabela do Refletor
```

Resposta: BFCDBFADAFBFAFE.

Experimento 4: Entendendo a Justaposição dos Rotores

Em sua forma mais típica, o Enigma tinha 5 rotores diferentes. Para sua operação, a cada dia, 3 rotores eram selecionados dentre esses 5. A figura abaixo apresenta mais 4 rotores, para perfazermos um total de 5 rotores. Veja também os vetores que caracterizam esses rotores. Lembramos que cada rotor ainda possui 6 possibilidades de configuração.



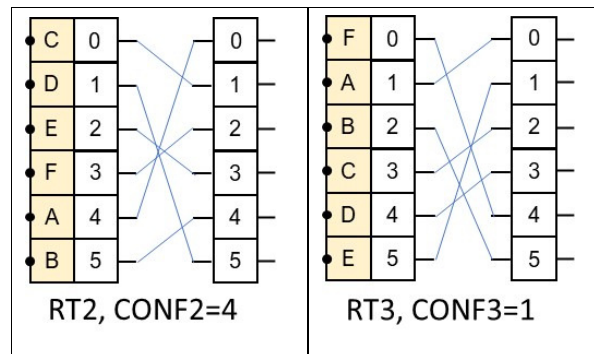
RT2 = { 1, 5, 3, 2, 0, 4 }

RT3 = { 4, 0, 5, 2, 3, 1 }

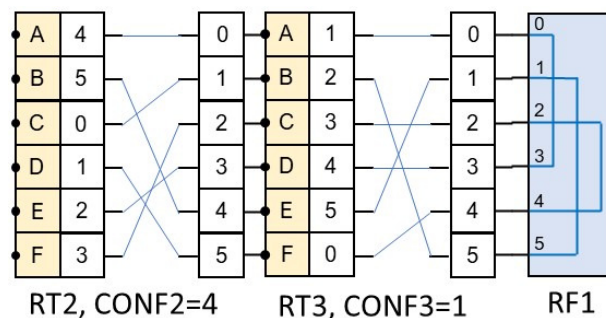
RT4 = { 3, 4, 1, 5, 2, 0 }

RT5 = { 5, 2, 3, 4, 1, 0 }

Para este experimento vamos usar os rotores RT2 e RT3, nas configurações 4 e 1 respectivamente como mostrado abaixo.



A figura abaixo apresenta o esquema deste experimento. Note que os rotores foram girados de forma que a letra “A” de cada um deles ocupasse o topo (Posição Inicial A para todos os rotores). Agora temos as seguintes permutações: $A \leftrightarrow E$, $B \leftrightarrow C$, $D \leftrightarrow F$. É importante que se vá percebendo a quantidade de permutações possíveis.



Problema: Escreva a sub-rotina “ENIGMA4” que recebe a mensagem “msg” em claro e a armazena cifrada no vetor “gsm”, usando os rotores RT2 e RT3 nas configurações 4 e 1 respectivamente, junto com o refletor RF1. Comprove também a decifragem. Veja a listagem abaixo

```
RT_TAM      .equ      6          ;Tamanho dos rotores
CONF2       .equ      4          ;Configuração do Rotor 2
CONF3       .equ      1          ;Configuração do Rotor 3

EXP4:       MOV        #MSG,R5
            MOV        #GSM,R6
            CALL        #ENIGMA4      ;Cifrar
            ;
            ; Dependendo da solução, pode ser necessária uma
            ; Sub-rot para restaurar posição original dos rotores
            ;
            MOV        #GSM,R5
            MOV        #DCF,R6
            CALL        #ENIGMA4      ;Decifrar
            JMP        $
```




```

NOP
;
; Sua rotina ENIGMA (Experimento 4)
;
ENIGMA4:
    ...
;
; Área de dados
.data
MSG:      .byte      "CABECAFEFACAFAD",0      ; Mensagem em claro
GSM:      .byte      "XXXXXXXXXXXXXXXXXX",0    ; Mensagem cifrada
DCF:      .byte      "XXXXXXXXXXXXXXXXXX",0    ; Mensagem decifrada
;
; Rotores disponíveis
ROTORES:
RT1:      .byte      2, 4, 1, 5, 3, 0 ; Trama do Rotor 1
RT2:      .byte      1, 5, 3, 2, 0, 4 ; Trama do Rotor 2
RT3:      .byte      4, 0, 5, 2, 3, 1 ; Trama do Rotor 3
RT4:      .byte      3, 4, 1, 5, 2, 0 ; Trama do Rotor 4
RT5:      .byte      5, 2, 3, 4, 1, 0 ; Trama do Rotor 5
;
; Refletores disponíveis
REFLETORES:
RF1:      .byte      3, 5, 4, 0, 2, 1 ; Tabela do Refletor 1
RF2:      .byte      4, 5, 3, 2, 0, 1 ; Tabela do Refletor 2
RF3:      .byte      3, 2, 1, 0, 5, 4 ; Tabela do Refletor 3

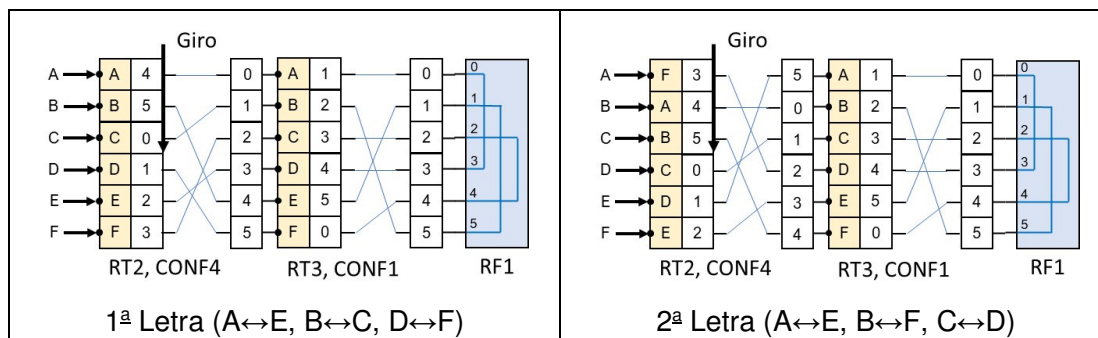
```

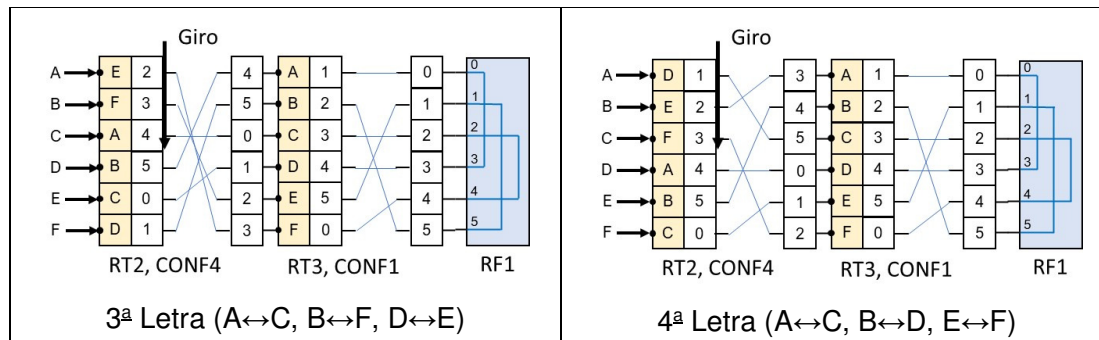
Resposta: BECABEDEBEDEF

Experimento 5: Entendendo a rotação dos Rotores

Se analisarmos com cuidado o resultado do experimento 4, vamos concluir que ele apenas trocou as letras $A \leftrightarrow E$, $B \leftrightarrow C$, $D \leftrightarrow F$. Isto é muito fácil de decifrar. Para aumentar a complexidade, os rotores giram. O rotor mais à esquerda gira um passo após a cifragem de cada letra e o rotor à direita gira um passo a cada volta completa do rotor da esquerda. No nosso caso, como temos apenas 6 letras, a cada 6 letras o rotor da esquerda dá uma volta completa e o rotor da direita um passo.

Cuidado, para não se confundir, pois o corpo do Enigma permanece fixo. A figura abaixo ilustra a posição dos rotores para cifrar as 4 primeiras letras. O padrão de contatos dos rotores foi redesenhado para facilitar a elaboração do programa.





Problema: Escreva a sub-rotina “ENIGMA5” que recebe a mensagem “msg” em claro e a armazena cifrada no vetor “gsm”, usando os rotores RT2 e RT3 nas configurações 4 e 1 respectivamente, junto com o refletor RF1. Inclua agora a rotação dos rotores.

Cuidado, antes de decifrar a mensagem, é preciso voltar os rotores para a posição inicial.

```
RT_TAM      .equ      6          ;Tamanho dos rotores
CONF2       .equ      4          ;Configuração do Rotor 2
CONF3       .equ      1          ;Configuração do Rotor 3

EXP5:        MOV        #MSG,R5
              MOV        #GSM,R6
              CALL       #ENIGMA5      ;Cifrar
              ;
              ; Agora é necessário uma
              ; Sub-rot para restaurar posição original dos rotores
              ;
              MOV        #GSM,R5
              MOV        #DCF,R6
              CALL       #ENIGMA5      ;Decifrar
              JMP        $
              NOP

;
; Sua rotina ENIGMA (Experimento 5)
;
ENIGMA5:      ...

; Dados para o enigma
.data
MSG:          .byte      "CABECAFEFACAFAD",0      ;Mensagem em claro
GSM:          .byte      "XXXXXXXXXXXXXXXXXX",0    ;Mensagem cifrada
DCF:          .byte      "XXXXXXXXXXXXXXXXXX",0    ;Mensagem decifrada

; Rotores disponíveis
ROTORES:
RT1:          .byte      2, 4, 1, 5, 3, 0      ;Trama do Rotor 1
RT2:          .byte      1, 5, 3, 2, 0, 4      ;Trama do Rotor 2
RT3:          .byte      4, 0, 5, 2, 3, 1      ;Trama do Rotor 3
RT4:          .byte      3, 4, 1, 5, 2, 0      ;Trama do Rotor 4
RT5:          .byte      5, 2, 3, 4, 1, 0      ;Trama do Rotor 5
```



```
; Refletores disponíveis  
REFLETORES:  
RF1:      .byte      3, 5, 4, 0, 2, 1 ;Tabela do Refletor 1  
RF2:      .byte      4, 5, 3, 2, 0, 1 ;Tabela do Refletor 2  
RF3:      .byte      3, 2, 1, 0, 5, 4 ;Tabela do Refletor 3
```

Resposta: BEFFEBDABCEBDEE

Observação: O Enigma original, além de girar os rotores a cada letra, oferece ainda flexibilidade para especificar a Posição Inicial dos Rotores e um Painel de Plugs para fazer até 10 trocas entre duas letras. A isso tudo foram adicionados um teclado e um painel de letras para permitir seu uso. Eram usados 3 rotores escolhidos entre um total de 5 e 1 refletor escolhido entre 3 disponíveis. O total de possibilidades é astronômico, pois chega a 158.962.555.217.826.360.000 ou, aproximadamente, $1,5 \times 10^{20}$. Imaginando que você consiga testar 1 Giga possibilidades por segundo, quanto tempo você demoraria, na pior das hipóteses, para decifrar uma mensagem? A chave era trocada todo dia, à meia-noite. Como será que em apenas 24 horas os ingleses, várias vezes, conseguiram decifrar mensagens cifradas pelo Enigma?



Proposta para o problema a ser resolvido

Construir o programa Enigma com rotores de 32 símbolos, de acordo com a tabela abaixo. Todo resto, ou seja, espaços, pontuação, números etc., deve ser preservado.

ASCII	3B	3C	3D	3E	3F	40	41	42	43	...	59	5A
Símbolos	;	<	=	>	?	@	A	B	C	...	Y	Z

Temos a disponibilidade de 5 rotores e 3 refletores. Nosso Enigma vai usar 3 rotores justapostos, denominados: rotor da esquerda, rotor central e rotor da direita, e um refletor. Os rotores giram a cada letra cifrada. O vetor CHAVE especifica a configuração do cifrador, da forma descrita abaixo:

CHAVE: **.byte A, B, C, D, E, F, G**

A = número do rotor à esquerda e B = sua configuração;

C = número do rotor central e D = sua configuração;

E = número do rotor à direita e F = sua configuração;

G = número do refletor.

```
;-----  
; Main loop here  
;-----  
RT_TAM           .equ   32           ;Tamanho dos rotores (32 simbolos)  
  
VISTO1:           MOV           #MSG,R5  
                  MOV           #GSM,R6  
                  CALL          #ENIGMA       ;Cifrar  
                  ;  
                  CALL          #RESETE       ;Restaurar posição original  
                  ;  
                  MOV           #GSM,R5  
                  MOV           #DCF,R6  
                  CALL          #ENIGMA       ;Cifrar  
                  JMP          $  
                  NOP  
  
;  
; Definição da chave do Enigma  
CHAVE:           .byte 2, 6, 3, 10, 5, 12, 2  
;  
; Coloque aqui sua sub-rotina ENIGMA  
ENIGMA: ...  
                  ;  
;  
; Área de dados  
                  .data  
MSG:            .byte "UMA NOITE DESTAS, VINDO DA CIDADE PARA O ENGENHO NOVO,"  
                  .byte " ENCONTREI NO TREM DA CENTRAL UM RAPAZ AQUI DO BAIRRO,"
```



```
.byte " QUE EU CONHECO DE VISTA E DE CHAPEU.",0 ;Don Casmurro

GSM: .byte "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
      .byte "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
      .byte "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX",0
DCF: .byte "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
      .byte "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
      .byte "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX",0

;Rotores com 32 posições
ROTORES:
RT1: .byte 13, 23, 0, 9, 4, 2, 5, 11, 12, 17, 21, 6, 28, 25, 30, 10
      .byte 22, 1, 3, 26, 24, 31, 8, 14, 29, 15, 18, 16, 19, 7, 27, 20
RT2: .byte 6, 24, 2, 8, 25, 20, 16, 29, 23, 0, 7, 19, 30, 17, 12, 15
      .byte 5, 4, 26, 10, 11, 18, 28, 27, 14, 9, 13, 1, 21, 31, 22, 3
RT3: .byte 6, 15, 23, 7, 27, 13, 19, 3, 16, 4, 17, 20, 24, 25, 0, 10
      .byte 30, 26, 22, 1, 8, 11, 14, 31, 9, 28, 5, 18, 12, 2, 29, 21
RT4: .byte 15, 16, 5, 18, 31, 26, 19, 28, 1, 2, 14, 12, 24, 20, 21, 0
      .byte 11, 23, 4, 10, 7, 3, 25, 29, 27, 8, 17, 6, 9, 13, 22, 30
RT5: .byte 13, 25, 1, 26, 6, 12, 9, 2, 28, 11, 16, 15, 4, 8, 3, 31
      .byte 5, 18, 23, 17, 24, 27, 0, 22, 29, 19, 7, 10, 14, 21, 20, 30

;Refletores com 32 posições
REFLETORES:
RF1: .byte 26, 23, 31, 9, 29, 20, 16, 11, 27, 3, 14, 7, 21, 28, 10, 25
      .byte 6, 22, 24, 30, 5, 12, 17, 1, 18, 15, 0, 8, 13, 4, 19, 2
RF2: .byte 20, 29, 8, 9, 23, 27, 21, 11, 2, 3, 25, 7, 13, 12, 22, 16
      .byte 15, 28, 30, 26, 0, 6, 14, 4, 31, 10, 19, 5, 17, 1, 18, 24
RF3: .byte 14, 30, 7, 5, 15, 3, 18, 2, 23, 17, 29, 28, 25, 27, 0, 4
      .byte 19, 9, 6, 16, 26, 22, 21, 8, 31, 12, 20, 13, 11, 10, 1, 24
```

Resposta :

```
UMA NOITE DESTAS, VINDO DA CIDADE PARA O ENGENHO NOVO,
OWB >?>M> PQKL;@, JRGWC PF <<IYTU J>LX S =@??YWG BCHG,

ENCONTREI NO TREM DA CENTRAL UM RAPAZ AQUI DO BAIRRO,
CH@W<NZHF LM Z>WI OR WIRIC@C FO XVREW PBWT CN SLXSC=,

QUE EU CONHECO DE VISTA E DE CHAPEU.
FFK ;J UIK?V<< YZ DBVWG L BC N;V;VN.
```



Tabela a ser usada na correção

Pontos	Configuração	Rotação	Análogo aos Experimentos	
40%	Não	Não	1	Só consulta o rotor
60%	Não	Não	1, 2	Rotor + Refletor
80%	Sim	Não	1, 2, 3	Rotor com configuração inicial + Refletor
90%	Sim	Não	1, 2, 3, 4	Com dois rotores
100%	Sim	Sim	1, 2, 3, 4, 5	Completo

Observação: Os experimentos numerados de 1 até 5 não devem ser entregues. Eles não “valem nota”. A tabela acima faz referência a eles apenas para tornar clara a possibilidade de resolução parcial do problema.

→ → → Sensacional desafio ← ← ←

O primeiro aluno que construir um programa em assembly para o MSP430 capaz de descobrir a chave usada na cifração de uma mensagem qualquer, recebe 200% de pontuação. Abaixo está um exemplo para verificação. Qual a chave usada para cifrar esta mensagem? Não são aceitos programas feitos em C ou que rodem em computador. A execução precisa ser no MSP430.

```
UMA NOITE DESTAS, VINDO DA CIDADE PARA O ENGENHO NOVO,  
<UJ CD;@J =>YZHW, M=SNE T; OS>GYZ WNY< A ;I=DA<U @AQE,  
  
ENCONTREI NO TREM DA CENTRAL UM RAPAZ AQUI DO BAIRRO,  
;MBPCHS>H DE R<@X WI UJXUL=S OL PJOGS =NTM BK <KELCE,  
  
QUE EU CONHECO DE VISTA E DE CHAPEU.  
ZFC K> DSGOBDN TU FHTUG = YZ LBXMKT.
```

Para facilitar seu programa:

```
GSM: .byte <UJ CD;@J =>YZHW, M=SNE T; OS>GYZ WNY< A ;I=DA<U @AQE,  
.byte ;MBPCHS>H DE R<@X WI UJXUL=S OL PJOGS =NTM BK <KELCE,  
.byte ZFC K> DSGOBDN TU FHTUG = YZ LBXMKT.
```

Quanto tempo você demorou?



TABELA ASCII PADRÃO

	0	1	2	3	4	5	6	7
0	NUL	DLE	SP	0	@	P	`	p
1	SOH	DC1	!	1	A	Q	a	q
2	STX	DC2	"	2	B	R	b	r
3	ETX	DC3	#	3	C	S	c	s
4	EOT	DC4	\$	4	D	T	d	t
5	ENQ	NAK	%	5	E	U	e	u
6	ACK	SYN	&	6	F	V	f	v
7	BEL	ETB	'	7	G	W	g	w
8	BS	CAN	(8	H	X	h	x
9	HT	EM)	9	I	Y	i	y
A	LF	SUB	*	:	J	Z	j	z
B	VT	ESC	+	;	K	[k	{
C	FF	FS	,	<	L	\	l	
D	CR	GS	-	=	M]	m	}
E	SO	RS	.	>	N	^	n	~
F	SI	UP	/	?	O	_	o	DEL

Exemplos: símbolo 'A' = código ASCII 41h
símbolo 'B' = código ASCII 42h
símbolo 'a' = código ASCII 61h
símbolo 'z' = código ASCII 7Ah

Códigos especiais usados na Tabela ASCII

Hexa	ASCII	Significado
00	NUL	NULL
01	SOH	Start Of Heading
02	STX	Start of TeXt
03	ETX	End of TeXt
04	EOT	End Of Transmission
05	ENQ	ENQuire
06	ACK	ACKnowledge
07	BEL	BELL
08	BS	Back Space
09	HT	Horizontal Tab
0A	LF	Line Feed
0B	VT	Vertical Tab
0C	FF	Form Feed
0D	CR	Carriage Return
0E	SO	Shift Out
0F	SI	Shift In
7F	DEL	DELete"

Hexa	ASCII	Significado
10	DLE	Data Link Escape
11	DC1	Device Control 1
12	DC2	Device Control 2
13	DC3	Device Control 3
14	DC4	Device Control 4
15	NAK	Negative AcKnowledge
16	SYN	SYNchronism idle
17	ETB	End of Transmission Block
18	CAN	CANcel
19	EM	End of Medium
1A	SUB	SUBstitute
1B	ESC	ESCape
1C	FS	File Separator
1D	GS	Group Separator
1E	RS	Record Separator
1F	UP	Unit Separator
20	SP	SPace