

| jun 02, 17 22:22 C:\Users\User\Desktop\p3print_win\mastermind.as Page 1/19 | | | |
|--|------|--|--|
| ;===== | | | |
| ; Programa Demol.as | | | |
| ; | | | |
| ; Descricao: Demonstracao da utilização da janela de texto | | | |
| ; | | | |
| ; Autor: Nuno Horta | | | |
| ; Data: 28/05/2013 | | | |
| ;===== Ultima Alteracao:28/05/2013 ===== | | | |
| ;===== | | | |
| ; ZONA I: Definicao de constantes | | | |
| ; Pseudo-instrucao : EQU | | | |
| ;===== | | | |
| ; STACK POINTER | | | |
| SP_INICIAL | EQU | FDFh | |
| ; I/O a partir de FF00h | | | |
| IO_CURSOR | EQU | FFFCh | |
| IO_WRITE | EQU | FFFEh | |
| IO_STAT | EQU | FFFDh | |
| DISP_1 | EQU | FFF0h | |
| DISP_2 | EQU | FFF1h | |
| DISP_3 | EQU | FFF2h | |
| DISP_4 | EQU | FFF3h | |
| TEMP_UNIT | EQU | FFF6h | |
| TEMP_STATE | EQU | FFF7h | |
| MASCARA_END | EQU | FFFAh | |
| MASCARA_INT | EQU | 8001h | |
| ;FFFAh gera ebbb | | | |
| TAB_INT0 | EQU | FE00h | |
| TAB_INT1 | EQU | FE01h | |
| TAB_TIME | EQU | FE0Fh | |
| | | | |
| LCD_WRITE | EQU | FFF5h | |
| LCD_CURSOR | EQU | FFF4h | |
| | | | |
| READ | EQU | FFFFh | |
| XY_INICIAL | EQU | 0614h | |
| N1 | EQU | 0123h | |
| M1 | EQU | 0006h | |
| FIM_TEXTO | EQU | '@' | |
| ;===== | | | |
| ; ZONA II: Definicao de variaveis | | | |
| ; Pseudo-instrucoes : WORD - palavra (16 bits) | | | |
| ; STR - sequencia de caracteres. | | | |
| ; Cada caracter ocupa 1 palavra | | | |
| ;===== | | | |
| ORIG | | 8000h | |
| VarTexto1 | STR | '***** Mastermind *****' | |
| VarTexto2 | STR | ' | |
| VarTexto3 | STR | ' | |
| VarTexto4 | STR | ' | |
| VarTexto5 | STR | ' | |
| VarTexto6 | STR | ' | |
| VarTexto7 | STR | ' | |
| VarTexto8 | STR | ' | |
| VarTexto9 | STR | ' | |
| VarTexto10 | STR | ' | |
| VarTexto11 | STR | ' | |
| VarTexto12 | STR | ' | |
| VarTexto13 | STR | ' | |
| VarTexto14 | STR | ' | |
| VarTexto15 | STR | ' | |
| VarTexto16 | STR | ' | |
| VarTexto17 | STR | ' | |
| VarTexto18 | STR | ' | |
| VarTexto19 | STR | ' | |
| VarTexto20 | STR | ' | |
| VarTexto21 | STR | ' | |
| VarTexto22 | STR | ' | |
| VarTexto23 | STR | ' | |
| VarTexto24 | STR | ' | |
| VarTexto25 | STR | ' | |
| VarTexto26 | STR | ' | |
| VarTexto27 | STR | ' | |
| VarTexto28 | STR | ' | |
| VarTexto29 | STR | ' | |
| VarTexto30 | STR | ' | |
| ValidColors | STR | 'b', 'p', 'e', 'v', 'z', 'a' | |
| randNumb | STR | 0000h, 0000h, 0000h, 0000h, 0000h, 0000h, 0000h, 0000h | |
| randNumbcopia | STR | 0000h, 0000h, 0000h, 0000h, 0000h, 0000h, 0000h, 0000h | |
| userinput | STR | 0000h, 0000h, 0000h, 0000h, 0000h, 0000h, 0000h, 0000h | |
| validacao_output | STR | '', '', '', '', '', '', '', '' | |
| jogador1 | STR | 0000h, 0000h | |
| jogador2 | STR | 0000h, 0000h | |
| jogador3 | STR | 0000h, 0000h | |
| totaltentativas | WORD | 0000h | |
| totaljogadores | WORD | 0000h | |
| jogador_playing | WORD | 0000h | |
| fimjogo | WORD | 0000h | |

| jun 02, 17 22:22 C:\Users\User\Desktop\p3print_win\mastermind.as Page 2/19 | | | |
|--|------|--|--|
| VarTexto4 | STR | ' | |
| VarTexto5 | STR | ' | |
| VarTexto6 | STR | ' | |
| VarTexto7 | STR | ' | |
| VarTexto8 | STR | ' | |
| VarTexto9 | STR | ' | |
| VarTexto10 | STR | ' | |
| VarTexto11 | STR | ' | |
| VarTexto12 | STR | ' | |
| VarTexto13 | STR | ' | |
| VarTexto14 | STR | ' | |
| VarTexto15 | STR | ' | |
| VarTexto16 | STR | ' | |
| VarTexto17 | STR | ' | |
| VarTexto18 | STR | ' | |
| VarTexto19 | STR | ' | |
| VarTexto20 | STR | ' | |
| VarTexto21 | STR | ' | |
| VarTexto22 | STR | ' | |
| VarTexto23 | STR | ' | |
| VarTexto24 | STR | ' | |
| VarTexto25 | STR | ' | |
| VarTexto26 | STR | ' | |
| VarTexto27 | STR | ' | |
| VarTexto28 | STR | ' | |
| VarTexto29 | STR | ' | |
| VarTexto30 | STR | ' | |
| ValidColors | STR | 'b', 'p', 'e', 'v', 'z', 'a' | |
| randNumb | STR | 0000h, 0000h, 0000h, 0000h, 0000h, 0000h, 0000h, 0000h | |
| randNumbcopia | STR | 0000h, 0000h, 0000h, 0000h, 0000h, 0000h, 0000h, 0000h | |
| userinput | STR | 0000h, 0000h, 0000h, 0000h, 0000h, 0000h, 0000h, 0000h | |
| validacao_output | STR | '', '', '', '', '', '', '', '' | |
| jogador1 | STR | 0000h, 0000h | |
| jogador2 | STR | 0000h, 0000h | |
| jogador3 | STR | 0000h, 0000h | |
| totaltentativas | WORD | 0000h | |
| totaljogadores | WORD | 0000h | |
| jogador_playing | WORD | 0000h | |
| fimjogo | WORD | 0000h | |

jun 02, 17 22:22 C:\Users\User\Desktop\p3print_win\mastermind.as Page 3/19

```

ou o jogo ou quando se acabaram as tentativas, indicando o fim de jogo
tentativa      WORD      0000h
tempo          WORD      0000h ;tempo de jogo
tempo_init     WORD      0000h ; tempo inicial
numbjogadores  WORD      0000h ; número de jogadores inicializado a 0
numbcores      WORD      0000h ; número de peças usado para os niveis de dific
uldade
seed           WORD      0000h ; variável que guarda o valor da Seed

;=====
; ZONA III: Codigo
; conjunto de instrucoes Assembly, ordenadas de forma a realizar
; as funcoes pretendidas
;=====
ORIG 000h
JMP inicio

;=====
;Iniciação das Interrupções
;=====
INIT_INT:      PUSH R1
               MOV R1, RotinaInt0
               MOV M[TAB_INT0], R1
               POP R1
               RET

;=====
;Inicialização do Temporizador
;=====
INIT_TIME:     PUSH R1
               MOV R1, M[tempo_init]
               MOV M[tempo], R1
               MOV R1, 000Ah
               MOV M[TEMP_UNIT], R1 ;definir como tempo de con
tagem 1 s no porto correto
               MOV R1, 0001h
               MOV M[TEMP_STATE], R1 ;Iniciar o Timer
               MOV R1, RotinaTime
               MOV M[TAB_TIME], R1
               MOV R1, MASCARA_INT
               MOV M[MASCARA_END], R1
               POP R1
               RET

;=====
;Interrupção 0
;=====
RotinaInt0:    PUSH R1
               MOV R1, 1500h
               PUSH VarTexto28
               PUSH R1
               CALL EscString
               MOV R1, 1518h
               PUSH randNumb
               PUSH R1
               CALL EscString
               MOV R1, 0000h
               CMP M[jogador_playing], R1
               JMP.Z Salta1

```

jun 02, 17 22:22 C:\Users\User\Desktop\p3print_win\mastermind.as Page 4/19

```

               INC R1
               CMP M[jogador_playing], R1
               JMP.Z Salta2
               INC R1
               CMP M[jogador_playing], R1
               JMP.Z Salta3

Salta1:        POP R1
               PUSH FIMJOGA1
               BR FimInt0

Salta2:        POP R1
               PUSH FIMJOGA2
               BR FimInt0

Salta3:        POP R1
               PUSH FIMJOGA3
               BR FimInt0
FimInt0:       RTI

;=====
;Rotina que pede ao utilizador para premir c para continuar
;=====
PressEnter:    PUSH R1
               MOV R1, 1601h
               PUSH VarTexto27
               PUSH R1
               CALL EscString
CicloInput4:   CMP M[IO_STAT], R0
               BR.Z CicloInput4
               MOV R1, 'c'
               CMP M[READ], R1
               BR.NZ CicloInput4
               POP R1
               RET

;=====
;Interrupção Timer
;=====
RotinaTime:    PUSH R1
               PUSH R2
               PUSH R3
               MOV R1, 1
               SUB M[tempo], R1
               MOV R1, M[tempo]
               MOV R2, R1 ; registo auxiliar
               MOV R3, 100
               DIV R2, R3 ; ver o resto da divisã
o por 100 para mover para o display 3
               MOV M[DISP_3], R2 ; mover o resultado da div
isão por 100 para o disp_3
               MOV R2, 10
               DIV R3, R2
               MOV M[DISP_2], R3 ; mover o result
ado da divisão por 10 para o disp_2
               MOV M[DISP_1], R2 ; mover o resto da divisã
o por 10 para o disp_1
               CMP R1, 0000h ; ver se o tempo já aca
bou
               MOV R1, 000Ah
               MOV M[TEMP_UNIT], R1 ;definir como tempo de
contagem 1 s no porto correto
               MOV R1, 0001h
               MOV M[TEMP_STATE], R1 ;Iniciar o Timer
               MOV R1, 0000h
               CALL.Z RotinaInt0 ; mostrar a sequência g
erada e fazer jump para o fim
               POP R3

```

jun 02, 17 22:22 C:\Users\User\Desktop\p3print_win\mastermind.as Page 5/19

```

                POP R2
                POP R1
                RTI
;=====
;
; Receber uma seed- rotina de interrupção guardada no tempo
;=====
=
GetSeed: PUSH R1
        MOV R1, 0001h
        MOV M[TEMP_UNIT], R1      ; conta 100 ms de cada vez para recebe
r a seed
        MOV R1, 0001h
        MOV M[TEMP_STATE], R1     ; iniciar a contagem do tempo
        ;MOV R1, 0001h           ; somar 200 a cada 100ms à seed
        INC M[seed]      ;, R1
        POP R1
        RTI
;=====
;
; Inicialização da rotina de interrupção para receber a seed
;=====
=
InitSeed:  PUSH R1
        MOV R1, GetSeed
        MOV M[TAB_TIME], R1
        MOV R1, 0001h
        MOV M[TEMP_UNIT], R1
        MOV R1, 0001h
        MOV M[TEMP_STATE], R1
        MOV R1, 8000h
        MOV M[MASCARA_END], R1
        POP R1
        RET
;=====
;
; Escrita do menu inicial
;=====
MenuInicial: PUSH R1
            PUSH R2
            PUSH R3
            MOV R3, 0A05h
            PUSH VarTexto1
            PUSH R3
            CALL EscString
            ADD R3, 0200h
            PUSH VarTexto18
            PUSH R3
            CALL EscString
Ciclo3:    MOV R1, M[IO_STAT]
            CMP M[IO_STAT], R0
            BR.Z Ciclo3
            MOV R2, M[READ]
            CMP R2, 's'
            BR.NZ Ciclo3
            POP R3
            POP R2
            POP R1
            RET
;=====
;
; Escrita do Menu Final

```

jun 02, 17 22:22 C:\Users\User\Desktop\p3print_win\mastermind.as Page 6/19

```

;=====
;
MenuFinal: PUSH R1
        MOV R1, 0707h
        PUSH VarTexto29
        PUSH R1
        CALL EscString
        ADD R1, 0100h
        PUSH VarTexto30
        PUSH R1
        CALL EscString
        RET
;=====
;
; Receber o número de jogadores
;=====
=
MenuOpcoes: PUSH R1
        MOV R1, 002Ah
        MOV M[IO_CURSOR], R1
        MOV R1, 0000h
        PUSH VarTexto19
        PUSH R1
        CALL EscString
CicloInput1: CMP M[IO_STAT], R0
            BR.Z CicloInput1
            MOV R1, M[READ]
            CMP R1, '1'
            BR.Z Jogador1
            CMP R1, '2'
            BR.Z Jogador2
            CMP R1, '3'
            BR.Z Jogador3
            BR CicloInput1
Jogador1: MOV M[IO_WRITE], R1
            MOV R1, 0001h
            MOV M[numbjogadores], R1
            MOV M[totaljogadores], R1
            BR NumeroCores
Jogador2: MOV M[IO_WRITE], R1
            MOV R1, 0002h
            MOV M[numbjogadores], R1
            MOV M[totaljogadores], R1
            BR NumeroCores
Jogador3: MOV M[IO_WRITE], R1
            MOV R1, 0003h
            MOV M[numbjogadores], R1
            MOV M[totaljogadores], R1
            BR NumeroCores
NumeroCores: MOV R1, 0126h
            MOV M[IO_CURSOR], R1
            MOV R1, 0100h
            PUSH VarTexto23
            PUSH R1
            CALL EscString
CicloInput2: CMP M[IO_STAT], R0
            BR.Z CicloInput2
            MOV R1, M[READ]
            CMP R1, '4'

```

jun 02, 17 22:22 C:\Users\User\Desktop\p3print_win\mastermind.as Page 7/19

```

        BR.Z Cor4
        CMP R1, '5'
        BR.Z Cor5
        CMP R1, '6'
        BR.Z Cor6
        BR CicloInput2          ; se não tiver sido escolhida nenh
uma cor válida voltar a pedir um novo número

Cor4:MOV M[IO_WRITE], R1          ;escrever no ecrã o número de cores
        MOV R1, 0004h
        MOV M[numbcores],R1
        BR Escolhetempo
Cor5:MOV M[IO_WRITE], R1
        MOV R1, 0005h
        MOV M[numbcores], R1
        BR Escolhetempo
Cor6:MOV M[IO_WRITE], R1
        MOV R1, 0006h
        MOV M[numbcores], R1

Escolhetempo: MOV R1, 0200h
                PUSH VarTexto26
                PUSH R1
                CALL EscString
CicloInput3:   CMP M[IO_STAT], R0
                BR.Z CicloInput3
                MOV R1, M[READ]
                CMP R1, '1'
                BR.Z Tempo1
                CMP R1, '2'
                BR.Z Tempo2
                CMP R1, '4'
                BR.Z Tempo4
                BR CicloInput3
Tempo1: MOV R1, 003Ch
        MOV M[tempo_init], R1      ;60 segundos
        BR FimOpcoes
Tempo2: MOV R1, 0078h
        MOV M[tempo_init], R1      ; 120 segundos
        BR FimOpcoes
Tempo4: MOV R1, 00F0h
        MOV M[tempo_init], R1      ; 240 segundos
        BR FimOpcoes

FimOpcoes:POP R1
                RET
;=====
;Escrita do Tabuleiro
;=====
WRITE_BOARD:  PUSH R1
                MOV      R1, 0000h
                PUSH     VarTextol      ; Passagem de parametros pelo STACK
                PUSH     R1              ; Passagem de parametros pelo STACK
                CALL     EscString
                ADD      R1, 0200h
                PUSH     VarTexto2
                PUSH     R1
                CALL     EscString
                ADD      R1, 0100h
                PUSH     VarTexto3
                PUSH     R1
                CALL     EscString

```

jun 02, 17 22:22 C:\Users\User\Desktop\p3print_win\mastermind.as Page 8/19

```

                ADD      R1, 0100h
                PUSH     VarTexto4
                PUSH     R1
                CALL     EscString
                ADD      R1, 0100h
                PUSH     VarTexto5
                PUSH     R1
                CALL     EscString
                ADD      R1, 0100h
                PUSH     VarTexto6
                PUSH     R1
                CALL     EscString
                ADD      R1, 0100h
                PUSH     VarTexto7
                PUSH     R1
                CALL     EscString
                ADD      R1, 0100h
                PUSH     VarTexto8
                PUSH     R1
                CALL     EscString
                ADD      R1, 0100h
                PUSH     VarTexto9
                PUSH     R1
                CALL     EscString
                ADD      R1, 0100h
                PUSH     VarTextol0
                PUSH     R1
                CALL     EscString
                ADD      R1, 0100h
                PUSH     VarTextol1
                PUSH     R1
                CALL     EscString
                ADD      R1, 0100h
                PUSH     VarTextol2
                PUSH     R1
                CALL     EscString
                ADD      R1, 0100h
                PUSH     VarTextol3
                PUSH     R1
                CALL     EscString
                ADD      R1, 0100h
                PUSH     VarTextol4
                PUSH     R1
                CALL     EscString
                ADD      R1, 0200h
                PUSH     VarTextol5
                PUSH     R1
                CALL     EscString
                ADD      R1, 0100h
                PUSH     VarTextol6
                PUSH     R1
                CALL     EscString
                ADD      R1, 0100h
                PUSH     VarTextol7
                PUSH     R1
                CALL     EscString
                POP R1
                RET
;=====
;Escrever que o jogador 1 está a jogar
;=====
EscJogador1: PUSH R1

```

jun 02, 17 22:22 C:\Users\User\Desktop\p3print_win\mastermind.as Page 9/19

```

MOV R1, 0100h
PUSH VarTexto20
PUSH R1
CALL EscString
POP R1
RET

;=====
;Escrever que o jogador 2 está a jogar
;=====
EscJogador2: PUSH R1

MOV R1, 0100h
PUSH VarTexto21
PUSH R1
CALL EscString
POP R1
RET

;=====
;Escrever que o jogador 3 está a jogar
;=====
EscJogador3: PUSH R1

MOV R1, 0100h
PUSH VarTexto22
PUSH R1
CALL EscString
POP R1
RET

;=====
;Quando carrega na interrupção 0 apresenta a sequência gerada
;=====
;ORIG FE00h
;INT0 WORD APRESEQCORR
;RTI

;=====
; Gera e guarda os números aleatórios no vetor randnumb
;=====
GenerateRand: PUSH R1

PUSH R2
PUSH R3
PUSH R4
MOV R1, M[seed]
MOV R3, 0000h
MOV R4, 0000h

RANDOMNUMB: MOV R3, R1 ; R3 é utilizado de modo a não alterar o v
alor de R1

ROR R3 , 1
BR.C XOR1 ; Salta para XOR1 se NI0==1
SHR R1, 1
BR NEW_COLOR
XOR1: XOR R1, MASCARA_INT

NEW_COLOR: MOV R3, R1; R3 é utilizado de modo a não alterar o valor de R1
MOV R2, 0006h ; R2 é usado para gerar a cor usando
o resto da divisão
DIV R3, R2
CMP R2, 0000h

```

jun 02, 17 22:22 C:\Users\User\Desktop\p3print_win\mastermind.as Page 10/19

```

BR.Z B1
CMP R2, 0001h
BR.Z P1
CMP R2, 0002h
BR.Z E1
CMP R2, 0003h
BR.Z V1
CMP R2, 0004h
BR.Z Z1
CMP R2, 0005h
BR.Z A1
B1:MOV R2, 'b'
MOV M[R4+randNumb], R2
BR novorand
P1:MOV R2, 'p'
MOV M[R4+randNumb], R2
BR novorand
E1:MOV R2, 'e'
MOV M[R4+randNumb], R2
BR novorand
V1:MOV R2, 'v'
MOV M[R4+randNumb], R2
BR novorand
Z1:MOV R2, 'z'
MOV M[R4+randNumb], R2
BR novorand
A1:MOV R2, 'a'
MOV M[R4+randNumb], R2

novorand: INC R4
CMP R4, M[numbcores] ; ver se já foram geradas 4
cores

JMP.NZ RANDOMNUMB ; se não tiverem sido gera-se um no
vo número

MOV M[seed], R1
POP R4
POP R3
POP R2
POP R1

RET

;=====
; Escreve o vetor randNumb, com a sequência gerada
;=====
EscreveRandNumb: PUSH R1

MOV R1, 1601h
PUSH randNumb
PUSH R1
CALL EscString
POP R1
RET

;=====
;Guarda o que o utilizador escrever no vetor userInput e escrever, imediatament
e a seguir à leitura, no ecrã apenas se o input for de uma cor válida
;=====
User_Input: PUSH R1

PUSH R2
PUSH R3
PUSH R4
MOV R4, 0000h
ADD R6, 0100h

```

jun 02, 17 22:22 C:\Users\User\Desktop\p3print_win\mastermind.as Page 11/19

```

        PUSH R6                ; salvaguarda o valor de R5 para
passar para a próxima linha na tentativa seguinte, dado que R5 guarda a linha
        MOV M[IO_CURSOR], R6

NEW_INPUT:      MOV R2, 0000h
                MOV R3, 0000h    ; reset dos registos auxiliares

        TEST_INPUT:  CMP M[IO_STAT], R0
                    BR.Z TEST_INPUT

                        MOV R1, M[READ] ; guardar em R1 a cor introduzi
da
VALID_INPUT1:   MOV R3, R2

                        INC R2
                        CMP R1, M[R3+ValidColors]
                        BR.Z GUARDA
                        CMP R3, 0005h ; ver se ainda não foi comparado c
om todas as cores

                        BR.NZ VALID_INPUT1
                        BR NEW_INPUT; se já foi comparado com todas ass
cores e não for válido pede-se um novo input e restabelece-se os valores de R2 e
R3

GUARDA:         MOV M[R4+userinput], R1                ; mover o input do
utilizador (previamente guardado em R1) para o userinput
                MOV M[IO_WRITE], R1
                INC R4                                ; o próximo
input será escrito na posição de memória contígua
                INC R6                                ; o próximo
input será escrito na posição de ecrã contígua
                MOV M[IO_CURSOR], R6
                CMP R4, M[numbcores]                  ; v
er se o utilizador já escreveu todas as cores
                BR.Z FIM_INPUT                        ; se já for
am introduzidas todas as cores prossegue-se para o fim
                JMP NEW_INPUT                        ; se a cor introduzida
foi válida pede-se uma nova cor

FIM_INPUT:      POP R6
                POP R4
                POP R3
                POP R2
                POP R1
                RET

;=====
;Validação das cores ( guarda no vetor validacao_output as validações)
;=====
Validacao: PUSH R1
                PUSH R2
                PUSH R3
                PUSH R4
                PUSH R5
                MOV R1, 0000h
                MOV R2, 0000h
                MOV R4, 0000h
                MOV R5, 0000h

Valid_P:  MOV R3, M[R1+randNumbcopia]
        CMP M[R1+userinput], R3

```

jun 02, 17 22:22 C:\Users\User\Desktop\p3print_win\mastermind.as Page 12/19

```

        BR.Z GuardaP
Continuacao: INC R1
                CMP R1, M[numbcores]                ; ver se já se comparou pa
ra todos os randNumbcopia cópias se estão na posição correta
                BR.NZ Valid_P
                MOV R1, 0000h                        ; reset ao valor de contagem do r
andNumbcopia

                CMP R4, M[numbcores]
                BR Valid_B

GuardaP: MOV R5, 'p'
        MOV M[R4+validacao_output], R5
        MOV R5, 'q'
        MOV M[R1+randNumbcopia], R5
        MOV R5, 0000h
        MOV M[R1+userinput], R5
        INC R4
        CMP R4, M[numbcores]
        JMP.Z FimValidacaoganhhou
        BR Continuacao

Valid_B: MOV R3, M[R1+randNumbcopia]
        CMP M[R2+userinput], R3
        BR.Z GuardaB
        INC R1
        CMP R1, 0004h
        BR.NZ Valid_B
        MOV R1, 0000h
        INC R2
        CMP R2, M[numbcores]
        BR.Z FimValidacao
        BR Valid_B

GuardaB: MOV R5, 'b'
        MOV M[R4+validacao_output], R5
        INC R4
        MOV R5, 'q'                                ; cor inválida
        MOV M[R1+randNumbcopia], R5
        INC R2
        MOV R1, 0000h                                ; reiniciar a contagem para o próximo userin
put

        CMP R2, M[numbcores]
        BR.Z FimValidacao
        BR Valid_B

FimValidacaoganhhou: MOV R1, 0001h
                    MOV M[fimjogo], R1

FimValidacao: POP R5
                POP R4
                POP R3
                POP R2
                POP R1
                RET

;=====
;Criar cópia do randnumb
;=====
CopiaRandnumb: PUSH R1
                PUSH R2
                MOV R2, 0000h

```

jun 02, 17 22:22 C:\Users\User\Desktop\p3print_win\mastermind.as Page 13/19

```

CicloCopia:      MOV R1, M[R2+randNumb]      ; gerar uma cópia de randnumb
em randnumbcopia, dado que randnumb copia vai ser alterado
                MOV M[R2+randNumbcopia], R1
                INC R2
                CMP R2, M[numbcores]
                BR.NZ CicloCopia
                POP R2
                POP R1
                RET
;=====
; Reset à Validação
;=====
ResetValidacao: PUSH R2
                PUSH R1
                MOV R1, 0000h
                MOV R2, ' '      ; fazer o reset ao vetor vali
dacao_output
                MOV M[R1+validacao_output], R2
                INC R1
                MOV M[R1+validacao_output], R2
                INC R1
                MOV M[R1+validacao_output], R2
                INC R1
                MOV M[R1+validacao_output], R2
                INC R1
                MOV M[R1+validacao_output], R2
                INC R1
                MOV M[R1+validacao_output], R2
                INC R1
                MOV M[R1+validacao_output], R2
                POP R2
                POP R1
                RET
;=====
; Escrita no LCD
;=====
EscLCD:         PUSH R1
                PUSH R2
                PUSH R3
                PUSH R4
                PUSH R5
                MOV R4, 0001h
                MOV R5, 0000h
                CALL LimpaLCD
;escrever o número de tentativas
                MOV R1, 8000h
                PUSH VarTexto24
                PUSH R1
                CALL EscStringLCD
                MOV R1, 8007h
                OR R1, 0007h      ; escrever 8 colunas a seguir
                MOV M[LCD_CURSOR], R1
                MOV R3, M[R4+jogador1]      ; R3 fica com o número de tentati
vas do jogador1
                ADD M[totaltentativas], R3      ; guarda o total d
e tentativas, qu será usado para calcular a média
                OR R3, 0030h      ; converter para ASCII
                MOV M[LCD_WRITE], R3
                ADD R1, 0002h      ; escrever 2 colunas a seguir ->
jogador2
                MOV M[LCD_CURSOR], R1

```

jun 02, 17 22:22 C:\Users\User\Desktop\p3print_win\mastermind.as Page 14/19

```

                MOV R3, M[R4+jogador2]
                ADD M[totaltentativas], R3      ; guarda o total d
e tentativas, qu será usado para calcular a média
                OR R3, 0030h
                MOV M[LCD_WRITE], R3
                ADD R1, 0002h      ; escrever 2 colunas a seguir->j
ogador3
                MOV M[LCD_CURSOR], R1
                MOV R3, M[R4+jogador3]
                ADD M[totaltentativas], R3      ; guarda o total
de tentativas, qu será usado para calcular a média
                OR R3, 0030h
                MOV M[LCD_WRITE], R3
                ADD R1, 0002h      ; escrever 2 colunas a seguir->me
dia tentativas
                MOV M[LCD_CURSOR], R1
;escrever o número médio de tentativas
                MOV R5, M[totaljogadores]
                MOV R2, M[totaltentativas]
                DIV R2, R5
                OR R2, 0030h
                MOV M[LCD_WRITE], R2      ; escrever a média
                ADD R1, 0001h
                MOV M[LCD_CURSOR], R1
                MOV R3, '.'
                MOV M[LCD_WRITE], R3
                ADD R1, 0001h
                MOV M[LCD_CURSOR], R1
                OR R5, 0030h
                MOV M[LCD_WRITE], R5      ; escrever as décimas de segundo da mé
dia
;escrever os tempos de jogo
                MOV R1, 8010h
                PUSH VarTexto25
                PUSH R1
                CALL EscStringLCD      ; a partir daqui escrevem-se os
tempos de jogo
                OR R1, 0008h      ; escrever 8 colunas a seguir
                MOV M[LCD_CURSOR], R1
                MOV R3, M[jogador1]
                OR R3, 0030h      ; converter para ASCII
                MOV M[LCD_WRITE], R2
                ADD R1, 0002h      ; escrever 2 colunas a seguir ->
jogador2
                MOV M[LCD_CURSOR], R1
                MOV R3, M[jogador2]
                OR R3, 0030h
                MOV M[LCD_WRITE], R2
                ADD R1, 0002h      ; escrever 2 colunas a seguir->j
ogador3
                MOV M[LCD_CURSOR], R1
                MOV R3, M[jogador3]
                OR R3, 0030h
                MOV M[LCD_WRITE], R2
                POP R5
                POP R4
                POP R3
                POP R2
                POP R1

```

jun 02, 17 22:22 C:\Users\User\Desktop\p3print_win\mastermind.as Page 15/19

```

RET
;=====
; Escrita da Validação
;=====
EscreveValidacao: ADD R5, 0100h          ; escrever a validação na linha se
guinte para cada uma das tentativas

PUSH validacao_output
PUSH R5
CALL EscString

FIM_escrevalidacao: RET

;=====
;=====
; Jogo
;=====
JOGO: MOV R7, 000Ah          ; utilizado para comparar co
m o número de tentativas M[tentativa]
MOV R6, 031Bh              ; utilizado para definir o valor ini
cial da linha e coluna para o input do utilizador
MOV R5, 0336h              ; utilizado para definir o valor ini
cial da linha e coluna para a validação
CALL LimpaDisplays
CALL LimpaLCD
CALL INIT_INT
CALL InitSeed
ENI
CALL LimpaJanela

CALL MenuInicial
MOV M[TEMP_STATE], R0
CALL LimpaJanela
CALL MenuOpcoes
CALL INIT_TIME
CALL LimpaJanela

CALL GenerateRand
CALL WRITE_BOARD
CALL EscJogador1

JOGA1: CALL User_Input
CALL CopiaRandnumb
CALL Validacao
CALL EscreveValidacao
CALL ResetValidacao
CMP M[fimjogo], R0

BR.NZ FIMJOGA1
INC M[tentativa]
CMP M[tentativa], R7
BR.NZ JOGA1
FIMJOGA1: MOV M[TEMP_STATE], R0
PUSH jogador1
CALL GuardaStats
MOV R1, 0001h
CMP M[numbjogadores], R1
JMP.Z FimJogo          ; ir para o fim d

o jogo

CALL PressEnter
MOV M[fimjogo], R0
CALL LimpaJanela

CALL GenerateRand
CALL WRITE_BOARD
CALL EscJogador2
MOV R7, 000Ah          ; utilizado para com
parar com o número de tentativas M[tentativa]

```

jun 02, 17 22:22 C:\Users\User\Desktop\p3print_win\mastermind.as Page 16/19

```

MOV R6, 031Bh          ; utilizado para definir o valor ini
cial da linha e coluna para o input do utilizador
MOV R5, 0336h          ; utilizado para definir o valor ini
cial da linha e coluna para a validação
MOV M[tentativa], R0
INC M[jogador_playing]
CALL INIT_INT
CALL INIT_TIME
ENI
JOGA2: CALL User_Input
CALL CopiaRandnumb
CALL Validacao
CALL EscreveValidacao
CALL ResetValidacao
CMP M[fimjogo], R0

BR.NZ FIMJOGA2
INC M[tentativa]
CMP M[tentativa], R7
BR.NZ JOGA2
FIMJOGA2: MOV M[TEMP_STATE], R0
PUSH jogador2
CALL GuardaStats
MOV R1, 0002h
CMP M[numbjogadores], R1
JMP.Z FimJogo          ; ir para o fim do jo
go

CALL PressEnter
MOV M[fimjogo], R0          ; fazer reset ao resultado do jogo
CALL LimpaJanela

CALL GenerateRand
CALL WRITE_BOARD
CALL EscJogador3
MOV R7, 000Ah          ; utilizado para com
parar com o número de tentativas M[tentativa]
MOV R6, 031Bh          ; utilizado para definir o valor ini
cial da linha e coluna para o input do utilizador
MOV R5, 0336h          ; utilizado para definir o valor ini
cial da linha e coluna para a validação
MOV M[tentativa], R0
CALL INIT_INT
CALL INIT_TIME
ENI
INC M[jogador_playing]
JOGA3: CALL User_Input
CALL CopiaRandnumb
CALL Validacao
CALL EscreveValidacao
CALL ResetValidacao
CMP M[fimjogo], R0

BR.NZ FIMJOGA3
INC M[tentativa]
CMP M[tentativa], R7
BR.NZ JOGA3
FIMJOGA3: PUSH jogador3
CALL GuardaStats
FimJogo: MOV M[TEMP_STATE], R0
CALL PressEnter
CALL LimpaJanela

CALL MenuFinal
CALL EscLCD
JMP Fim
;=====

```


jun 02, 17 22:22 C:\Users\User\Desktop\p3print_win\mastermind.as Page 17/19

```
; Função que devolve as estatísticas para cada jogador
;=====
GuardaStats:PUSH R1
                PUSH R2
                MOV R1, M[SP+4]      ;endereço do jogador
                MOV R2, M[tempo]     ;tempo de jogo
                MOV M[R1],R2         ; guardar o tempo de jogo
                INC R1
                MOV R2, M[tentativa] ; guardar o número de tentativas
                MOV M[R1], R2
                POP R2
                POP R1
                RETN 1
;=====
; LimpaJanela: Rotina que limpa a janela de texto.
; Entradas: --
; Sidas: ---
; Efeitos: ---
;=====
LimpaJanela:    PUSH R2
                MOV R2, READ
                MOV M[IO_CURSOR], R2
                POP R2
                RET
;=====
; LimpaLCD: Rotina que limpa o LCD
;=====
LimpaLCD:       PUSH R1
                MOV R1, 0020h
                MOV M[LCD_CURSOR], R1
                POP R1
                RET
;=====
; LimpaDisplays : Rotina que limpa os display de 7 sementos
;=====
LimpaDisplays:PUSH R1
                MOV R1, 0000h
                MOV M[DISP_1], R1
                MOV M[DISP_2], R1
                MOV M[DISP_3], R1
                MOV M[DISP_4], R1
                POP R1
                RET
;=====
; EscCarLCD: Rotina que efectua a escrita de um caracter para o LCD.
; O caracter pode ser visualizado na janela de texto.
; Entradas: R1 - Caracter a escrever
; Sidas: ---
; Efeitos: alteracao da posicao de memoria M[IO]
;=====
EscCarLCD:      MOV M[LCD_WRITE], R1
                RET
;=====
; EscStringLCD: Rotina que efectua a escrita de uma cadeia de caracter, terminad
a
; pelo caracter FIM_TEXTO, no LCD numa posicao
; especificada. Pode-se definir como terminador qualquer caracter
;=====
```

jun 02, 17 22:22 C:\Users\User\Desktop\p3print_win\mastermind.as Page 18/19

```
; ASCII.
; Entradas: pilha - posicao para escrita do primeiro carater
; pilha - apontador para o inicio da "string"
; Sidas: ---
; Efeitos: ---
;=====
EscStringLCD:   PUSH R1
                PUSH R2
                PUSH R3
                MOV R2, M[SP+6]      ; Apontador para inicio da "string"
                MOV R3, M[SP+5]      ; Localizacao do primeiro carater
                MOV M[LCD_CURSOR], R3
                MOV R1, M[R2]
                CMP R1, FIM_TEXTO
                BR.Z FimEscLCD
                CALL EscCarLCD
                INC R2
                INC R3
                BR CicloLCD
FimEscLCD:      POP R3
                POP R2
                POP R1
                RETN 2                ; Actualiza STACK
;=====
; EscCar: Rotina que efectua a escrita de um caracter para o ecran.
; O caracter pode ser visualizado na janela de texto.
; Entradas: R1 - Caracter a escrever
; Sidas: ---
; Efeitos: alteracao da posicao de memoria M[IO]
;=====
EscCar:         MOV M[IO_WRITE], R1
                RET
;=====
; EscString: Rotina que efectua a escrita de uma cadeia de caracter, terminada
; pelo caracter FIM_TEXTO, na janela de texto numa posicao
; especificada. Pode-se definir como terminador qualquer caracter
; ASCII.
; Entradas: pilha - posicao para escrita do primeiro carater
; pilha - apontador para o inicio da "string"
; Sidas: ---
; Efeitos: ---
;=====
EscString:      PUSH R1
                PUSH R2
                PUSH R3
                MOV R2, M[SP+6]      ; Apontador para inicio da "string"
                MOV R3, M[SP+5]      ; Localizacao do primeiro carater
                MOV M[IO_CURSOR], R3
                MOV R1, M[R2]
                CMP R1, FIM_TEXTO
                BR.Z FimEsc
                CALL EscCar
                INC R2
                INC R3
                BR Ciclo
Ciclo:          MOV R3, M[SP+5]
                MOV R1, M[R2]
                CMP R1, FIM_TEXTO
                BR.Z FimEsc
                CALL EscCar
                INC R2
                INC R3
                BR Ciclo
FimEsc:         POP R3
                POP R2
                POP R1
                RETN 2                ; Actualiza STACK
;=====
```

```
;  
;----- Programa principal -----  
;-----  
inicio:      MOV      R1, SP_INICIAL  
              MOV      SP, R1  
              CALL JOGO  
  
Fim:         BR       Fim
```