```
;================================================================================
=
;        R - T Y P E
;================================================================================
=

;================================================================================
=
;       GRUPO 34:
;       86408 - Diogo Fernandes
;       86505 - Ricardo Velhinho
;================================================================================
=

;================================================================================
=
;CONSTANTES
;================================================================================
=

Ini_cursor              EQU             FFFFh
Cursor                  EQU             FFFCh
Escrita                 EQU             FFFEh
Escrita_LCD     EQU     FFF5h
Cursor_LCD      EQU     FFF4h
Inic_SP                 EQU             FDFFh
Mascara                 EQU             FFFAh
Masc_inic               EQU             4000h
Masc_jogo               EQU             C03Fh
Masc_final              EQU             FFFFh
COM_Tempo               EQU             FFF7h
Int_temp                EQU             FFF6h
Display7seg             EQU             FFF0h
LED                             EQU     FFF8h
Ult_pos                 EQU             174Eh
Lim_dir                 EQU             004Eh
Lim_esq                 EQU             0000h
Lim_sup                 EQU             0000h
Lim_inf                 EQU             1700h
Loc_esc1                EQU             0C23h
Loc_esc2                EQU             0E20h
Loc_esc3                EQU             0C22h
Loc_esc4                EQU             0024h
sem_pos_tiro    EQU     0001h
sem_pos_obs             EQU             0000h
Linha23                 EQU             1700h
Linha0                  EQU             0000h
ConvASCII               EQU             '0'
pos_nave_ini    EQU             0503h


;================================================================================
=
;INTERRUPCOES
;================================================================================
=

                                        ORIG    FE00h
INT_0                   WORD    baixo_0
INT_1                   WORD    cima_0
INT_3                   WORD    esquerda_0
INT_2                   WORD    direita_0
```



```
INT_4                   WORD    tiro_0
INT_5                   WORD    pausa_0
INT_6                   WORD    reinicia_0
INT_7                   WORD    reinicia_0
INT_8                   WORD    reinicia_0
INT_9                   WORD    reinicia_0
INT_A                   WORD    reinicia_0
INT_B                   WORD    reinicia_0
INT_C                   WORD    reinicia_0
INT_D                   WORD    reinicia_0
INT_E                   WORD    inicio_0
INT_15                  WORD    Contador

;================================================================================
=
;STRINGS
;================================================================================
=

                                        ORIG    8000h
l_campo                         STR             '#'
tras_nav                        STR             ')'
frent_nav                       STR             '>'
esq_nav                         STR             '\'
dir_nav                         STR             '/'
apaga                           STR             ' '
tiro_nav                        STR             '-'
tela_ini1                       STR             'Prepare-se@'
tela_ini2                       STR             'Prima o botao IE@'
Fim_Esc                         STR             '@'
asteroide                       STR             '*'
buraco_negro            STR             'O'
colunas                         STR     'Coluna@'
linhas                          STR     'Linha@'
Tela_fim1                       STR             'FIM DO JOGO@'
Pontuacao                       STR             'Pontuacao:@'
Pausa                           STR             ' PAUSA @'
Tiros                           STR             'TIROS:@'


;================================================================================
=
;FLAGS
;================================================================================
=

Flag_direita    WORD    0h
Flag_esquerda   WORD    0h
Flag_cima               WORD    0h
Flag_baixo              WORD    0h
Flag_tiro               WORD    0h
Flag_ini_jogo   WORD    0h
Flag_obs                WORD    0h
Flag_jogo               WORD    0h
Flag_cria_obs   WORD    6h
Flag_cria_bn    WORD    0h
Flag_reinicia   WORD    0h
Flag_game_over  WORD    1h
Flag_Pausa              WORD    0h
Flag_Hardness   WORD    2h

;================================================================================
```

```
=
;VARIAVEIS DE CONTROLO
;================================================================================
=

Pos_nave                WORD    0503h
Score                   WORD    0000h
Var_para_random WORD    0000h
Random                  WORD    AA41h
Pos_tiro                TAB             10
Pos_ast                 TAB             12
Pos_bn                  TAB             5


;================================================================================
=
;CODIGO
;================================================================================
=

;================================================================================
=
;ROTINA DE INICIALIZACAO
;================================================================================
=
;Inicia a mascara de interrupcoes, cursor e SP
;Entrada: ------
;Saida: ------

                        ORIG    0000h
inicio:         MOV             R1, Inic_SP
                MOV             SP, R1                          ;mudanca
 do SP para o endereco Inic_SP
                MOV             R1,     Masc_inic               ;usa a m
ascara apenas da rotina da tela inicial
                MOV             M[Mascara], R1          ;inicializacao d
a mascara

;================================================================================
=
;CICLO DE INICIO DE JOGO
;================================================================================
=
;Espera que seja premido o botao para iniciar o jogo
;Entradas: Flag_ini_jogo
;Saidas: ------
                        MOV     R1, Ini_cursor
                        MOV             M[Cursor], R1
;iniciacao do cursor
reinicio:               CALL    apaga_ecra
                        CALL    Repoe_vars
                        CALL    Tela_inicial                    ; vai ch
amar a funcao que desenha a Tela inicial
                        ENI

ciclo_ini:              MOV             R1, M[Flag_ini_jogo]
                        CMP             R1, 1h              ;compara cons
tantemente a flag que controla quando irÃ¡ ser iniciado o jogo
                        BR.NZ           ciclo_ini


;================================================================================
```



```
=
;ROTINA INICIO DE JOGO
;================================================================================
=
;Rotina que inicia o jogo quando e premido o botao IE
;Entradas: Ini_Cursor
;Saidas: ------

                                MOV             R1, Masc_jogo
                                MOV             M[Mascara], R1
                                CALL    LCD_inicial
Inicio_jogo:    MOV             R1, 1
                                MOV             M[Int_temp], R1
        ;ativa o contador
                                MOV             M[COM_Tempo], R1
        ;ativa a velocidade do contador(100ms)
                                MOV             M[Flag_ini_jogo], R0
                                CALL    apaga_ecra
                                CALL    Mapa
                                CALL    Nave

;================================================================================
=
;CICLO MAIN
;================================================================================
=
;ciclo_jogo onde ocorre todo o jogo e verifica cada botao que e premido
;Entradas:      Flag_direita, Flag_esquerda, Flag_baixo, Flag_cima , Flag_Pausa,
 Flag_ini_jogo, Score, Flag_tiro, Flag_jogo, Flag_game_over
;Saidas: ------

ciclo_jogo:             PUSH    R1
                                MOV             R1, M[Score]
                                CMP             R1, 32h
        ;compara o score com 32h = 50d
                                CALL.Z  aumenta_dificuldade
;caso chegue aos 50 pontos, aumenta a velocidade com que se realiza o movimento
dos asteroides
                                MOV             R1, M[Flag_ini_jogo]
                                CMP             R1, 1h
                                CALL.Z  reinicia
                                MOV             R1, M[Flag_Pausa]
                                CMP             R1, 1h
                                CALL.Z  pausa
                                MOV             R1, M[Flag_direita]
                                CMP             R1, 1h
                                CALL.Z  direita
                                MOV             R1, M[Flag_esquerda]
                                CMP             R1, 1h
                                CALL.Z  esquerda
                                MOV             R1, M[Flag_baixo]
                                CMP             R1, 1h
                                CALL.Z  baixo
                                MOV             R1, M[Flag_cima]
                                CMP             R1, 1h
                                CALL.Z  cima
                                MOV             R1, M[Flag_tiro]
                                CMP             R1, 1h
                                CALL.Z  tiro
                                MOV             R1, M[Flag_jogo]
                                CMP     R1, 1h
                                CALL.Z  ops_mapa
```

```
                        POP             R1
                        CMP             M[Flag_game_over], R0
                        JMP.Z   fim_jogo
                        JMP             ciclo_jogo

;================================================================================
=
;OPERACOES MAPA
;================================================================================
=
;Rotina que efetua o movimento do tiro em cada ciclo_jogo de relogio
;Entradas: Flag_jogo, Pos_tiro, Flag_Hardness, Flag_obs
;Saidas: Pos_tiro

ops_mapa:               DSI
                        PUSH            R1
                        PUSH            R2
                        INC                     M[Flag_obs]
                        ;incrementa a flag de obstaculo pois quando este
 chegar a 2

                                ;realiza o movimento do obstaculo
                        CALL            mov_tiro
                ;vai realizar o movimento do tiro
                        CALL            Colisao_tiro_obs

                        MOV                     R1, M[Flag_obs]
                        MOV             R2, M[Flag_Hardness]
                        CMP                     R1, R2
                        ;verifica se a flag de obstaculo chegou a 2
                        CALL.Z          ops_obs
                        CALL            Colisao_tiro_obs

                        CALL            num_tiros
                        MOV                     M[Flag_jogo], R0
                        POP                     R2
                        POP                     R1
                        ENI
                        RET


;================================================================================
=
;OPERACOES OBSTACULOS
;================================================================================
=
;Rotina que que chama o movimento dos obstaculos assim como a geraÃ$ao de um nov
o
;Entradas: Flag_cria_obs, Flag_obs
;Saidas: ------

ops_obs:        PUSH            R1
                MOV             R1, M[Flag_cria_obs]
                CMP                     R1, 6h
                ;verifica se cria um obstaculo novo na ultima co
luna do ecra
                CALL.Z          cria_obs
                ;caso a flag tenha atingido o valor 6, cria novo obstacu
lo
                CALL            move_obs
                ;move todos os obstaculos uma posicao para a esquerda
                CALL            Colisao_tiro_obs
                CALL            Colisao_nave
```



```
                        MOV                     M[Flag_obs], R0
        ;repoe a flag das operacoes com obstaculos
                        POP             R1
                        RET

;================================================================================
=
;DESENHO TELA INICIAL
;================================================================================
=
;Quando inicia o programa desenha a tela de inicio
;Entradas: tela_ini1, tela_ini2 , Loc_esc1, Loc_esc2
;Saidas: ------

Tela_inicial:   MOV             R1, Loc_esc1                    ;copia p
ara R1 o endereco(linha e coluna) de onde vai ser escrito a tela_ini1 (primeira
mensagem inicial)
                                MOV             R4, tela_ini1
; copia para R4 a string que contem a primeira mensagem
frase1:                 MOV             R3,     M[R4]
;copia para R3 a letra presente na string de R4
                                CMP             R3, M[Fim_Esc]
;compara a letra com o final da string(@) para ver se acabou de escrever ou nao
                                BR.Z    fim_frase1
                                MOV     M[Cursor], R1
                                MOV             M[Escrita], R3
                                INC             R4
                ;vai percorrer a string
                                INC             R1
                ;vai percorrer o endereco onde escreve
                                BR              frase1
fim_frase1:             MOV             R4, tela_ini2
;vai realizar o mesmo para a segunda mensagem
                                MOV             R1, Loc_esc2
frase2:                 MOV             R3, M[R4]
                                CMP             R3, M[Fim_Esc]
                                BR.Z    fim_frase2
                                MOV             M[Cursor], R1
                                MOV             M[Escrita], R3
                                INC             R4
                                INC             R1
                                BR              frase2
fim_frase2:             RET

;================================================================================
==
;DESENHO MAPA
;================================================================================
==
;Depois de iniciar o jogo desenha o mapa
;Entradas: Tiros
;Saidas: ------

Mapa:                   PUSH    R1
                                PUSH    R2
                                PUSH    R3
                                MOV             R1, R0
                                MOV             R2, Tiros
Esc0:                   MOV             R3, M[R2]
                                CMP             R3, M[Fim_Esc]
                                BR.Z    Esc_lim
                                MOV             M[Cursor], R1
```

```
                        MOV             M[Escrita], R3
;escreve a palavra "tiros" na primeira linha
                        INC             R1
                        INC             R2
                        BR              Esc0

Esc_lim:        MOV             R2, M[l_campo]                          ; mete e
m R2 o caratere #
l_sup:          MOV             M[Cursor], R1                   ;mete o cursor n
a primeira linha
                MOV             M[Escrita], R2
; escreve o caratere #
                INC     R1
                CMP             R1, 004Fh
                BR.NZ l_sup
;caso tenha atingido o limite da direita (004Fh), passar a delimitacao inferior
                MOV             R1, 1700h

l_inf:          MOV             M[Cursor], R1                           ;mete o
cursor na ultima linha
                MOV             M[Escrita], R2
;escreve o caratere #
                INC             R1
                CMP             R1, 174Fh
                BR.NZ l_inf                                             ;caso te
nha atingido o limite da direita (174Eh), acabar a delimitacao
                POP     R3
                POP     R2
                POP     R1
                RET



;=================================================================================
;DESENHO NAVE
;=================================================================================
;Desenha no mapa de jogo a nave consoante a posicao em que se encontra o canhao
;Entradas: Pos_nave
;Saidas: ------

Nave:                           PUSH    R1
                                PUSH    R2
                                MOV             R1, M[Pos_nave]
;coloca em R1 a posicao na nave(0503)
                                MOV             R2, M[frent_nav]
; coloca em R2 o caratere de frente_nav
                                CALL    escreve_nave
;coloca o cursor na posicao e escreve o caratere
                                CALL    coord
; vai atualizar o LCD
                                DEC             R1
                                MOV             R2, M[tras_nav]
                                CALL    escreve_nave
                                SUB             R1, 0100h
        ;subtrai uma linha
                                MOV             R2, M[esq_nav]
                                CALL    escreve_nave
                                ADD             R1, 0200h
        ;adiciona duas linhas
                                MOV             R2, M[dir_nav]
                                CALL    escreve_nave
                                POP             R2
                                POP             R1
```



```
                                RET

;=================================================================================
;APAGA NAVE
;=================================================================================
;Apaga a nave do mapa
;Entradas: Pos_nave
;Saidas: ------

apaga_nave:                     PUSH    R1
                                PUSH    R2
                                MOV             R2, M[apaga]
;vai utilizar o mesmo mÃ©todo de escrever a nave, mas com o caratere ' '(vazio)
                                MOV             R1, M[Pos_nave]
                                CALL    escreve_nave
;apaga a parte da frente
                                DEC             R1
                ;passa para a parte de tras
                                CALL    escreve_nave
                                ADD             R1, 0100h
        ;passa para a parte esquerda
                                CALL    escreve_nave
                                SUB     R1, 0200h
;passa para a parte direita
                                CALL    escreve_nave
                                POP             R2
                                POP             R1
                                RET

escreve_nave:           MOV             M[Cursor], R1
                                MOV             M[Escrita], R2
                                RET



;=================================================================================
;ROTINA DE MOVIMENTO DIREITA
;=================================================================================
;Efetua o movimento da nave uma posicao para a direita
;Entradas: Pos_nave
;Saidas: Nova pos_nave

direita:                DSI
                                PUSH    R1
                                MOV             R1, M[Pos_nave]
                                MVBH    R1, R0                          ;vai ign
orar a parte das linhas
                                CMP             R1, Lim_dir
;verifica se a nave pode andar para a direita
                                BR.Z    int_mov_dir                     ;caso te
nha atingido, nao ha movimento
                                CALL    apaga_nave
                                CALL    Mov_direita                     ;vai pas
sar a posicao nave um valor para a direita
                                CALL    Nave                            ;vai esc
rever de novo a nave na nova posicao
int_mov_dir:    MOV             R1, R0
                                MOV             M[Flag_direita], R1
; vai se repor a flag
                                CALL    Colisao_nave                    ;verific
a se houve colisao
                                POP             R1
```

```
                        ENI
                        RET

Mov_direita:    INC     M[Pos_nave]
                        RET

;=================================================================================
;ROTINA MOVIMENTO NAVE ESQUERDA
;=================================================================================
;Move a nave uma posicao para a esquerda
;Entradas: Pos_nave
;Saidas: Nova Pos_nave

esquerda:               DSI
                                PUSH    R1
                                MOV             R1, M[Pos_nave]
                                DEC             R1
                                                ;decrementa a posicao da
 nave por 1 valor
                                MVBH    R1, R0
                                ;vai ignorar as linhas
                                CMP             R1, Lim_esq
                                        ;verifica se a posicao nova atin
giu o limite ou nao
                                BR.Z    int_mov_esq
                                ;caso tenha atingido,nao ha movimento
                                CALL    apaga_nave
                                CALL    Mov_esquerda
                        ; move uma posicao para a esquerda
                                CALL    Nave
int_mov_esq:            MOV             R1, R0
                                MOV             M[Flag_esquerda], R1
                        ;vai repor a flag
                                CALL    Colisao_nave
                        ;verifica se houve colisao com algum obstaculo
                                POP             R1
                                ENI
                                RET

Mov_esquerda:           DEC     M[Pos_nave]
                                RET



;=================================================================================
=
;ROTINA MOVIMENTO NAVE BAIXO
;=================================================================================
=
;Efetua o movimento da nave uma posicao para baixo
;Entradas: Pos_nave
;Saidas: Nova Pos_nave

baixo:                  DSI
                                PUSH    R1
                                PUSH    R3
                                MOV             R1, M[Pos_nave]
                                ADD             R1, 0200h
                ;decrementa 2 linha na posicao
                                MOV     R3, R0
                                MVBH    R3, R1
                                CMP     R3, Lim_inf
```



```
        ;verifica se as linhas chegaram ao limite (verifica 2 linhas abaixo para
 nao atingir os limites)
                                BR.Z    int_mov_baixo
;caso tenha atingido, nao ha movimento
                                CALL    apaga_nave
                                CALL    Mov_baixo
                                CALL    Nave
                                CALL    Colisao_nave
;verifica se houve colisao com algum obstaculo
int_mov_baixo:          MOV             R1, R0
                                MOV             M[Flag_baixo], R1
        ;vai repor a flag
                                POP             R3
                                POP             R1
                                ENI
                                RET

Mov_baixo:                      PUSH    R1
                                MOV             R1, M[Pos_nave]
                                ADD     R1, 0100h
                                MOV             M[Pos_nave], R1
;vai adicionar uma linha a posicao da nave
                                POP             R1
                                RET

;=================================================================================
=
;ROTINA MOVIMENTO NAVE CIMA
;=================================================================================
=
;Efetua o movimento da nave uma posicao para cima
;Entradas: Pos_nave
;Saidas: Pos_nave

cima:                           DSI
                                PUSH    R1
                                PUSH    R3
                                MOV             R1, M[Pos_nave]
                                SUB             R1, 0200h
        ;subtrai 2 linhas a posicao atual
                                MOV             R3, R0
                                MVBH    R3, R1
                                CMP             R3, Lim_sup
        ;verifica se a posicao nova atingiu o limite ou nao(verifica 2 linhas ac
ima para nao atingir o limite)
                                BR.Z    int_mov_cima
;caso tenha atingido nao ha movimento
                                CALL    apaga_nave
                                CALL    Mov_cima
                                CALL    Nave
                                CALL    Colisao_nave
;verifica se houve colisao com algum obstaculo
int_mov_cima:           MOV             R1, R0
                                MOV             M[Flag_cima], R1
;vai repor a flag de interrupcao
                                POP             R3
                                POP             R1
                                ENI
                                RET

Mov_cima:                       PUSH    R1
                                MOV             R1, M[Pos_nave]
```

```
                                    SUB     R1, 0100h
                                    MOV             M[Pos_nave], R1
;vai subtrair 1 linha a posicao atual
                                    POP             R1
                                    RET

;================================================================================
=
;DISPARA TIRO
;================================================================================
=
;Desenha na posicao a frente da nave um tiro
;Entradas: Pos_nave, tiro_nav
;Saidas: Tiro

tiro:               DSI
                            PUSH            R1
                            PUSH            R2
                            PUSH            R3
                            MOV                     M[Flag_tiro], R0
                            MOV                     R1, Pos_tiro
                            MOV                     R2, R0
OutraPos:           CMP                     M[R1], R0
;verifica qual das entradas da tabela esta livre
                            BR.Z            faz_tiro
;caso esteja livre
                            INC                     R1
                            INC                     R2
                            CMP                     R2, 10
        ;verifica se percorreu as 10 entradas na tabela
                            BR.NZ           OutraPos
                            POP                     R3
                            POP                     R2
                            POP             R1
                            ENI
                            RET

faz_tiro:       MOV     R2, M[tiro_nav]
                                    MOV     R3, M[Pos_nave]
                                    INC     R3
                                    MOV     M[R1], R3
                                    CALL    escreve0
        ;vai escrever o tiro 1 posicao a frente da nave
                                    POP             R3
                                    POP             R2
                                    POP     R1
                                    ENI
                                    RET

;================================================================================
=
;MOVIMENTO DO TIRO
;================================================================================
=
;Rotina que efetua o movimento de todos os tiros em cada ciclo de relogio
;Entradas: Flag_jogo, Pos_tiro
;Saidas: ------

mov_tiro:                   PUSH    R1
                                    PUSH    R2
                                    MOV             R1, Pos_tiro
                                    MOV             R2, R0
```



```
OutraPos1:                  CMP             M[R1], R0
                                    CALL.NZ faz_mov
;verifica se o tiro tem uma posicao
                                    INC     R1
                                    INC             R2
                                    CMP             R2, 10
                                    BR.NZ   OutraPos1
;faz o movimento de todos os tiros no ecra
                                    POP             R2
                                    POP             R1
                                    RET

faz_mov:                    PUSH    R2
                                    PUSH    R3
                                    MOV             R3, M[R1]
                                    MOV             R2, M[apaga]
;apaga o local onde estava o tiro
                                    CALL    escreve0
                                    INC             R3
                                    MOV             R2, M[tiro_nav]
;escreve no local da nova posicao do tiro
                                    CALL    escreve0
                                    MOV             M[R1], R3
        ;atualiza a posicao do tiro
                                    CALL    el_tiro
                                    POP             R3
                                    POP             R2
                                    RET

el_tiro:                    MOV             M[R1], R3
                                    AND             R3, 00FFh
                                    CMP             R3, 004Eh
        ;verifica se o tiro ja esta na ultima coluna
                                    BR.NZ   nao_el

                                    MOV             R3, M[R1]
                                    MOV             R2, M[apaga]
;apaga o local onde estava o tiro
                                    MOV             M[R1], R0
        ;mete o tiro sem posicao
                                    CALL    escreve0
nao_el:                     RET

escreve0:                   MOV             M[Cursor], R3
                                    MOV             M[Escrita], R2
                                    RET

;================================================================================
=
;APAGAR ECRA
;================================================================================
=
;Rotina que apaga a tela de ecra
;Entradas: ------
;Saidas: -------

apaga_ecra:     PUSH    R1
                                    PUSH    R2
                                    MOV             R1, M[Linha0]
ciclo_apaga:            MOV             R2, M[apaga]
                                    MOV             M[Cursor], R1
;coloca o cursor na primeira linha
```

```
                                        MOV             M[Escrita], R2
;escreve na posicao o caratere '  '(vazio)
                                        INC             R1
        ;vai percorrer todas as posicoes da linha
                                        CMP             R1, 174Fh
;verifica se ja esta na ultima linha e ultima coluna
                                        BR.Z    acaba
                                        MVBL    R2, R1
                                        CMP             R2, 004Fh
;verifica se chegou a ultima coluna de cada linha
                                        BR.NZ   ciclo_apaga         ;caso esteja
, incrementa 1 linha e repete o processo e caso nao esteja, volta a repetir o pr
ocesso
                                        ADD             R1, 0100h
                                        AND             R1, FF00h
;quando incrementa uma linha, vai apagar os valores das colunas
                                        BR              ciclo_apaga
;vai repetir o processo
acaba:                          POP             R2
                                        POP             R1
                                        RET


;================================================================================
=
;CRIA OBSTACULO
;================================================================================
=
;Rotina responsavel pela criaÃ$Ã£o do obstaculo
;Entradas: Flag_cria_obs, Flag_cria_bn, Pos_ast, Var_para_random
;Saidas: ------

cria_obs:                               PUSH    R1
                                                PUSH    R2
                                                PUSH    R3
                                                CALL    PosAleatoria
                ; vai buscar uma posicao aleatoria para colocar o novo obstaculo
                                                MOV             M[Flag_cria_obs]
, R0                    ; vai repor a flag para criar obstaculo a 0
                                                MOV             R1, M[Flag_cria_
bn]
                                                CMP             R1, 3
                                ; compara a flag que cria buraco negro com o val
or 3
                                                JMP.Z   cria_bn
                        ; caso seja 3, vai criar um buraco negro
                                                INC             M[Flag_cria_bn]
                        ; vai incrementar 1 valor na flag que cria buraco negro
sempre que e feito um asteroide
cria_ast:                               MOV             R1, Pos_ast
                                                MOV             R2, R0
nova_pos_ast1:                  CMP             M[R1], R0
                ;compara a posicao do asteroide com a posicao 0000
                                                BR.Z    des_ast
                                ;caso seja igual, desenha um novo asteroide
                                                INC             R1
                                                ;vai incrementar uma posicao do asteroid
e
                                                INC             R2
                                                ;vai incrementar um valor em R2 para per
correr as 10 posicoes
                                                CMP             R2, 12
                                                BR.NZ   nova_pos_ast1
```



```
                ;caso nao tenha percorrido todas as posicoes, volta a repetir o
processo
                                                POP             R3
                                                POP             R2
                                                POP             R1
                                                RET

des_ast:                                MOV             R2, M[Var_para_random]
                                                MOV             R3, M[asteroide]
                                                MOV             M[R1], R2
                                ;coloca na posicao de cada asteroide a posicao r
andom
                                                MOV             M[Cursor], R2
                        ;vai colocar o cursor numa posicao random
                                                MOV             M[Escrita], R3
                        ;vai escrever o caratere do asteroide( '*' )
                                                POP             R3
                                                POP             R2
                                                POP             R1
                                                RET

cria_bn:                        MOV             R1, Pos_bn
                                                MOV             R2, R0
nova_pos_bn1:                           CMP             M[R1], R0
                ;compara  a posicao de bn com 0000
                                                BR.Z    des_bn
                                ;caso seja igual, vai desenhar um buraco negro
                                                INC             R1
                                                ;incrementa uma posicao de buraco negro
                                                INC             R2
                                                CMP             R2, 5
                                ;percorre as 6 posicoes dos buracos negros
                                                BR.NZ nova_pos_bn1
FimBN:                          MOV     M[Flag_cria_bn], R0
        ;repoe a flag do buraco negro
                                                POP             R3
                                                POP             R2
                                                POP             R1
                                                RET

des_bn:                                 MOV             R2, M[Var_para_random]
                                                MOV             R3, M[buraco_neg
ro]
                                                MOV             M[Cursor], R2
                        ;vai colocar o cursor na posicao random
                                                MOV             M[Escrita], R3
                        ;vai escrever o caratere do buraco negro( 'O' )
                                                MOV             M[R1], R2
                                ;coloca na posicao de cada buraco negro a posica
o random
                                                BR              FimBN


;================================================================================
;MOVE OBSTACULO
;================================================================================
;Rotina que controla o movimento dos obstaculos
;Entradas: Pos_ast , sem_pos_obs , Flag_cria_bn
;Saidas: ------

move_obs:                               PUSH    R1
                                                PUSH    R2
                                                PUSH    R3
```

```
                                    MOV             R1, Pos_ast
                        ; coloca em R1 a posicao do asteroide
                                    MOV             R2, R0
                                    MOV             R3, sem_pos_obs
                ;coloca em R3 uma posicao com a qual a posicao do asteroide nunc
a vai ser igual
nova_pos_ast:           CMP         M[R1], R3
;verifica cada uma das posicoes da tabela para indicar se tem ou nao um elemento
                                    CALL.NZ move_ast
                                    INC             R1
                                ;vai percorrer todos os asteroides
                                    INC             R2
                                    CMP             R2, 12
                                    BR.NZ   nova_pos_ast
        ;enquanto nao percorrer os 11 obstaculos, vai repetir o processo
nova_pos_bn:            CMP         M[R1], R3
        ;verifica cada uma das posicoes para indicar se tem um elemento
                                    CALL.NZ move_bn
                                    INC             R1
                                ;vai incrementar a posicao do obstaculo
                                    INC             R2
                                    CMP             R2, 17
                                    BR.NZ   nova_pos_bn
                ;enquanto nao percorrer os 15 obstaculos, vai repetir o processo
 relativamente aos buracos negros
                                    INC             M[Flag_cria_obs]
                                    POP             R3
                                    POP             R2
                                    POP     R1
                                    RET

move_ast:                       PUSH    R4
                                    PUSH    R5
                                    MOV             R4, M[R1]
                                    MOV             R5, ' '
                                    MOV             M[Cursor], R4
                                    MOV             M[Escrita], R5
                ;escreve na posicao do asteroide o caratere vazio (' ')
                                    DEC             R4
                                ;vai uma posicao para a esquerda
                                    MOV             R5, M[asteroide]
                                    MOV             M[Cursor], R4
                                    MOV             M[Escrita], R5
                ;escreve na nova posicao o caratere do asteroide ' * '
                                    MOV             M[R1], R4
                        ;atribui a nova posicao Ã  posicao do asteroide
                                    CALL    elimina_obs
                                    POP             R5
                                    POP             R4
                                    RET

move_bn:                        PUSH    R4
                                    PUSH    R5
                                    MOV             R4, M[R1]
                                    MOV             R5, ' '
                                    MOV             M[Cursor], R4
                                    MOV             M[Escrita], R5
                ;escreve na posicao do obstaculo o caratere vazio (' ')
                                    DEC             R4
                ;passa uma posicao para a esquerda
                                    MOV             R5, M[buraco_negro]
                                    MOV             M[Cursor], R4
```



```
                                    MOV             M[Escrita], R5
;escreve o buraco negro
                                    MOV             M[R1], R4
                                    CALL    elimina_obs
;verifica se chegou a primeira coluna
fim_mov_bn:                 POP             R5
                                    POP             R4
                                    RET

elimina_obs:            PUSH    R4
                                    PUSH    R5
                                    MOV     R5, ' '
                                    MOV             R4, M[R1]
                ;colocase em R4 a posicao do asteroide
                                    AND             R4, 00FFh
                ;ignora as linhas
                                    CMP             R4, R0
                ;verifica se esta na primeira coluna
                                    BR.NZ nao_elimina
        ;se estiver apaga o obstaculo
                                    MOV             R4, M[R1]
                                    MOV             M[Cursor], R4
                                    MOV             M[Escrita], R5
                                    MOV             M[R1], R0
nao_elimina:            POP             R5
                                    POP     R4
                                    RET

;=========================================================================
;COLISAO TIRO-OBSTACULOS
;=========================================================================
;Rotina que verifica se houve colisao entre os tiros e os obstaculos
;Entradas: Pos_tiro, Pos_ast, Pos_bn
;Saidas: ------
Colisao_tiro_obs:       PUSH    R1
                                    PUSH    R2
                                    PUSH    R3
                                    PUSH    R4
                                    PUSH    R5
                                    MOV     R1, Pos_tiro
                                    MOV     R2, R0
                                    MOV             R5, R0

OutroTiro3:                     MOV             R4, Pos_ast
                                    CMP             M[R1], R0
                            ;verifica se o tiro esta a ser utilizado ou nao
                                    JMP.Z   salta
                                    MOV             R3, M[R1]
                                    MOV             R2, R0

ast:                            CMP             M[R4], R3
                ;verifica se a posicao do asteroide e igual a do tiro
                                    JMP.Z   colidiu_ast

                                    INC             R4
                                ;percorre todos os asteroides
                                    INC             R2
                                    CMP             R2, 12
                                    BR.NZ ast

                                    MOV     R2, R0
                                    MOV             R4, Pos_bn
```

```
bn:                                     CMP             M[R4], R3
                        ;verifica se a posicao do buraco negro e igual a do tiro
                                        JMP.Z   colidiu_bn
                                        INC             R4
                                ;percorre todos os buracos negros
                                        INC             R2
                                        CMP             R2, 5
                                        JMP.NZ bn

salta:                          INC             R1
                                        INC             R5
                                        CMP             R5, 10
                        ;permite percorrer todos os 10 tiros
                                        JMP.NZ  OutroTiro3
FimColisao:                     POP             R5
                                        POP             R4
                                        POP             R3
                                        POP             R2
                                        POP             R1
                                        RET

colidiu_ast:            MOV             M[Cursor], R3
                                        MOV             R2, M[apaga]
                                        MOV             M[Escrita], R2
                                        CALL    acende_LED
                                        INC             M[Score]
                                        CALL    Display
;incrementa o score no display de 7 segmentos
                                        MOV             M[R1], R0
        ;passa o tiro para uma posicao nula
                                        MOV     M[R4], R0
;da reset na posicao do asteroide
                                        JMP             FimColisao

colidiu_bn:                     MOV             M[Cursor], R3
                                        MOV             R2, M[buraco_negro]
                                        MOV             M[Escrita], R2
                                        MOV             M[R1], R0
                                        JMP             FimColisao

;================================================================================
;POSICAO ALEATORIA
;================================================================================
;rotina que gera uma posicao aleatoria de acordo com a posicao anterior

PosAleatoria:   PUSH    R4
                                                        PUSH    R5
                                                        CALL    ValorAleatorio
                                                        MOV             R4, 0016
h
                                                        DIV             R5, R4
                                                        ADD             R4, 0001
h
                                                        SHL             R4, 8
                                                        MVBL    R4, 004Eh
                                                        MOV             M[Var_pa
ra_random], R4
                                                        POP             R5
                                                        POP             R4
                                                        RET

; ValorAleatorio: Utiliza a rotina descrita no enunciado para gerar um numero de
```



```
 16 bits aleatorio

ValorAleatorio: MOV             R5, M[Random]
                                                                        TEST    R5, 0001
h
                                                                        BR.Z    Rotacao0
                                                                        XOR
R5, 8016h
Rotacao0:                                       ROR             R5, 1
                                                                        MOV
M[Random], R5   ;Coloca-o na posiÃ§Ã£o de memoria correspondente ao Random
                                                                        RET

;================================================================================
==
;ESCRITA LCD
;================================================================================
==
;Rotina que escreve as palavras "Linha" e "Coluna" no LCD
;Entradas: colunas, linhas, Fim_Esc
;Saidas: ------

LCD_inicial:                                    PUSH    R1
                                                                PUSH    R2
                                                                MOV     R1, 800Ah
                                                                MOV     R2, colunas
                                                                CALL    Escrita_sec
        ;vai escrever na primeira linha a partir da coluna A a frase 'coluna'
                                                                MOV     R1, 8002h
                                                                MOV     R2, linhas
                                                                CALL    Escrita_sec
                ;vai escrever na primeira linha a partir da coluna 2 a frase 'li
nha'
                                                                POP     R2
                                                                POP     R1
                                                                RET

Escrita_sec:                                    PUSH    R3
                                                                PUSH    R4
escreve1:                                               MOV     R3, M[R2]
        ;vai colocar em R3 cada letra
                                                                MOV     R4, Fim_Esc
                                                                CMP     R3, M[R4]
                ;verifica se chegou ao fim da string
                                                                BR.Z    Fim_LCD
                ;caso nao seja, continua a escrever e repete o processo
                                                                MOV     M[Cursor_LCD], R
1
                                                                MOV     M[Escrita_LCD],
R3
                                                                INC     R1
                                                                INC     R2
                                                                BR      escreve1
Fim_LCD:                                                POP     R4
                                                                POP     R3
                                                                RET

;================================================================================
=
;ESCRITA COORDENADAS
;================================================================================
=
```

```
;Rotina que escreve as coordenadas do canhao da nave no LCD
;Entradas: Pos_nave , ConvASCII
;Saidas: ------

coord:                          PUSH    R1
                                PUSH    R3
                                PUSH    R7
                                MOV             R7, M[Pos_nave]
;coloca em R7 a posicao da nave

escreve_linhas:         SHR             R7, 8
                                MOV             R1, 10
                                DIV             R7, R1
        ;vai dividir o 8 bits correspondentes as linhas por 10

                                ;vai colocar em R1 o valor das unidades e em R7
o valor das dezenas
                                ADD             R7, ConvASCII
                                ADD             R1, ConvASCII
                                MOV             R3, 8000h
                                MOV             M[Cursor_LCD], R3
                                MOV             M[Escrita_LCD], R7
; vai escrever na primeira linha na coluna 0 os valores das linhas
                                INC             R3
                                MOV             M[Cursor_LCD], R3
                                MOV             M[Escrita_LCD], R1

escreve_colunas:        MOV             R7, M[Pos_nave]
                                AND             R7, 00FFh
        ;vai ignorar as linhas
                                MOV             R1, 10
                                DIV             R7, R1
        ;coloca em R7 o valor das dezenas das colunas e em R1 o valor das unidad
es
                                ADD             R7, ConvASCII
                                ADD             R1, ConvASCII
                                MOV             R3, 8008h
                                MOV             M[Cursor_LCD], R3
                                MOV             M[Escrita_LCD], R7
;vai escrever na primeira linha na coluna 8 os valores das colunas
                                INC             R3
                                MOV             M[Cursor_LCD], R3
                                MOV             M[Escrita_LCD], R1
                                POP             R7
                                POP             R3
                                POP             R1
                                RET


;=================================================================================
=
;ACENDER LEDS
;=================================================================================
=
;Rotina que acende todos os pontos dos LEDS
;Entradas: LED
;Saidas: ------

;Saida: R5
acende_LED:                                     PUSH    R5
                                                MOV             R5, FFFF
h
                                                MOV             M[LED],
```



```
R5                      ;acende todos os leds
NaoAcende:              POP             R5
                                                RET

;=================================================================================
=
;COLISAO COM A NAVE
;=================================================================================
=
;Rotina que verifica se houve colisao da nave com um obstaculo em todas as parte
s da nave
;Entrada: Pos_nave, Pos_ast, Flag_game_over, Masc_final, Flag_reinicia, Flag_Pau
sa
;Saida: ------
Colisao_nave:   PUSH    R1
                                PUSH    R2
                                PUSH    R3
                                MOV             R1, M[Pos_nave]
                                CALL    verifica_col            ;verifica se hou
ve colisao com um obstaculo na parte da frente da nave
                                DEC             R1
                                CALL    verifica_col            ;verifica se hou
ve colisao com um obstaculo na parte de tras da nave
                                ADD             R1, 0100h
                                CALL    verifica_col            ;verifica se hou
ve colisao com um obstaculo na parte de baixo da nave
                                SUB             R1, 0200h
                                CALL    verifica_col            ;verifica se hou
ve colisao com um obstaculo na parte da cima da nave
                                POP             R3
                                POP             R2
                                POP             R1
                                RET

verifica_col:   MOV             R2, R0
                                MOV             R3, Pos_ast
outro_obs:              CMP             M[R3], R1                       ;verific
a se a posicao do asteroide e igual a da nave
                                BR.NZ   nao_colidiu
                                MOV             M[Flag_game_over], R0   ;caso ha
ja colisao vai ativar a flag que acaba o jogo
                                MOV             R2, 16
nao_colidiu:    INC             R2                                      ;increme
nta o R2 para percorrer todos os obstaculos
                                INC             R3
;Percorre as posicoes dos obstaculos
                                CMP             R2, 17
                                BR.NZ   outro_obs
                                RET


fim_jogo:               CALL    apaga_ecra
                                CALL    Tela_final
                                MOV             R1, Masc_final
;ativa todas a interrupcoes de forma a se poder clicar em qualquer botao para re
iniciar
                                CALL    Repoe_vars
                                MOV             M[Mascara], R1

ciclo_final:    MOV             R1, M[Flag_reinicia]
                                CMP             R1, 1
;verifica se foi  ativada alguma interrupcao que reinicie o jogo
```

```
                                BR.NZ   ciclo_final
                                MOV             M[Flag_Pausa], R0
                                JMP.Z   reinicio

;================================================================================
=
;REPOSICAO VARIAVEIS
;================================================================================
=
;Rotina que vai repor todas as variaveis quando o jogo Ã© reiniciado
;Entradas: Todas as flags, contador e Score e tabelas
;Saidas: -----

Repoe_vars:     PUSH    R1
                                PUSH    R2
                                MOV             M[COM_Tempo], R0
;desliga o contador
                                MOV             R1,     6
                ;coloca o valor 6 em R1 para repor a flag que cria obstaculo
                                MOV             M[Flag_direita], R0
                                MOV             M[Flag_esquerda], R0
                                MOV             M[Flag_cima], R0
                                MOV             M[Flag_baixo], R0
                                MOV             M[Flag_ini_jogo], R0
                                MOV             M[Flag_obs], R0
                                MOV             M[Flag_jogo], R0
                                MOV             M[Flag_cria_obs], R1
                                MOV             M[Flag_cria_bn], R0
                                MOV             M[Flag_reinicia], R0
                                MOV             R1, 1
                                MOV             M[Flag_game_over], R1
                                MOV             R1, 2
                                MOV             M[Flag_Hardness], R1
                                MOV             M[Score], R0
                                MOV             R1, Pos_tiro
                                MOV             R2, R0
repoe_tab:              MOV             M[R1], R0
;vai colocar a tabela toda a valores iniciais
                                INC             R1
                                INC             R2
                                CMP             R2, 26
        ;o R2 permite percorrer todos 16 obstaculos
                                BR.NZ   repoe_tab
                                MOV             R1, pos_nave_ini
                                MOV             M[Pos_nave], R1
                                MOV             R1, FFF0h
                                MOV             R2, R0
repoe_disp:             MOV             M[R1], R0
                                INC             R1
                                INC             R2
                                CMP             R2, 5
                                BR.NZ   repoe_disp
                                POP             R2
                                POP             R1
                                RET

;================================================================================
=
;ESCRITA DO FIM DE JOGO
;================================================================================
=
;Rotina onde ocorre a escrita da mensagem final de jogo na tela
```



```
;Entradas: Loc_esc3 , Tela_fim1, Fim_Esc, Loc_esc2, Pontuacao, ConvASCII , Score
;Saidas: -----

Tela_final:             PUSH    R1
                                PUSH    R2
                                PUSH    R3
                                MOV             R1, Loc_esc3
                                MOV             R2, Tela_fim1
prox_letra:             MOV             R3, M[R2]
;coloca em R3 cada letra
                                CMP             R3, M[Fim_Esc]
;verifica se chegou ao fim da string
                                BR.Z    fim_tela1
                                MOV             M[Cursor], R1
                                MOV             M[Escrita], R3
;escreve uma letra da frase
                                INC             R2
                ;percorre a frase
                                INC             R1
                ;percorre a linha
                                BR              prox_letra
fim_tela1:              MOV             R1, Loc_esc2
                                MOV             R2, Pontuacao
prox_letra2:    MOV             R3, M[R2]
                                CMP             R3, M[Fim_Esc]
                                BR.Z    fim_frase3
                                MOV             M[Cursor], R1
                                MOV             M[Escrita], R3
;Vai escrever a segunda frase
                                INC             R2
                                INC     R1
                                BR              prox_letra2
fim_frase3:             CALL    conv_dec
                                POP             R3
                                POP             R2
                                POP     R1
                                RET


conv_dec:               PUSH    R2
                                PUSH    R3
                                PUSH    R4
                                MOV             R4, R0
;colocase a 0 e incrementase ate chegar a 4 para repetir o processo 4 vezes
                                MOV             R2, M[Score]
                                ADD             R1, 4
;vai colocar em R1 o valor de 4 e vai decrementando

        ;para percorrer as 4 posicoes da linha com os valores
nova_un:                MOV             R3, 10
                                DIV             R2, R3
;vai dividir o Score por 10
                                ADD             R3, ConvASCII
                                MOV             M[Cursor], R1
                                MOV             M[Escrita], R3
;escreve o resto no primeiro display
                                DEC             R1
                                INC             R4
                                CMP             R4, 4
                                BR.NZ   nova_un
                                POP             R4
                                POP             R3
                                POP             R2
```

```
                        RET

;=================================================================================
;REINICIO DE JOGO
;=================================================================================
;Rotina onde quando se clica no botao IE durante o jogo e tudo reiniciado
;Entradas: ------
;Saidas: ------

reinicia:       DSI
                PUSH    R1
                CALL    apaga_ecra
                CALL    Repoe_vars
                CALL    Mapa
                CALL    Nave
                CALL    num_tiros
                MOV             R1, 1
                MOV             M[Int_temp], R1
                MOV             M[COM_Tempo], R1
                POP             R1
                ENI
                RET

;=================================================================================
;NUMERO DE TIROS
;=================================================================================
;Rotina que mostra no canto superior esquerdo o numero de tiros disponiveis
;Entradas: Pos_tiro, ConvASCII
;Saidas: ------

num_tiros:      PUSH    R1
                PUSH    R2
                PUSH    R3
                PUSH    R4
                PUSH    R5
                MOV             R1,     Pos_tiro
                MOV             R2, R0
                MOV             R3, 10
OutraPos4:      CMP             M[R1], R0
;verifica se o tiro na tabela se encontra vazio
                BR.Z    com_tiro
                DEC             R3
com_tiro:       INC             R1
                INC             R2
                CMP             R2, 10
        ;verifica se percorreu os 10 tiros
                BR.NZ OutraPos4
                MOV             R4, 10
                MOV             R5, 0006h

                DIV             R3, R4
                ADD             R3, ConvASCII
                ADD             R4, ConvASCII
                CALL    escreve2
;escreve o numero das unidades
                INC             R5
            ;passa 1 posicao a frente na primeira linha
                MOV             R3, R4

                CALL    escreve2
;escreve o numero das dezenas
                POP             R5
```



```
                POP             R4
                POP             R3
                POP             R2
                POP             R1
                RET

escreve2:       MOV             M[Cursor], R5
                MOV             M[Escrita], R3
                RET

;=================================================================================
=
;AUMENTA DIFICULDADE
;=================================================================================
=
;Rotina que aumenta a velocidade do movimento dos obstaculo
;Entradas: Flag_Hardness
;Saidas: ------
aumenta_dificuldade:    PUSH    R1
                                                MOV     R1, 1h
                                                MOV     M[Flag_Hardness], R1
                                                POP             R1
                                                RET


;=================================================================================
=
;PAUSA DE JOGO
;=================================================================================
=
;Rotina que efetua uma pausa no jogo
;Entradas: Flag_Pausa, COM_Tempo, Fim_Esc, Loc_esc4, Flag_tiro
;Saidas: ------

pausa:          PUSH    R1
                PUSH    R2
                MOV             M[COM_Tempo], R0
            ;desliga o contador
                MOV             M[Flag_Pausa], R0
            ;repoe a flag  de pausa
                MOV             R1, Loc_esc4

                MOV             R2, Pausa
esc:            MOV             R3, M[R2]
                CMP             R3, M[Fim_Esc]
                BR.Z    ciclo_p
                MOV             M[Cursor], R1
                MOV             M[Escrita], R3
            ;vai escrever na primeira linha a palavra "Pausa"
                INC     R1
                INC             R2
                BR              esc
ciclo_p:        MOV     R1, M[Flag_Pausa]
                CMP             R1, 1
                BR.NZ   ciclo_p
            ;caso a flag nao seja ativada novamente, vai permanecer em pausa
                CALL    Mapa
            ;retira a palavra "Pausa"
                MOV             M[Flag_tiro], R0
                MOV             M[Flag_Pausa], R0
                MOV             R1, 1
                MOV             M[COM_Tempo], R1
```

```
                ;volta a repor o contador
                        POP             R2
                        POP             R1
                        RET

;================================================================================
;DISPLAY 7 SEGMENTOS
;================================================================================
;Rotina onde escreve a pontuacao do jogador no display
;Entradas: Score, Display7seg
;Saidas: ------

Display:        PUSH    R1
                        PUSH    R3
                        PUSH    R4
                        MOV             R1, M[Score]
                        MOV             R3, Display7seg
novo_seg:       MOV             R4, 10
                        DIV             R1, R4
                        ADD             R4, ConvASCII
                        MOV             M[R3], R4
;colocase no display o valor do resto que contem o valor das unidades
                        INC             R3
;percorre os 4 displays
                        CMP             R3, FFF4h
;verifica se chegou ao ultimo display de 7 segmentos
                        BR.NZ novo_seg
                        POP             R4
                        POP             R3
                        POP             R1
                        RET

;================================================================================
;ROTINA TRATAMENTO BOTAO I0
;================================================================================
;Ativa a flag_baixo
;Entradas: Flag_baixo
;Saidas: ------

baixo_0:        DSI
                        PUSH    R1
                        MOV             R1, 1h
                        MOV             M[Flag_baixo], R1
;interrupcao que vai ativar a rotina que move para baixo
                        MOV             M[Flag_reinicia], R1
                        POP             R1
                        ENI
                        RTI

;================================================================================
;ROTINA TRATAMENTO BOTAO I1
;================================================================================
;Indica ao programa que foi premido o botao I1
;Entradas: Flag_cima
;Saidas: ------

cima_0:         DSI
                        PUSH    R1
                        MOV             R1, 1h
                        MOV             M[Flag_cima], R1
;interrupcao que ativa a rotina que movimenta a nave para cima
                        MOV             M[Flag_reinicia], R1
```



```
                        POP             R1
                        ENI
                        RTI

;================================================================================
;ROTINA TRATAMENTO BOTAO I2
;================================================================================
;Indica ao programa quando se carregou no botao I3
;Entradas: Flag_esquerda
;Saidas: ------

esquerda_0:     DSI
                        PUSH    R1
                        MOV             R1, 1h
                        MOV             M[Flag_esquerda], R1
;interrupcao que vai ativar a rotina que movimenta a nave para a esquerda
                        MOV             M[Flag_reinicia], R1
                        POP             R1
                        ENI
                        RTI

;================================================================================
;ROTINA TRATAMENTO BOTAO I3
;================================================================================
;Indica ao programa quando e premido o botao I4
;Entradas: Flag_direita
;Saidas: ------

direita_0:      DSI
                        PUSH    R1
                        MOV             R1, 1h
                        MOV             M[Flag_direita], R1
;interrupcao que vai fazer com que se ative a rotina para se mover para a direit
a
                        MOV             M[Flag_reinicia], R1
                        POP             R1
                        ENI
                        RTI

;================================================================================
;ROTINA TRATAMENTO BOTAO I4
;================================================================================
;Indica ao programa que foi premido o botao I4
;Entradas: Flag_cima
;Saidas: ------

tiro_0:         DSI
                        PUSH    R1
                        MOV             R1, 1h
                        MOV             M[Flag_tiro], R1
;interrupcao que ativa a criacao e movimento do tiro
                        MOV             M[Flag_reinicia], R1
                        POP             R1
                        ENI
                        RTI

================================================================================
;ROTINA TRATAMENTO BOTAO I5
;================================================================================
;Indica ao programa que foi premido o botao I4
;Entradas: Flag_cima
;Saidas: ------
```

```
pausa_0:        DSI
                PUSH    R1
                MOV             R1, 1h
                MOV             M[Flag_Pausa], R1
;interrupcao que ativa a criacao e movimento do tiro
                MOV             M[Flag_reinicia], R1
                POP             R1
                ENI
                RTI

;*********************************************************************************
;ROTINA TRATAMENTO IE
;*********************************************************************************
;Muda o valor da Flag_ini_jogo para o comeco do jogo
;Entrada: Flag_ini_jogo , Flag_reinicia
;Saida: ------

inicio_0:       DSI
                PUSH    R1
                MOV             R1, 1h
                MOV             M[Flag_ini_jogo], R1            ;interru
pcao que controla a flag que permite iniciar o jogo
                MOV             M[Flag_reinicia], R1
                POP             R1
                ENI
                RTI

;*********************************************************************************
;ROTINA DE FIM DE JOGO
;*********************************************************************************
;Rotina onde ocorre qualquer interrupcao

reinicia_0:     DSI
                PUSH    R1
                MOV             R1, 1h
                MOV             M[Flag_reinicia], R1
                POP             R1
                ENI
                RTI

;*********************************************************************************
;ROTINA DO CLOCK
;*********************************************************************************
;Rotina onde ocorre a interrupÃ§ao I15

Contador:       PUSH    R1
                MOV             R1, 1
                MOV             M[Flag_jogo], R1
                ;incrementa a flag que indica quando ocorre o movimento do tiro
para controlar o seu movimento
                MOV             M[LED], R0
                ; vai apagar os leds
                MOV             M[Int_temp], R1
                ;repoe o tempo do contador
                COM             M[COM_Tempo]
                ;ativa novamente o contador
                POP     R1
                RTI
```