(Visto 1) PROBLEMA 1: Cifrador Enigma

A solução deste problema pode ser individual ou em dupla de laboratório. Cada aluno/dupla deverá apresentar sua solução ao professor durante a aula de laboratório, que fará perguntas para verificar o entendimento de cada aluno. A listagem do programa solução deverá ser enviada no mesmo dia. Esta listagem 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 corresponda ao que foi apresentado, a nota será zerada**.

Importante: Será aceita apenas uma listagem por aluno/dupla.

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

Siga o formato: **TP1-xxxxxxxx.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)

Escreva na primeira linha do seu arquivo de código (linha de número 1) suas matrículas e seus nomes completos.
Na segunda linha a pontuação pretendida.

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", os alunos/duplas serão **automaticamente reprovados**.

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 → G, B → E, C → T, ..., Z → J.

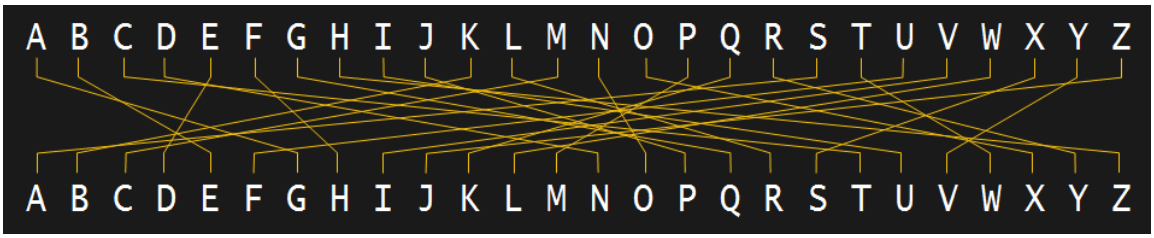


Ilustração de um rotor e a permuta entre as 26 letras realizada por uma série de condutores (linhas amarelas).
(Fonte: hackaday.com/2017/08/22/the-enigma-enigma-how-the-enigma-machine-worked)

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

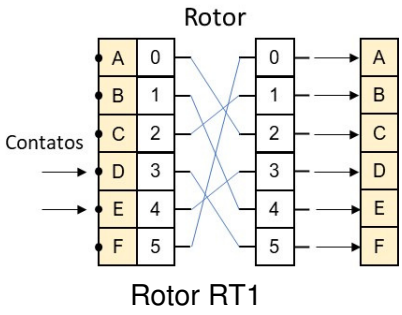
Experimento 1: Entendendo um Rotor

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.



Rotor RT1

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       "XXXXXXXXXXXXXXX",0      ;Mensagem cifrada
RT1:            .byte       2, 4, 1, 5, 3, 0         ;Trama do Rotor 1
```

Resposta: BCEDBCADACBCACF

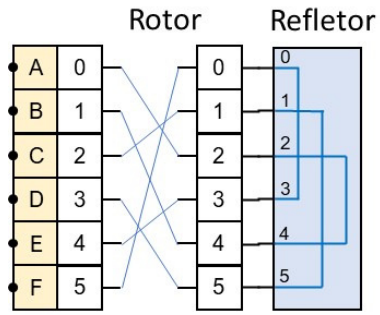
Experimento 2: Entendendo um Refletor

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 ↔ B, C ↔ D, E ↔ F.



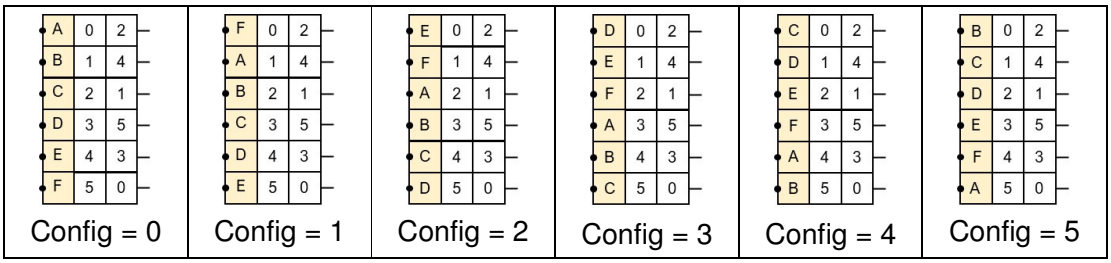
Problema: Escreve a sub-rotina "ENIGMA2" que recebe a mensagem "msg" em claro e a armazena cifrada no vetor "gsm", usando o rotor RT1 e o refletor RF1. Alerta, sua solução terá dois acessos ao rotor RT1, o primeiro é direto e o segundo é na forma inversa. Pense nisso! Após a cifragem, comprove sua solução 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       "XXXXXXXXXXXXXXX",0     ;Mensagem cifrada
DCF:        .byte       "XXXXXXXXXXXXXXX",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).

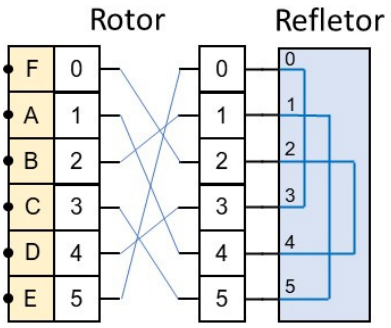


As 6 possíveis configurações do rotor RT1.
Para que a figura não ficasse grande, a trama interna do rotor foi omitida. Apresentam-se apenas a entrada e a saída. São os contatos 0-2, 1-4, ..., 5-0.

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.



Rotor RT1, Configuração 1

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

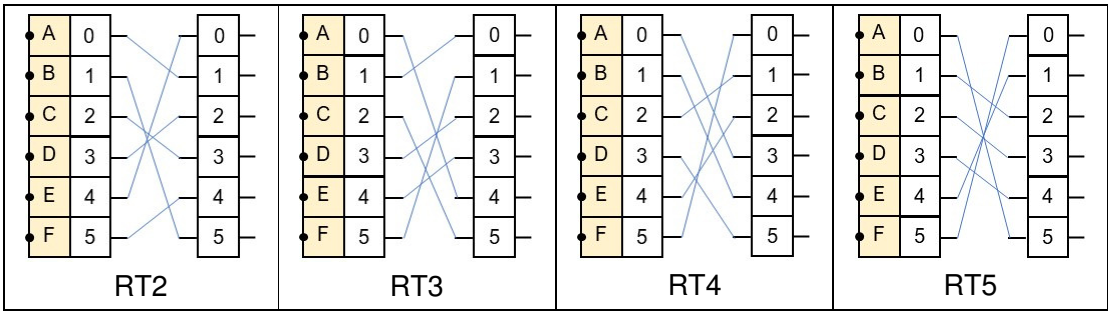
```
RT_TAM          .equ        6           ;Tamanho dos rotores
CONF1           .equ        1           ;Configuração do Rotor 1

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       "XXXXXXXXXXXXXXX",0     ;Mensagem cifrada
DCF:          .byte       "XXXXXXXXXXXXXXX",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.

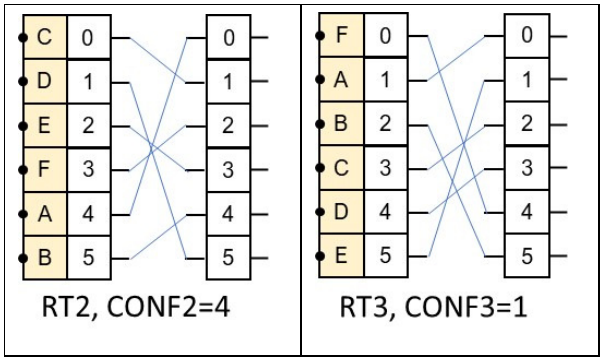
Experimento 4: Entendendo a Justaposição dos Rotores

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.

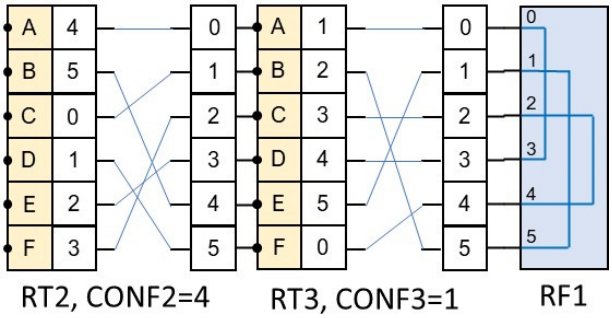


RT2 = { 1, 5, 3, 2, 0, 4 } RT3 = { 4, 0, 5, 2, 3, 1 }
RT4 = { 3, 4, 1, 5, 2, 0 } RT5 = { 5, 2, 3, 4, 1, 0 }

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 ↔ E, B ↔ C, D ↔ 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             $
```

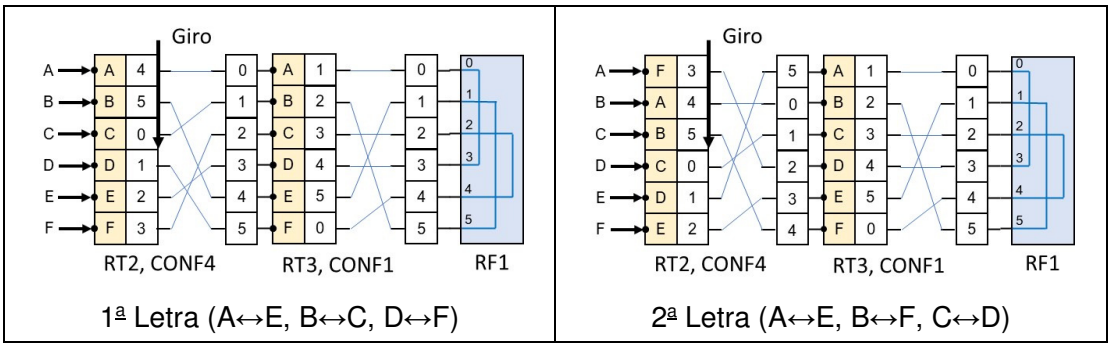
```
                    NOP
;
; Sua rotina ENIGMA (Experimento 4)
;
ENIGMA4:                ...
;
; Área de dados
            .data
MSG:        .byte       "CABECAFEFACAFAD",0     ;Mensagem em claro
GSM:        .byte       "XXXXXXXXXXXXXXX",0     ;Mensagem cifrada
DCF:        .byte       "XXXXXXXXXXXXXXX",0     ;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: BECABEDADEBEDEF

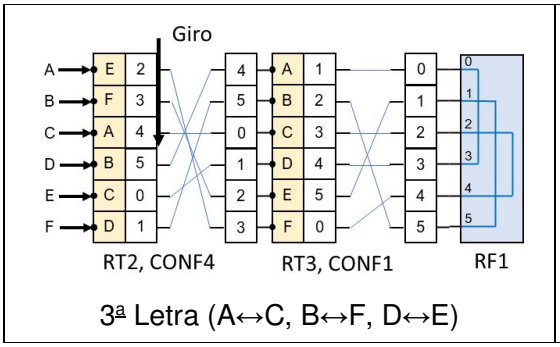
Experimento 5: Entendendo a rotação dos Rotores

Se analisarmos com cuidado o resultado do experimento 4, vamos concluir que ele apenas trocou as letras A ↔ E, B ↔ C, D ↔ 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.

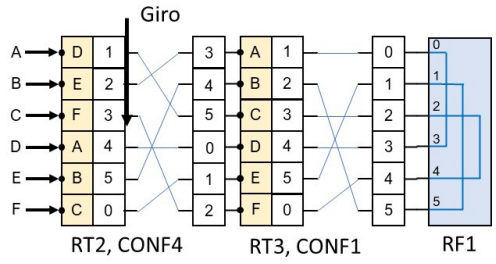
Cuidado, para não se confundir, pois o corpo do Enigma permanece fixo. A figura abaixo ilustra a posição dos rotores para cifrar as 4 primeiras letras. O padrão de contatos dos rotores foi redesenhado para facilitar a elaboração do programa.



1ª Letra (A↔E, B↔C, D↔F) | 2ª Letra (A↔E, B↔F, C↔D)



3ª Letra (A↔C, B↔F, D↔E)



4ª Letra (A↔C, B↔D, E↔F)

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       "XXXXXXXXXXXXXXX",0     ;Mensagem cifrada
DCF:        .byte       "XXXXXXXXXXXXXXX",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 26 símbolos, de acordo com a tabela abaixo. Todo resto, ou seja, espaços, pontuação, números etc., deve ser preservado.

ASCII (hexa)	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 (RESQ), rotor do meio ou central (RMEIO) e rotor da direita (RDIR), e um refletor (REF). 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.
```

Abaixo está a mensagem a ser cifrada, que tem 160 caracteres incluindo o zero final.
São 129 caracteres a serem cifrados e 31 caracteres a serem ignorados.

```
           1            2           3            4
123 45678 901234  56789 01 234567 8901 2 3456789 0123
UMA NOITE DESTAS, VINDO DA CIDADE PARA O ENGENHO NOVO,

       5            6            7           8
 456789012 34 5678 90 1234567 89 01234 5678 90 123456
 ENCONTREI NO TREM DA CENTRAL UM RAPAZ AQUI DO BAIRRO,

     9           0             1         2
 789 01 2345678 90 12345 6 78 901234 67890123456789
 QUE EU CONHECO DE VISTA E DE CHAPEU.@MACHADO\ASSIS
```

"Esqueleto para sua solução" (veja o arquivo disponibilizado: V1_Esq.asm)

```
;-------------------------------------------------------------------------------
; Main loop here
;-------------------------------------------------------------------------------

VISTO1:           MOV         #MSG_CLARA,R5
                  MOV         #MSG_CIFR,R6
                  CALL        #ENIGMA           ;Cifrar
                  ;
                  CALL        #RESETE           ;Restaurar posição original
                  ;
                  MOV         #MSG_CIFR,R5
```

```
                    MOV         #MSG_DECIFR,R6
                    CALL        #ENIGMA                 ;Decifrar
                    JMP         $
                    NOP

;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
; Coloque aqui sua sub-rotina ENIGMA %
;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
ENIGMA: ...
                    ;
                    ;

********************************
******    Área de dados    ******
********************************
            .data

; Chave = 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.
;               A B   C D    E F   G
CHAVE:      .word 2,4,  5,8,   3,3,  2          ;<<<===========

*********************************************
*** Área dos dados do Enigma. Não os altere ***
*********************************************
RT_TAM:     .word 26            ;Tamanho
RT_QTD:     .word 05            ;Quantidade de Rotores
RF_QTD:     .word 03            ;Quantidade de Refletores

VAZIO:      .space 12           ;Para facilitar endereço do rotor 1

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

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

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; Área da mesagem em claro, cifrada e decifrada ;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

MSG_CLARA:
      .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.@MACHADO\ASSIS",0

MSG_CIFR:
      .byte "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
      .byte "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
      .byte "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX",0

MSG_DECIFR:
      .byte "ZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ"
      .byte "ZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ"
      .byte "ZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ",0

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; Coloque aqui suas Variáveis ;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
```

Com a intenção de facilitar seu trabalho de debug da solução, a tabela abaixo apresenta os endereços das posições de memória mais importantes. É claro que seu programa NÃO PODE USAR DIRETAMENTE ESSES VALORES. Eles devem ser calculados durante a execução usando os labels disponíveis.

Label	**Endereço**		**Label**	**Endereço**
RT1	0x2420		CHAVE	0x2400
RT2	0x243A			
RT3	0x2454		MSG_CLARA	0x24F0
RT4	0x246E		MSG_CIFR	0x2590
RT5	0x2488		MSG_DECIFR	0x2630
RF1	0x24A2			
RF2	0x24BC			
RF3	0x24D6			

Tabela a ser usada na correção
(rotores e refletores com 26 posições)

Nome	Pontos	Configu-ração	Rotação	Análogo aos Experimentos	
Sol1	40%	Não	Não	1	Só consulta o rotor da direita
Sol2	60%	Não	Não	1, 2	Rotor da direita + refletor
Sol3	80%	Sim	Não	1, 2, 3	Rotor da direita com configuração inicial + refletor
Sol4	90%	Sim	Não	1, 2, 3, 4	Com rotores do meio e da direita com configuração + refletor
Sol5	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.

Respostas com a chave: **2,4, 5,8, 3,3, 2**

Solução 1 (só com o rotor da direita: RT3), apenas cifrar:

```
UMA NOITE DESTAS, VINDO DA CIDADE PARA O ENGENHO NOVO,
MHX YFDBP CPWBXW, JDYCF CX SDCXCP KXUX F PYAPYZF YFJF,

 ENCONTREI NO TREM DA CENTRAL UM RAPAZ AQUI DO BAIRRO,
 PYSFYBUPD YF BUPH.CX.SPYBUXR.MH.UXKXN.XLMD.CF.VXDUUF,

 QUE EU CONHECO DE VISTA E DE CHAPEU.@MACHADO\ASSIS
 LMP PM SFYZPSF CP JDWBX P CP SZXKPM.@HXSXCF\XWWDW
```

Solução 2 (só com o rotor da direita: RT3 e o refletor: RF2), cifrar e decifrar:

```
UMA NOITE DESTAS, VINDO DA CIDADE PARA O ENGENHO NOVO,
YAM ONHGS WSEGME, ZHOWN WM QHWMWS LMBM N SOTSOIN ONZN,

 ENCONTREI NO TREM DA CENTRAL UM RAPAZ AQUI DO BAIRRO,
 SOQNOGBSH ON GBSA WM QSOGBMP YA BMLMV MCYH WN RMHBBN,

 QUE EU CONHECO DE VISTA E DE CHAPEU.@MACHADO\ASSIS
 CYS SY QNOISQN WS ZHEGM S WS QIMLSY.@AMQIMWN\MEEHE
```

Solução 3 (só com o rotor da direita:RT3 configurado e o refletor:RF2), cifrar e decifrar:

```
UMA NOITE DESTAS, VINDO DA CIDADE PARA O ENGENHO NOVO,
HIT ZYMAF QFWATW, RMZQY QT GMQTQF BTVT Y FZCFZUY ZYRY,

 ENCONTREI NO TREM DA CENTRAL UM RAPAZ AQUI DO BAIRRO,
 FZGYZAVFM ZY AVFI QT GFZAVTK HI VTBTN TDHM QY PTMVVY,

 QUE EU CONHECO DE VISTA E DE CHAPEU.@MACHADO\ASSIS
 DHF FH GYZUFGY QF RMWAT F QF GUTBFH.@ITGUTQY\TWWMW
```

Solução 4 (rotores do meio:RT5 e da direita:RT3 configurados e o refletor:RF2), cifrar e decifrar:

```
UMA NOITE DESTAS, VINDO DA CIDADE PARA O ENGENHO NOVO,
WTL SCEMI BINMLN, RESBC BL OEBLBI JLVL C ISFISYC SCRC,

 ENCONTREI NO TREM DA CENTRAL UM RAPAZ AQUI DO BAIRRO,
 ISOCSMVIE SC MVIT BL OISMVLA WT VLJLX LKWE BC DLEVVC,

 QUE EU CONHECO DE VISTA E DE CHAPEU.@MACHADO\ASSIS
 KWI IW OCSYIOC BI RENML I BI OYLJIW.@TLOYLBC\LNNEN
```

Solução 5 – (rotores: esquerda+meio+direita configurados e girando + o refletor), cifrar e decifrar:

```
UMA NOITE DESTAS, VINDO DA CIDADE PARA O ENGENHO NOVO,
XND ZAKUB HIZAZQ, OFKCC PW VHZEQG KOMV T SLUDMCP YZTC,

ENCONTREI NO TREM DA CENTRAL UM RAPAZ AQUI DO BAIRRO,
SMHPDVSCN HI AMYS CM IPOKFUK IN LRQPY ZCVN OR IZLHLL,

 QUE EU CONHECO DE VISTA E DE CHAPEU.@MACHADO\ASSIS
 DOU CH WGQJHMB JK CCIJJ Q ST RYQZUP.@DONXKOD\RBHTA
```

➔ ➔ ➔ Sensacional desafio ← ← ←

O primeiro aluno que construir um programa em assembly para o MSP430 capaz de descobrir a chave usada na cifragem 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, Python ou que rodem em computador. A execução precisa ser no MSP430.

```
UMA NOITE DESTAS, VINDO DA CIDADE PARA O ENGENHO NOVO,
GSR HICBK VWABTO, CYTRU XO JWRUJF MLOZ D PPRFHGW ABXZ,

 ENCONTREI NO TREM DA CENTRAL UM RAPAZ AQUI DO BAIRRO,
 RORJCZMGX YZ INYQ VJ IRLGVLD YU LNXLR SZCE BR EKYDCE,

 QUE EU CONHECO DE VISTA E DE CHAPEU.@MACHADO\ASSIS
 AFO CE NDBJHYY IJ YTCDV X MN QWUQLW.@BKQBMJC\ZNION
```

Para facilitar seu programa:

```
MSG_CIFR:
   .byte "GSR HICBK VWABTO, CYTRU XO JWRUJF MLOZ D PPRFHGW ABXZ"
   .byte "RORJCZMGX YZ INYQ VJ IRLGVLD YU LNXLR SZCE BR EKYDCE,"
   .byte "AFO CE NDBJHYY IJ YTCDV X MN QWUQLW.@BKQBMJC\ZNION"
```

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	**NUL**L
01	SOH	**S**tart **O**f **H**eading
02	STX	**S**tart of **TeX**t
03	ETX	**E**nd of **TeX**t
04	EOT	**E**nd **O**f **T**ransmission
05	ENQ	**ENQ**uire
06	ACK	**ACK**nowledge
07	BEL	**BEL**L
08	BS	**B**ack **S**pace
09	HT	**H**orizontal **T**ab
0A	LF	**L**ine **F**eed
0B	VT	**V**ertical **T**ab
0C	FF	**F**orm **F**eed
0D	CR	**C**arriage **R**eturn
0E	SO	**S**hift **O**ut
0F	SI	**S**hift **I**n
7F	DEL	**DEL**ete"

Hexa	ASCII	Significado
10	DLE	**D**ata **L**ink **E**scape
11	DC1	**D**evice **C**ontrol **1**
12	DC2	**D**evice **C**ontrol **2**
13	DC3	**D**evice **C**ontrol **3**
14	DC4	**D**evice **C**ontrol **4**
15	NAK	**N**egative **A**c**K**nowledge
16	SYN	**SYN**cronism idle
17	ETB	**E**nd of **T**ransmission **B**lock
18	CAN	**CAN**cel
19	EM	**E**nd of **M**edium
1A	SUB	**SUB**stitute
1B	ESC	**ESC**ape
1C	FS	**F**ile **S**eparator
1D	GS	**G**roup **S**eparator
1E	RS	**R**ecord **S**eparator
1F	UP	**U**nit **S**eparator
20	SP	**SP**ace