```
;                                                                    GRUPO 50
;

;--------------------------------------------------------------------------------
--------------------------------
;--------------------------------------------------------------------------------
--------------------------------
;                                                 Definicao de constantes

;--------------------------------------------------------------------------------
--------------------------------
;--------------------------------------------------------------------------------
--------------------------------

SP_INICIAL                    EQU   FDFFh
CURSOR                              EQU   FFFCh
CURSOR_INIT                   EQU   FFFFh
WINDOW_WRITER                       EQU   FFFEh
FIM_TEXTO                     EQU   '@'
INTMASK                             EQU   FFFAh
INT1MASK                      EQU   0000000000000010b
INTMAINMASK                   EQU   1000000000000111b  ;I15(TEMP), I0(SALTA),
I1(DESCE LVL), I2(AUMENTA LVL)
TEMP                          EQU   FFF6h
TEMP_AMOUNT                   EQU   1
TEMP_ENABLE                   EQU   FFF7h

DISPLAY_7                     EQU   FFF0h
LEDS                          EQU   FFF8h
LCD_WRITE                     EQU   FFF5h
LCD_CONTROL                   EQU   FFF4h

Mascara_aleatorio             EQU   1000000000010110b
Valor_divisao_aleatorio             EQU   000Ch

LVL_Maximo                    EQU   0001h
LVL_Minimo                    EQU   0004h

VALOR_LVL_1                   EQU   0004h
VALOR_LVL_2                   EQU   0003h
VALOR_LVL_3                   EQU   0002h

CLEAN                         EQU   ' '
Simbolo_Horizontal                  EQU   '-'
simbolo_passaro_1             EQU   'O'
simbolo_passaro_2             EQU   '>'
simbolo_tubos                       EQU   'X'
passaro_inicio_cursor               EQU   0C14h

VarT1_Cursor                        EQU   0C23h
VarT2_Cursor                        EQU   0D1Ch
VarT3_Cursor                        EQU   0C23h
VarT4_Cursor                        EQU   0D20h
Var_T4_Cursor_2                     EQU   0D31h

;--------------------------------------------------------------------------------
--------------------------------
;--------------------------------------------------------------------------------
--------------------------------
;                                                   Definicao de Variaveis

;--------------------------------------------------------------------------------
--------------------------------
;--------------------------------------------------------------------------------
```

```
---------------------------------
                            ORIG        8000h


Var_Texto_1                 STR   'Prepare-se' , FIM_TEXTO
Var_Texto_2                 STR   'Prima o interruptor I1' , FIM_TEXTO
Var_Texto_3                 STR   'FIM DE JOGO' , FIM_TEXTO
Var_Texto_4                 STR   'Pontuacao Max' , FIM_TEXTO
Distancia                   STR   'Distancia' , FIM_TEXTO

Posicao_Cursor_Hor                WORD  0000h

CURSOR_PASSARO                    WORD  0C14h

Contador_avanca_tubos             WORD  0004h       ;tubos avancam de X em X
segundos
VALOR_LVL                   WORD  0004h             ;comeca na velocidade
minima

Contador_gravidade                WORD  0004h
Contador_gravidade_inicial        WORD  0004h

GAME_STARTED                      WORD  0000h
FLAG_I0                           WORD  0000h
FLAG_GRAVIDADE                    WORD  0000h
FLAG_CRASH                  WORD  0000h
FLAG_I1                           WORD  0000h
FLAG_I2                           WORD  0000h

Posicao_tubos                     TAB   13      ;primeiros 2 bits=posicao da
porta (linha);outros 2 =posicao do tubo(coluna)
Ponteiro_tabela                   WORD  Posicao_tubos

Mascara_aleatoria_inicial         WORD  0000h
Mascara_cursor                    WORD  0000h

Var_est_cria_tubos                WORD  0000h
Var_est_cria_tubos_MAIN           WORD  0006h

VALOR_DISTANCIA_LCD               WORD  0000h
NUMERO_OBSTACULOS           WORD  0000h

;-------------------------------------------------------------------------------
---------------------------------
;-------------------------------------------------------------------------------
---------------------------------
;                                           TABELA DE INTERRUPCOES

;-------------------------------------------------------------------------------
---------------------------------
;-------------------------------------------------------------------------------
---------------------------------
                            ORIG  FE00h
INT0                        WORD  Salta

                            ORIG  FE01h
INT1                        WORD  Start

INT2                        WORD  LVL

                            ORIG  FE0Fh
INT15                             WORD  Temp_Cont

;-------------------------------------------------------------------------------
```

```
;--------------------------------------------------------------------------------
;                                                      CODIGO

;--------------------------------------------------------------------------------
;--------------------------------------------------------------------------------

                            ORIG  0000h
                            JMP   Inicio

;Rotinas de servico a interrupcoes
Salta:                            INC   M[FLAG_I0]
                            RTI


Start:                            CMP M[GAME_STARTED], R0   ;verifica se o
jogo esta ativo
                            BR.NZ Start_1
                            INC M[GAME_STARTED] ;se o jogo nao estiver ativo
incrementa a FLAG de estado de jogo
                            BR Fim_Start
Start_1:                    INC M[FLAG_I1]    ;se o jogo estiver ativo
incrementa a FLAG para mudar nivel
Fim_Start:                  RTI


LVL:                        INC M[FLAG_I2]
                            RTI
;--------------------------------------------------------------------------------
;--------------------------------------------------------------------------------
;                                                TEMPORIZADOR / RELOGIO

;--------------------------------------------------------------------------------
;--------------------------------------------------------------------------------

;Rotina do temporizador
Temp_Cont:                  PUSH R7

                            DEC M[Contador_gravidade]
                            DEC M[Contador_avanca_tubos]

                            MOV R7, TEMP_AMOUNT  ;Instrucoes que colocam o
temporizador de 0,1s em 0,1s
                            MOV M[TEMP], R7
                            MOV M[TEMP_ENABLE], R7

                            POP R7

                            RTI

;--------------------------------------------------------------------------------
;--------------------------------------------------------------------------------
;                                                  ROTINAS GERAIS

;--------------------------------------------------------------------------------
```

```
---------------------------------
;--------------------------------------------------------------------------------
---------------------------------

;Inicializacao cursor
Inicializa_CUR:                     MOV R7, CURSOR_INIT
                                MOV M[CURSOR], R7

                                RET

;Rotina que Escreve Caracter;
;ENTRADAS: R3 (simbolo a escrever)
EscCar:                         MOV M[WINDOW_WRITER], R3
                                RET

;Rotina que escreve string
;ENTRADAS: R1 (posicao do cursor) ; R2 (string a escrever)

EscString:                      MOV R3, M[R2]
                                CMP R3, FIM_TEXTO
                                BR.Z FimEsc
                                CALL EscCar
                                INC R1
                                MOV M[CURSOR], R1
                                INC R2
                                BR EscString

FimEsc:                         RET


;--------------------------------------------------------------------------------
---------------------------------
;--------------------------------------------------------------------------------
---------------------------------
;                                                                          LEDS

;--------------------------------------------------------------------------------
---------------------------------
;--------------------------------------------------------------------------------
---------------------------------

;Rotina que escreve e atualiza os LEDS
LED_ON:                             PUSH R1
                                PUSH R2

                                MOV R1, M[VALOR_LVL]

                                CMP R1, VALOR_LVL_1
                                BR.NZ Next_LED_1
                                MOV R2, F000h
                                MOV M[LEDS], R2
                                BR Fim_LEDS

Next_LED_1:                     CMP R1, VALOR_LVL_2
                                BR.NZ Next_LED_2
                                MOV R2, FF00h
                                MOV M[LEDS], R2
                                BR Fim_LEDS

Next_LED_2:                     CMP R1, VALOR_LVL_3
                                BR.NZ Next_LED_3
                                MOV R2, FFF0h
                                MOV M[LEDS], R2
                                BR Fim_LEDS
```

```
Next_LED_3:                 MOV R2, FFFFh
                            MOV M[LEDS], R2

Fim_LEDS:                   POP R2
                            POP R1

                                                    RET

;-------------------------------------------------------------------------------
---------------------------------
;-------------------------------------------------------------------------------
---------------------------------
;                                                   DISPLAY 7 SEGMENTOS

;-------------------------------------------------------------------------------
---------------------------------
;-------------------------------------------------------------------------------
---------------------------------

;Rotina para escrever no display de 7 segmentos
Display:                    PUSH R1
                            PUSH R2
                            PUSH R3

                            INC M[NUMERO_OBSTACULOS]

                            MOV R1, M[NUMERO_OBSTACULOS]
                            MOV R2, DISPLAY_7
                            MOV R3, 10
                            DIV R1, R3
                            MOV M[R2], R3

                            INC R2
                            MOV R3, 10
                            DIV R1, R3
                            MOV M[R2], R3

                            INC R2
                            MOV R3, 10
                            DIV R1, R3
                            MOV M[R2], R3

                            INC R2
                            MOV M[R2], R1

                            POP R3
                            POP R2
                            POP R1

                            RET

;-------------------------------------------------------------------------------
---------------------------------
;-------------------------------------------------------------------------------
---------------------------------
;                                                       LCD

;-------------------------------------------------------------------------------
---------------------------------
;-------------------------------------------------------------------------------
---------------------------------

;Rotina para escrever no LCD 'distancia'
```

```
LCD:                            PUSH R1
                                PUSH R2
                                PUSH R3

                                MOV R1, Distancia
                                MOV R2, 8000h
                                MOV R4, FIM_TEXTO

Ciclo_LCD:                      CMP M[R1], R3
                                BR.Z Fim_LCD
                                MOV R3, M[R1]
                                MOV M[LCD_CONTROL], R2
                                MOV M[LCD_WRITE], R3
                                MOV R3, FIM_TEXTO
                                INC R1
                                INC R2
                                BR Ciclo_LCD

Fim_LCD:                        POP R3
                                POP R2
                                POP R1

                                RET

;Rotina para atualizar no LCD o score
;OBS: Para poupar processamento a rotina so e chamada quando e necessario
atualizar o valor do score
LCD_SCORE:                      PUSH R1
                                PUSH R2
                                PUSH R3
                                PUSH R4

                                INC M[VALOR_DISTANCIA_LCD]  ;incrementa o valor da
distancia

                                MOV R1, 800Fh
                                MOV R2, M[VALOR_DISTANCIA_LCD]
                                MOV R3, R0

LCD_SCORE_CICLO:                CMP R3, 4
                                BR.Z Fim_LCD_SCORE
                                MOV R4, 10
                                DIV R2, R4
                                ADD R4, 48
                                MOV M[LCD_CONTROL], R1
                                MOV M[LCD_WRITE], R4
                                DEC R1
                                INC R3
                                BR LCD_SCORE_CICLO

Fim_LCD_SCORE:                        POP R4
                                POP R3
                                POP R2
                                POP R1

                                RET

;--------------------------------------------------------------------------------
---------------------------------
;--------------------------------------------------------------------------------
---------------------------------
;                                                       MENSAGENS

;--------------------------------------------------------------------------------
```

```
---------------------------------
;--------------------------------------------------------------------------------
---------------------------------

;Rotina que identifica a string a ser escrita (mensagem de inicio linha 12)
Texto_1:                        PUSH R1
                                PUSH R2
                                PUSH R7

                                MOV R7, VarT1_Cursor
                                MOV M[CURSOR], R7
                                MOV R1, VarT1_Cursor
                                MOV R2, Var_Texto_1

                                CALL EscString

                                POP R7
                                POP R2
                                POP R1

                                RET

;Rotina que identifica a string a ser escrita (mensagem de inicio linha 14)
Texto_2:                        PUSH R1
                                PUSH R2
                                PUSH R7

                                MOV R7, VarT2_Cursor
                                MOV M[CURSOR], R7
                                MOV R1, VarT2_Cursor
                                MOV R2, Var_Texto_2

                                CALL EscString

                                POP R7
                                POP R2
                                POP R1

                                RET

;Rotina que identifica a string a ser escrita (mensagem de FIM linha 12)
Texto_3:                        PUSH R1
                                PUSH R2
                                PUSH R7

                                MOV R7, VarT3_Cursor
                                MOV M[CURSOR], R7
                                MOV R1, VarT3_Cursor
                                MOV R2, Var_Texto_3

                                CALL EscString

                                POP R7
                                POP R2
                                POP R1

                                RET

;Rotina que identifica a string a ser escrita (mensagem de FIM linha 14)
Texto_4:                        PUSH R1
                                PUSH R2
                                PUSH R7
                                PUSH R3
```

```
                            MOV R7, VarT4_Cursor ;parte para escrever
'Pontuacao maxima:'
                            MOV M[CURSOR], R7
                            MOV R1, VarT4_Cursor
                            MOV R2, Var_Texto_4

                            CALL EscString

                            MOV R7, Var_T4_Cursor_2

                            MOV R1, M[NUMERO_OBSTACULOS]   ;parte para
escrever score
                            MOV R3, 10
                            DIV R1, R3
                            ADD R3, 48

                            MOV M[CURSOR], R7
                            CALL EscCar

                            DEC R7
                            MOV R3, 10
                            DIV R1, R3
                            ADD R3, 48

                            MOV M[CURSOR], R7

                            CALL EscCar

                            DEC R7
                            MOV R3, 10
                            DIV R1, R3
                            ADD R3, 48

                            MOV M[CURSOR], R7

                            CALL EscCar

                            DEC R7
                            MOV R3, R1
                            ADD R3, 48
                            MOV M[CURSOR], R7
                            CALL EscCar

                            POP R3
                            POP R7
                            POP R2
                            POP R1

                            RET

;--------------------------------------------------------------------------------
---------------------------------
;--------------------------------------------------------------------------------
---------------------------------
;                                                       CENARIO

;--------------------------------------------------------------------------------
---------------------------------
;--------------------------------------------------------------------------------
---------------------------------

;Rotina para limpar o mapa completo
Limpa:                              PUSH R7
                            PUSH R3
```

```
                            MOV R7, R0
                            MOV M[CURSOR], R7
                            MOV R3, CLEAN
CicloLimpa:                 CALL EscCar
                            INC R7
                            MOV M[CURSOR], R7
                            CMP R7, 174Fh
                            BR.NZ CicloLimpa

                            POP R3
                            POP R7

                            RET

;Rotina que posiciona o cursor e chama a rotina de escrita da margem horizontal
(limite superior)
limite_superior:            PUSH R1
                            PUSH R2
                            PUSH R3
                            PUSH R7

                            MOV R7, M[Posicao_Cursor_Hor]     ;Posiciona
Cursor
                            MOV M[CURSOR], R7
                            MOV R1, M[Posicao_Cursor_Hor]
                            MOV R2, 004Eh
                            MOV R3, Simbolo_Horizontal
                            CALL Esc_Margem_Hor      ;Rotina de escrita Margem
Horizontal

                            POP R7
                            POP R3
                            POP R2
                            POP R1

                            RET

;Rotina que posiciona o cursor e chama a rotina de escrita da margem horizontal
(limite inferior)
limite_inferior:            PUSH R1
                            PUSH R2
                            PUSH R3
                            PUSH R6

                            MOV R6, M[Posicao_Cursor_Hor] ;Posiciona Cursor
                            ADD R6, 1700h
                            MOV M[CURSOR], R6
                            MOV R1, R6
                            MOV R2, 004Eh
                            MOV R3, Simbolo_Horizontal
                            CALL Esc_Margem_Hor ;Rotina de escrita Margem
Horizontal

                            POP R6
                            POP R3
                            POP R2
                            POP R1

                            RET

;Rotina de escrita da margem horizontal
;ENTRADAS: R1 (Posicao Cursor); R2 (numero de iteracoes); R3 (simbolo de
escrita)
```

```
Esc_Margem_Hor:                 CALL EscCar
                            INC R1
                            MOV M[CURSOR], R1
                            DEC R2
                            CMP R2, R0
                            BR.NZ Esc_Margem_Hor

                            RET

;--------------------------------------------------------------------------------
---------------------------------
;--------------------------------------------------------------------------------
---------------------------------
;                                               PASSARO

;--------------------------------------------------------------------------------
---------------------------------
;--------------------------------------------------------------------------------
---------------------------------

;Rotina para escrver o passaro na posicao inicial
escreve_passaro_inicio: PUSH R3
                            PUSH R7

                            MOV R7, passaro_inicio_cursor
                            MOV M[CURSOR], R7
                            MOV R3, simbolo_passaro_1
                            CALL EscCar
                            INC R7
                            MOV M[CURSOR], R7
                            MOV R3 , simbolo_passaro_2
                            CALL EscCar

                            POP R7
                            POP R3

                            RET

;Rotina para escrever o passaro
;ENTRADAS: R7 (posicao do cursor)
escreve_passaro:            PUSH R7
                            PUSH R3

                            MOV M[CURSOR], R7
                            MOV R3, simbolo_passaro_1
                            CALL EscCar
                            INC R7
                            MOV M[CURSOR], R7
                            MOV R3 , simbolo_passaro_2
                            CALL EscCar

                            POP R3
                            POP R7

                            RET

;Rotina para limpar passaro
;ENTRADAS: R7 (Posicao do cursor)

limpa_passaro:                  PUSH R7
                            PUSH R3

                            MOV M[CURSOR], R7 ;Parte para limpar
                            MOV R3, CLEAN
```

```
                              CALL EscCar
                              INC R7
                              MOV M[CURSOR], R7
                              CALL EscCar

                              POP R3
                              POP R7
                              RET



;Rotina para subir o passaro uma posicao quando o interuptor I0 for ativo
Sobe_passaro:                       PUSH R3
                              PUSH R7
                              PUSH R5
                              DSI

                              MOV R7, M[CURSOR_PASSARO] ;Parte para Limpar

                              CMP R7, 0114h           ;Limite superior
                              BR.Z Fim_Sobe_passaro

                              CALL limpa_passaro

                              MOV R7, M[CURSOR_PASSARO] ;Parte para escrever
                              SUB R7, 0100h
                              MOV M[CURSOR_PASSARO], R7    ;atualiza posicao do
passaro

                              CALL escreve_passaro

Fim_Sobe_passaro:             MOV M[FLAG_I0], R0

                              MOV R5, 0004h
                              MOV M[Contador_gravidade], R5      ;reinicia
contador da gravidade
                              MOV M[Contador_gravidade_inicial], R5

                              ENI
                              POP R5
                              POP R7
                              POP R3

                              RET

;Rotina para aplicar gravidade/queda do passaro

Gravidade:                          PUSH R3
                              PUSH R7
                              PUSH R5
                              DSI

                              MOV R7, M[CURSOR_PASSARO] ;posiciona cursor


                              CMP R7 , 1614h
                              CALL.Z Inc_Flag_CRASH   ;testa o limite inferior

                              CALL limpa_passaro

                              MOV R7, M[CURSOR_PASSARO] ;Parte para escrever
                              ADD R7, 0100h
                              MOV M[CURSOR], R7
```

```
                              MOV M[CURSOR_PASSARO], R7 ;atualiza cursor do
passaro

                              CALL escreve_passaro


Fim_gravidade:                      MOV R5, M[Contador_gravidade_inicial]
;Decrementa o limite do contador
                              CMP R5, 0001h           ;limite do contador
                              BR.Z Fim_gravidade_2
                              DEC R5

Fim_gravidade_2:              MOV M[Contador_gravidade], R5 ;Coloca (novo) valor
no contador
                              MOV M[Contador_gravidade_inicial], R5

                              ENI
                              POP R5
                              POP R7
                                    POP R3

                              RET

;-------------------------------------------------------------------------------
---------------------------------
;-------------------------------------------------------------------------------
---------------------------------
;                                                       TUBOS

;-------------------------------------------------------------------------------
---------------------------------
;-------------------------------------------------------------------------------
---------------------------------

;Rotina para colocar na tabela das posicoes do tubo o valor inicial

;OBS: Usa Ponteiro_tabela que e a posicao (endereco) da tabela
;OBS: Coloca na tabela as duas coordenadas  (linha da posicao_porta, coluna da
posicao do tubo)
CriaTubos:                    PUSH R7
                              PUSH R3
                              PUSH R5
                              PUSH R4

                              CALL Aleatorio          ;gera valor aleatorio

                              MOV R3, M[Mascara_cursor]
                              MOV R7, M[Ponteiro_tabela]

                              MOV R4, 004Eh           ;atualiza tabela com a
coluna
                              ADD R4, R3 ;junta as duas coordenadas

                              MOV M[R7], R4

                              MOV R3, 000Eh
                              CMP M[Var_est_cria_tubos], R3
                              BR.NZ Fim_cria_tubos

                              BR    Fim_cria_tubos_2

Fim_cria_tubos:                     INC M[Var_est_cria_tubos]

                              INC M[Ponteiro_tabela]
```

```
Fim_cria_tubos_2:            MOV R5, 0006h     ;reinicia o contador para criar
tubos
                             MOV M[Var_est_cria_tubos_MAIN], R5

                             POP R4
                             POP R5
                             POP R3
                             POP R7

                             RET

;Rotina para mover tubos (escreve os tubos na posicao seguinte)
;           ENTRADAS: R7
AvancaTubo:                  PUSH R3
                             PUSH R2
                             PUSH R5
                             PUSH R7
                             PUSH R1
                             DSI

                             SUB R7, 0001h   ;avanca o cursor uma posicao para
a esquerda (1 coluna)

                             MOV R2, R7 ;duplica
                             MOV R1, R7 ;serve para avaliar se ja chegou ao fim

                             AND  R1, 00FFh
                             ADD   R1, 1700h ;para testar limite inferior

                             AND  R7, 00FFh         ;seleciona os bits da
posicao do tubo
                             ADD  R7, 0100h    ;R7 FICA COM POSICAO DO TUBO
(posicao do cursor)

                             MOV R3, simbolo_tubos

Ciclo_avancatubo:            MOV M[CURSOR], R7

                             CALL EscCar

                             ADD R7, 0100h     ;Sobe o cursor (escreve de baixo
para cima)

                             CMP R2, R7
                             BR.NZ Ciclo_avancatubo

                             ADD R7, 0500h        ;para gerar porta

Ciclo_avancatubo_2:                MOV M[CURSOR], R7


                             CALL EscCar

                             ADD R7, 0100h

                             CMP R7, R1  ;compara com a variavel de controlo
                             BR.NZ Ciclo_avancatubo_2

                             ENI
                             POP R1
                             POP R7
                             POP R5
                             POP R2
```

```
                            POP R3

                            RET

;Rotina para limpar tubos
;           ENTRADAS: R7
Limpatubo:                  PUSH R3
                            PUSH R2       ;valor da posicao da porta
                            PUSH R7       ;valor do cursor
                            PUSH R1       ;valor para testar se ja chegou ao
limite inferior
                            DSI

                            MOV R2, R7  ;duplica
                            MOV R1, R7  ;serve para avaliar se já chegou ao
fim (duplica)

                            AND  R1, 00FFh    ;seleciona coordenada
                            ADD   R1, 1700h   ;para testar limite inferior

                            AND  R7, 00FFh      ;seleciona os bits da posicao
do tubo
                            ADD  R7, 0100h    ;R7 FICA COM POSICAO DO TUBO
(posicao do cursor)

                            MOV R3, CLEAN
Ciclo_Limpa_tubos:                MOV M[CURSOR], R7

                            CALL EscCar

                            ADD R7, 0100h

                            CMP R2, R7  ;compara a posicao do cursor atual com
a posicao da porta(que e a primeira posicao livre)
                            BR.NZ Ciclo_Limpa_tubos

                            ADD R7, 0500h        ;cria espaco para a porta

Ciclo_limpatubo_2:                MOV M[CURSOR], R7


                            CALL EscCar

                            ADD R7, 0100h

                            CMP R7, R1  ;compara com a variavel de controlo
                            BR.NZ Ciclo_limpatubo_2

                                                  POP R1
                                                  POP R7
                                                  POP R2
                                                  POP R3

                                                  ENI
                                                  RET

;Rotina para avancar os tubos (PRINCIPAL)
AvancaTubos:                                            PUSH R5
                                                  PUSH R4     ;guarda numero
para percorrer toda a tabela
                                                  PUSH R6     ;fica com a
posicao da tabela (posicao_tubos)
                                                  PUSH R7     ;guarda a
posicao do mapa(cursor)
```

```
                                                    PUSH R2
                                                    CALL LCD_SCORE     ;valor
que guarda a distancia percorrida
                                                    MOV R6, Posicao_tubos
;primeira posicao da tabela
                                                    MOV R4, R0
Ciclo_avanca_tubos:                                       MOV R7, M[R6]   ;poe
o cursor na linha da tabela correta
                                                    CMP R7, 0000h
                                                    BR.Z Mini_ciclo_2
                                                    CALL Limpatubo
                                                    MOV R7, M[R6]
                                                    AND R7, 00FFh
      ;seleciona so os bits da coluna
                                                    MOV R2, 0001h
                                                    CMP  R7, 0014h    ;se
estiver na coluna do passaro incrementa nr obs ultrapassados
                                                    CALL.Z Display
                                                    CMP R7 , R2 ;Se a posicao
for na coluna 1 nao avanca mais
                                                    BR.NZ Mini_ciclo
                                                    MOV R7, M[R6]
                                                    AND R7, 00FFh
      ;seleciona so os bits da coluna
                                                    CALL Aleatorio
      ;gera novo valor aleatorio
                                                    MOV R2, M[Mascara_cursor]
                                                    ADD R2, 004Eh     ;reinicia
a posicao da tabela com o valor inicial
                                                    MOV M[R6], R2
                                                    BR    Mini_ciclo_2 ;Nao
escreve
Mini_ciclo:                              MOV R7, M[R6]
                                                    CALL AvancaTubo
                                                    MOV R7, M[R6]     ;Atualiza
posição da tabela com o novo valor (-1 coluna)
                                                    SUB R7, 0001h
                                                    MOV M[R6], R7
Mini_ciclo_2:                            INC R6
                                                    INC R4
                                                    CMP R4, 000Dh     ;verifica
se ja percorreu toda a tabela
                                                    JMP.NZ Ciclo_avanca_tubos
                                                    MOV   R5, M[VALOR_LVL]
;Reset do contador
                                                    MOV
      M[Contador_avanca_tubos], R5
```

```
                                                DEC
M[Var_est_cria_tubos_MAIN]

                                                POP R2
                                                POP R7
                                                POP R6
                                                POP R4

                                                POP R5

                                                RET
;-------------------------------------------------------------------------------
---------------------------------
;-------------------------------------------------------------------------------
---------------------------------
;                                                ALEATORIO

;-------------------------------------------------------------------------------
---------------------------------
;-------------------------------------------------------------------------------
---------------------------------

;Rotina que gera valor aleatorio
;OBS: Esta rotina gera a primeira posicao livre (primeira posicao da porta) do
tubo
;OBS:Sao gerados 12 valores porque se admitiu que nao apareceriam portas nas 3
posicoes mais proximas dos extremos
Aleatorio:                              PUSH  R6
                                                PUSH  R3
                                                PUSH  R2

                                                MOV   R6,
M[Mascara_aleatoria_inicial]
                                                SHR         R6, 1
                                                BR.C        Aleatorio_1
;testa bit menos significativo

                                                MOV R6,
M[Mascara_aleatoria_inicial]
                                                MOV R3,
Valor_divisao_aleatorio  ;valor para dividir a mascara (12)
                                                DIV   R6, R3
                                                ADD   R3, 0004h  ;para
compensar o limite de cima mais 3 posicoes
                                                SHL   R3, 8     ;para por o
valor nas coordenadas das linhas
                                                MOV M[Mascara_cursor], R3

                                                MOV R2, Mascara_aleatorio
                                                XOR
      M[Mascara_aleatoria_inicial], R2    ;Atualiza proximo valor da mascara
                                                ROR
      M[Mascara_aleatoria_inicial], 1

                                                BR    Fim_aleatorio

Aleatorio_1:                            MOV   R6, M[Mascara_aleatoria_inicial]
                                                MOV   R3,
Valor_divisao_aleatorio
                                                DIV         R6, R3
                                                ADD         R3, 0004h
                                                SHL         R3, 8
                                                MOV   M[Mascara_cursor], R3
```

```
                                                ROR
      M[Mascara_aleatoria_inicial], 1   ;atualiza o proximo valor da mascara

Fim_aleatorio:                          POP   R2
                                                POP   R3
                                                POP   R6

                                                RET

;Rotina que conta o numero de obstaculos ultrapassados
Inc_contador_obs:             INC M[NUMERO_OBSTACULOS]
                                                RET

;--------------------------------------------------------------------------------
---------------------------------
;--------------------------------------------------------------------------------
---------------------------------
;                                                COLISOES

;--------------------------------------------------------------------------------
---------------------------------
;--------------------------------------------------------------------------------
---------------------------------

;Rotina que incrementa FLAG de choque
Inc_Flag_CRASH:                     INC M[FLAG_CRASH]
                                                RET

;Rotina que testa colisoes
Testa_colisao:                          PUSH R2
                                                PUSH R3
                                                PUSH R6
                                                PUSH R5   ;para percorrer
as 13 posicoes da tabela

                                                MOV R3, Posicao_tubos
                                                MOV R5, R0

Ciclo_colisao_MAIN:                 MOV R2, M[R3]
                                                MOV R6, R2

                                                AND R6, 00FFh
                                                CMP R6, 0014h     ;se
estiver na coluna do passaro verifica as linhas
                                                BR.NZ Ciclo_colisao

                                                MOV R6, R2
                                                ADD R6, 0400h ;variavel de
controlo do ciclo

Ciclo_testa_colisao:                CMP M[CURSOR_PASSARO], R2
                                                BR.Z
Fim_testa_colisao    ;se estiver na posicao salta para o fim e nao faz nada

                                                ADD R2, 0100h
      ;adiciona uma linha

                                                CMP R2, R6
                                                BR.NZ Ciclo_testa_colisao

                                                CALL Inc_Flag_CRASH ;se
passou pelas 5 posicoes  e nao detetou entao o passaro chocou
```

```
                                                    BR Fim_testa_colisao

Ciclo_colisao:                          INC R5
                                                    INC R3

                                                    CMP R5, 000Dh     ;verifica
se ja chegou ao fim da tabela
                                                    BR.NZ Ciclo_colisao_MAIN

Fim_testa_colisao:                  POP R5
                                                    POP R6
                                                    POP R3
                                                    POP R2

                                                    RET

;-------------------------------------------------------------------------------
---------------------------------
;-------------------------------------------------------------------------------
---------------------------------
;                                                    NIVEIS

;-------------------------------------------------------------------------------
---------------------------------
;-------------------------------------------------------------------------------
---------------------------------

;Rotina para aumentar o nivel de jogo (interrupcao I2)
Aumenta_lvl:                            PUSH R2

                                                    MOV R2, LVL_Maximo
                                                    CMP M[VALOR_LVL], R2
;verifica se ja esta no lvl maximo
                                                    BR.Z Fim_aumenta_LVL

                                                    DEC
M[Contador_avanca_tubos]
                                                    DEC M[VALOR_LVL]
                                                    CALL LED_ON       ;atualiza
os LEDS

Fim_aumenta_LVL:              MOV M[FLAG_I2], R0

                                                    POP R2

                                                    RET


;Rotina para diminuir o nivel de jogo (interrupcao I1)
Diminui_lvl:                                  PUSH R2

                                                    MOV R2, LVL_Minimo
                                                    CMP M[VALOR_LVL], R2
;verifica se ja esta no lvl maximo
                                                    BR.Z Fim_diminui_LVL

                                                    INC
M[Contador_avanca_tubos]
                                                    INC M[VALOR_LVL]
                                                    CALL LED_ON
            ;atualiza os LEDS

Fim_diminui_LVL:                    MOV M[FLAG_I1], R0
```

```
                                                POP R2

                                                RET

;--------------------------------------------------------------------------------
--------------------------------
;--------------------------------------------------------------------------------
--------------------------------
;                                               PROGRAMA PRINCIPAL

;--------------------------------------------------------------------------------
--------------------------------
;--------------------------------------------------------------------------------
--------------------------------

Inicio:                                         MOV R7, SP_INICIAL
                                        MOV SP, R7

Print:                                          CALL Inicializa_CUR

                                        CALL Texto_1
                                        CALL Texto_2


Waiting:                                MOV R7, INT1MASK        ;Ativa
Interrupcao
                                        MOV M [INTMASK], R7

                                        MOV M [GAME_STARTED], R0
;flag do estado de jogo
                                        MOV R1, M[GAME_STARTED]
                                        ENI


WaitingLOOP:                                    INC
      M[Mascara_aleatoria_inicial]      ;gera valor inicial aleatorio
                                        CMP   M[GAME_STARTED], R1
      ;Deteta interrupcao
                                        BR.Z  WaitingLOOP

Desenha_Jogo:                                   CALL Limpa

                                        CALL limite_superior
                                        CALL limite_inferior

                                        CALL escreve_passaro_inicio

Temp_INIT:                              MOV   R7, INTMAINMASK         ;Ativa
interrupcoes
                                        MOV   M[INTMASK], R7

                                        MOV   R7, TEMP_AMOUNT
                                        MOV   M[TEMP], R7
                                        MOV   M[TEMP_ENABLE], R7
                                        CALL LED_ON
      ;Inicializa os LEDS
                                                CALL LCD

Ciclo_jogo:                             CMP R0, M[FLAG_I0]       ;Sobe passaro
                                        CALL.NZ Sobe_passaro

                                        CMP R0, M[FLAG_I1]            ;Testa
mudancas de lvl
```

```
                                    CALL.NZ Diminui_lvl

                                    CMP R0, M[FLAG_I2]
      ;Testa mudancas de lvl
                                    CALL.NZ Aumenta_lvl

                                    CMP M[Contador_gravidade], R0
      ;Aceleracao / gravidade
                                    CALL.Z Gravidade

                                    CMP M[Var_est_cria_tubos_MAIN], R0
;Cria tubos
                                    CALL.Z CriaTubos

                                    CMP M[Contador_avanca_tubos], R0
      ;Avanca tubos
                                    CALL.Z AvancaTubos

                                    CALL Testa_colisao
                                    CMP M[FLAG_CRASH], R0        ;Deteta
colisao
                                    BR.NZ Fim_jogo

                                    JMP Ciclo_jogo

Fim_jogo:                           MOV R7, 0         ;para temporizador
                                    MOV   M[TEMP_ENABLE], R7

                                    CALL Limpa

                                    CALL Texto_3
                                    CALL Texto_4           ;escreve score

Fim:                                JMP  Fim
```