nov 30, 17 15:19		Mast	terMind.as		Page 1/16
;MasterMind					
;Andre Afonso Nunes ;Rafael Nunes Henriq					
;Projeto desenvolvid tadores	o na Unida	ide Currici	ılar de Introd	ucao a Arquitet	ura de Compu
;		CONSTAN			
;					
SP INICIAL	EQU	FDFFh			
MASCARA_RANDOM andom	EQU		000010110b	;mascara par	ra a funcao r
STR_END	EQU	' @ '		;fim das str	
clrscreen lrscr	EQU	, ,		;espaco em b	oranco para c
INT_MASK_ADDR IO_DISPLAY_1	EQU EQU	FFFAh FFF0h			
IO_DISPLAY_2	EQU	FFF1h			
IO_DISPLAY_3	EQU	FFF2h			
IO_DISPLAY_4	EQU	FFF3h			
LCD_CTRL	EQU	FFF4h			
LCD_DISPLAY	EQU	FFF5h FFF6h			
TEMP_COUNT TEMP_CTRL	EQU EQU	FFF7h			
LEDS	EQU	FFF8h			
IO_CURSOR	EQU	FFFCh			
IO_WRITE	EQU	FFFEh			
;;		VARIAVE			
;					
	ORIG	8000h			
ni_random	WORD	0			
jogada	WORD	0	;guarda jogad	a	
codigo	WORD	0	; guarda codig	0	
resultado	WORD		; guarda compa		
n_jogadas	WORD		;pontuacao at	ual	
highscore limite	WORD WORD	15 12	;highscore ;limite jogad	as num iodo	
cursor	WORD	0000h	;posicao curs	as num jogo or	
ganhou			;flag	01	
;str	ings				
menu_inicial	STR	'Carregu	ne no botao IA	para iniciar',	STR_END
highscore_string menu_final	STR STR	'HIGHSCO	DRE:', STR_END	ue em IA para r	
em IB para sair', ST				-	
creditos	STR		s:', STR_END		
Andre	STR			ilva - 89408',	
Rafael	STR			es - 89530',	STR_END
perdeu_str ganhou_str	STR STR		, STR_END , STR_END		
;end	_interrupo	coes			

nov 30, 17 15:19		MasterMind.as	Page
	ORIG	FE01h	
INT1 INT2 INT3 INT4 INT5 INT6	WORD WORD WORD WORD WORD WORD	INT1F INT2F INT3F INT4F INT5F INT6F	
	ORIG	FEOAh	
INT10 INT11	WORD WORD	INT10F INT11F	
	ORIG	FEOFh	
INT15	WORD	INT15F	
, , , , , , , , , , , , , , , , , , ,		CODIGO	
	ORIG	0000h	
	MOV MOV MOV	R7, SP_INICIAL SP, R7 R7, FFFFh M[IO_CURSOR], R7	;inicializa ;inicializa
or IO	JMP	INICIO_JOGO	
;; ;		INTERRUPCOES	
;int_jogad	da		
<i>;</i>	esponden	te a interrupcao no registos	
INT1F:	SHL ADD RTI	R1, 3 R1, 1	
INT2F:	SHL ADD RTI	R1, 3 R1, 2	
INT3F:	SHL ADD RTI	R1, 3 R1, 3	
INT4F:	SHL ADD RTI	R1, 3 R1, 4	
INT5F:	SHL ADD RTI	R1, 3 R1, 5	

nov 30, 17 15:19		MasterMind.as	Page 3/16
INT6F:	SHL ADD RTI	R1, 3 R1, 6	
INT10F:	INC RTI	R6	;flag
INT 11F:	DEC RTI	R6	;flag
INT15F: gado ao fim nao precisa	BR.Z MOV MOV MOV	salta R7, 5 M[TEMP_COUNT], R7 R7, 1	;flag ;se ja tiver che
ntador para proximos 50 salta:	RTI	M[TEMP_CTRL], R7	;reinicializa co
;; ; ;		FUNCOES	
<pre>;clrscr_ ; ;Funcao que limpa ecra, ; as posicoes ; ;Input: - Output: -</pre>		nindo o caracter '' para todas	
clrscr: ;cursor	PUSH PUSH	R1 R2	
;caracter ' '	PUSH	R3	
;contador ;auxiliar	PUSH MOV MOV MOV	R4 R1, 0000h R2, clrscreen R3, 1896	
ciclo_clrscr:	CMP BR.Z	R3, R0 fim_clrscr	;imprime
em todas as posicoes	DEC MOV MOV INC MOV AND CMP	R3 M[IO_CURSOR], R1 M[IO_WRITE], R2 R1 R4, R1 R4, 00FFh R4, 78	; chegou
ao fim da linha?	BR.Z BR	next_line_clrscr ciclo_clrscr	. 5
next_line_clrscr:	MOV	R4, R1	;mete co
guinta novembro 30, 2017			Maste

nov 30, 17 15:19		MasterMind.as	Page
lunas a 0 e incrementa		- 4	
	AND ADD	R4, FF00h	
	MOV	R4, 0100h R1, R4	
	BR	ciclo_clrscr	
fim_clrscr:	POP	R4	
	POP	R3	
	POP	R2	
	POP RET	R1	
;print_s			
;			
;Funcao que escreve uma ; posicao que rec			
; ;Input: string, posicao		-	
<pre>print_str: ;caracter a imprimir</pre>	PUSH	R1	
cursor string;	PUSH	R2	
-	PUSH	R3	
;cursor IO	PUSH	R4	
;auxiliar			
	MOV	R2, M[SP+7]	
	MOV	R3, M[SP+6]	
ciclo_prtstr:	MOV	R1, M[R2]	
hegou ao fim da str	CMP	R1, STR_END	; ve
	BR.Z	fim_prtstr	
	MOV	M[IO_CURSOR], R3	
	MOV INC	M[IO_WRITE], R1 R2	
	INC	R3	
	MOV	R4, R3	
	AND	R4, 00FFh	,
ao fim da linha?	CMP	R4, 78	; ch
	BR.Z	next_line_prtscr	
	BR	ciclo_prtstr	
<pre>next_line_prtscr: a zero, linhas + 1</pre>	MOV	R4, R3	; co.
	AND	R4, FF00h	
	ADD	R4, 0100h	
	MOV	R3, R4	
	BR	ciclo_prtstr	
fim_prtstr:	POP	R4	
	POP	R3	
	POP POP	R2 R1	
	RETN	2	
:print_t	exto_LCD		
; ;Funcao que inicializa	o LCD e	imprime "Highscore:"	

nov 30, 17 15:19		MasterMind.as	Page 5/16
; ;Input: - Output: -			
print_texto_LCD:	PUSH	R1	
;cursor LCD	PUSH	R2	
;cursor da string	PUSH	R3	
;auxiliar string	MOV MOV	R1, 8000h R2, highscore_string	
ciclo_texto_LCD:	MOV CMP BR.Z MOV MOV INC INC BR	R3, M[R2] R3, STR_END fim_texto_LCD M[LCD_CTRL], R1 M[LCD_DISPLAY], R3 R2 R1 ciclo_texto_LCD	
<pre>fim_texto_LCD: dos algarismos</pre>	MOV MOV MOV INC MOV	R1, 800Ch R3, '-' M[LCD_CTRL], R1 M[LCD_DISPLAY], R3 R1 M[LCD_CTRL], R1	;posicao
	MOV POP POP POP	M[LCD_DISPLAY], R3 R3 R2 R1	
	RET		
;atual; ; ;Funcao que atualiza o		mpresso no LCD	
; ;Input: inteiro Outpu	t: -		
atualiza_LCD:	PUSH	R1	;valor a imprimi
r	PUSH PUSH MOV MOV	R2 R3 R1, M[SP+5] R3, R0	;dezenas ;auxiliar
	CMP BR.N	R1, 10 print_n_LCD	;se for menor qu
e 10 imprime	INC SUB	R2 R1,10	; dezenas
print_n_LCD:	ADD	R2, 48	;converte em ASC
II	ADD	R1, 48	
	MOV MOV MOV	R3, 800Ch M[LCD_CTRL], R3 M[LCD_DISPLAY],R2	;imprime dezenas
quinta novembro 30, 2017			MasterM

nov 30, 17 15:19		MasterMind.as	Page
	INC MOV MOV	R3 M[LCD_CTRL], R3 M[LCD_DISPLAY],R1	;impime uni
	POP POP	R3 R2 R1	
	RETN	1	
;input_5	iogada		
; ;Funcao que espera pela ; devolve o input ;	(jogada	do jogador e 1)	
;Input: - Output: jogac			
<pre>input_jogada: ; jogada</pre>	PUSH	R1	
;auxiliar	PUSH	R2	
;flag do temporizador	PUSH	R3	
;estado dos leds	PUSH	R6	
;auxiliar	PUSH	R7	
, danii idi	MOV MOV MOV	R7, 807Eh M[INT_MASK_ADDR], R7 R6, FFFFh M[LEDS], R6	;mascara interrutoro;in.
iza leds	MOV	R1, R0	;in.
iza jogada	MOV	R3, R0	;f1
0	ENI MOV MOV MOV	R7, 5 M[TEMP_COUNT], R7 R7, 1 M[TEMP_CTRL], R7	;inicializa
ador		11(1211 _0110)/ 10	,1110141124
ciclo_input:	MOV AND CMP	R2, R1 R2, 0000111000000000b R2, R0	; ve.
a se ja foram	JMP.NZ	controlo_input	;introduzid
algarismos	CMP	R3, 1	
ou 500ms?	BR.N	ciclo_input	; ja
	CMP JMP.Z	R6, R0 controlo_input	; ja acabou (
po?	SHR MOV MOV BR	R6, 1 M[LEDS], R6 R3, R0 ciclo_input	, , ,
controlo_input:	CMP	R2, R0	

nov 30, 17 15:19		MasterMind.as	Page 7/16
ntroduzidos A algaris	BR.NZ	fim_input	;foram i
ntroduzidos 4 algarism	MOV	R1, R0	;se nao
mete a 0			
fim_input:	DSI MOV	M[SP+7], R1	;guarda
jogada introduzida			_
s LEDS	MOV	M[LEDS], RO	;apaga o
	POP	R7	
	POP	R6	
	POP POP	R3 R2	
	POP	R1	
	RET		
;random	1		-
; ;Funcao que gera um co	odigo pse	udo-aleatorio	
; ;Input: ni_inicial Out	put: ni_	final , codigo	
random:	PUSH	R4	;guarda os registos util
izados	PUSH	R7	
	PUSH	R6	
	MOV	R6, 4	;contador
random_loop:	MOV	R4, M[SP+5]	;ni inicial
	AND	R4, 0001h	;if(n0 == 0)
	CMP JMP.Z	R4, R0 random_zero	
	JMP	random_else	
random_zero:	ROR	M[SP+5], 1	;rotate_right(ni)
	JMP	random_divisao	
<pre>random_else: cara))</pre>	MOV	R4, M[SP+5]	;rotate_right(xor(ni,mas
,	XOR	R4, MASCARA_RANDOM	
	ROR MOV	R4, 1 M[SP+5], R4	
	JMP	random_divisao	
random_divisao:	MOV	R4, M[SP+5]	;divisao de ni por M(6)
para retirar o resto(a	algarismo MOV) R7, 6	
	DIV	R4, R7	
6) da um valor entro	INC	R7 go e preciso incrementa	;resto da divisao por M(
o, aa am valor encre t	ADD	M[SP+6], R7	T POT T
	DEC	R6	;ja fez os 4?
	CMP	R6, R0	-
	JMP.Z	random_fim	
	SHL	M[SP+6], 3	;continua para o proximo
uinta novembro 30, 2017			Maste

JMP POP POP POP RET	random_loop R6 R7	;reposicao dos reg.
POP POP		;reposicao dos reg
POP	R7	
RET	R4	
seg		_
o display	de 7 segmentos (n jogad	das atual)
t: -		
PUSH	R1	
		• **
		; V
	·	; d
	·	; S
JMP . NN	maior_10	;
JMP	print_sete_seg	
INC SUB	R2 R1. 10	
JMP	ciclo_sete_seg	
MOV	M[IO_DISPLAY_4], RO	;limpa ·limpa
MOV	M[IO_DISPLAY_2], R2	;limpa ;print dez
		;print uni
POP POP	R2 R1	
RETN	1	
ica		
		certos e os existente
o Output:	resultado	
PUSH	R1	
PUSH	R2	
PUSH	R3	
MOV MOV	R1, M[SP+6] R2, M[SP+5]	
MOV	R3, R0	
PUSH	R0	
	PUSH PUSH MOV MOV CMP JMP.NN JMP INC SUB JMP MOV MOV MOV MOV MOV MOV POP POP RETN ica jogada ccoes auxili o Output: PUSH PUSH PUSH MOV	PUSH R1 PUSH R2 MOV R1, M[SP+4] MOV R2, R0 CMP R1, 10 JMP.NN maior_10 JMP print_sete_seg INC R2 SUB R1, 10 JMP ciclo_sete_seg MOV M[IO_DISPLAY_4], R0 MOV M[IO_DISPLAY_3], R0 MOV M[IO_DISPLAY_1], R1 POP R2 POP R1 RETN 1 ica jogada com o codigo oes auxiliares para verificar os o Output: resultado PUSH R1 PUSH R2 PUSH R3 MOV R1, M[SP+6] MOV R2, M[SP+5] MOV R3, R0 PUSH R0

nov 30, 17 15:19		MasterMind.as	Page 9/16
	PUSH PUSH PUSH PUSH CALL POP	R0 R1 R2 R3 ver_x R3	
<pre>;resultado ;jogada sem os certos</pre>	POP	R2	
;codigo sem os certos	POP	R1	
	PUSH PUSH PUSH PUSH CALL POP MOV	R0 R1 R2 R3 ver_o R3 M[SP+7], R3	
	POP POP	R3 R2 R1	
	RETN	2	
ver_x: ;CODIGO	PUSH	R1	
; JOGADA	PUSH	R2	
;GUARDA RESULTADO	PUSH	R3	
;DIGITO 1	PUSH	R4	
;DIGITO 2	PUSH	R5	
; CONTADOR	PUSH	R6	
	MOV MOV MOV	R1, M[SP+10] R2, M[SP+9] R3, M[SP+8] R6, R0	
loop_ver_x:	CMP	R6, 4	DODE 4
VEZES	BR.Z	fim_ver_x	;RODA 4
LGARISMOS DIREITA	MOV AND	R4, R1 R4, 0007H	;ISOLA A
HOARISMUS DIREITA	MOV AND	R5, R2 R5, 0007H	
	CMP BR.NZ ROL INC	R5, R4 continua_x R3, 2 R3	
LAGARIMOS	AND	R1, FFF8H	;APAGA A
lgarismos	AND	R2, FFF8h	;apaga a

nov 30, 17 15:19		MasterMind.as	Page 1
continua_x:	ROR	R1, 3	;pr
s 2 digitos	ROR	R2, 3	
	INC BR	R6	
		loop_ver_x	
fim_ver_x: codigo e jogada na pos	ROR icao cei	R1, 4 rta	; co.
	ROR	R2, 4	
	MOV	M[SP+11], R3	
	MOV MOV	M[SP+12], R2 M[SP+13], R1	;mete resul
na stack			,
	POP	R6	
	POP POP	R5 R4	
	POP	R3	
	POP POP	R2 R1	
	RETN	3	
ver_o:	PUSH	R1	
;codigo			
; jogada	PUSH	R2	
;resultado	PUSH	R3	
;digito 1	PUSH	R4	
	PUSH	R5	
;digito 2	PUSH	R6	
;contador			
	MOV	R1, M[SP+10]	
	MOV MOV	R2, M[SP+9] R3, M[SP+8]	
ciclo_codigo:	CMP	R1, R0	
e ja encontrou 4 corre	BR.Z	fim_ver_o	; sa.
c ja cheometou i cotte			
	MOV AND	R4, R1 R4, 0007h	;is
algarismo da direita	CMP	R4, R0	<i>;s</i> e
ver a 0 poupa verifica		fim_ciclo_cod	,50
r	MOV	R6, R0	; coi
ciclo_jogada:	CMP	R6, 4	; ja
os 4 algarismos		•	, , , , ,
	BR.Z	fim_ciclo_jog	
	MOV AND	R5, R2 R5, 0007h	
		- / * * * : : : :	

nov 30, 17 15:19		MasterMind.as	Page 11/16
	СМР	R5, R0	;se esti
ver a 0 passa para prox	BR.Z	proximo_digito_jog	
	CMP BR.NZ ROL ADD	R5, R4 proximo_digito_jog R3, 2 R3, 2	;encontr
ou correspondencia	AND	R1, FFF8h	;apaga d
igitos direita	AND	R2, FFF8h	
proximo_digito_jog:	ROR	R2, 3	;proximo
digito jogada	INC	R6	
;contador aumenta	BR	ciclo_jogada	
fim_ciclo_jog: iona os bits da jogada	ROR	R2, 4	;reposic
fim_ciclo_cod: digito codigo	SHR	R1, 3	;proximo
argree coarge	BR	ciclo_codigo	
<pre>fim_ver_o: resultado</pre>	MOV	M[SP+11], R3	;guarda
	POP POP POP POP POP	R6 R5 R4 R3 R2 R1	
	RETN	3	
;print_r	esultad	0	
; ;Funcao que imprime jog ; ;Input: cursor, resulta		resultado da comparacao	
<pre>print_resultado:</pre>	PUSH	R1	
;cursor	PUSH	R2	
;resultado	PUSH	R3	
; jogada	PUSH	R4	
;contador	PUSH	R5	
;auxiliar			
	MOV ADD	R1, M[SP+9] R1, 0100h	;imprime
na linha seguinte	AND	R1, FF00h	; na pri
meira coluna	MOV MOV	R2, M[SP+8] R3, M[SP+7]	
quinta novembre 20, 2017			Mastarh

nov 30, 17 15:19		MasterMind.as	Page 1
	MOV	R4, R0	
jogada em posicao para .	ROL imprimir	R3, 4	; co.
ciclo_prt_jog: imiu os 4?	CMP	R4, 4	; ja
II. 00 1.	BR.Z	prt_xo	
digito	ROL	R3, 3	;pr
-	MOV AND ADD MOV	R5, R3 R5, 0007h R5, 48 M[IO_CURSOR], R1	;imprime al
mo	MOV INC	M[IO_WRITE], R5 R1	
;proxima coluna	INC	R4	
;aumenta contador		11.1	
	BR	ciclo_prt_jog	
prt_xo:	MOV MOV MOV INC MOV MOV	R5, clrscreen M[IO_CURSOR], R1 M[IO_WRITE], R5 R1 M[IO_CURSOR], R1 M[IO_WRITE], R5	;dois espac
branco	MOV	M[IO_WRITE], RS	, dois espaci
resultado para imprimi.	MOV ROL	R4, R0 R2, 8	;pre
ciclo_posiciona: u 4 vezes ja esta posic	CMP ionado BR.Z MOV	R4, 4 salta_posiciona R5, R2	;se
s dois ultimos bits	AND	R5, C000h	;is
ao ja esta posicionado	CMP BR.NZ	R5, R0 salta_posiciona	;se n e zero
	ROL INC BR	R2, 2 R4 ciclo_posiciona	
salta_posiciona:	MOV	R4, R0	
ciclo_prt_xo: imiu os 4?	CMP	R4, 4	<i>;</i> ja
	JMP . Z	fim_prt_res	
s 2 bits	ROL MOV	R2, 2	;pr
	AND	R5, R2 R5, 0003h	;is
s dois bits	CMP BR.Z	R5, 0001h print_x	;se
zero, algarismo e 1, imp			, 50

nov 30, 17 15:19		MasterMind.as	Page 13/16
	BR.P	print_o	;se for
positivo, algarismo e l	BR	print_traco	;se for negativo
, algarismo e 0, imprim	me '-'		
print_x:	MOV	R5, 'x'	;imprime
	MOV MOV INC INC BR	M[IO_CURSOR], R1 M[IO_WRITE], R5 R1 R4 ciclo_prt_xo	
print_o: e 'o'	MOV	R5, 'o'	;imrprim
	MOV MOV INC INC BR	M[IO_CURSOR], R1 M[IO_WRITE], R5 R1 R4 ciclo_prt_xo	
<pre>print_traco: ;imprime '-'</pre>		MOV R5, '-'	
,	MOV MOV INC INC JMP	M[IO_CURSOR], R1 M[IO_WRITE], R5 R1 R4 ciclo_prt_xo	
fim_prt_res:	MOV	M[SP+10], R1	;guarda cursor
	POP POP POP POP	R5 R4 R3 R2 R1	
	RETN	3	
;; ;		CORPO PRINCIPAL	
INICIO_JOGO:	CALL	clrscr	;limpa e
	CALL	print_texto_LCD	;inicializa LCD
inicial cursos	PUSH PUSH	menu_inicial 0000h	;posicao
Interar edisos	CALL	print_str	
aiole Th	MOV MOV ENI MOV	R7, 0400h M[INT_MASK_ADDR], R7 R6, R0	.inon-m-rt-
ciclo_IA: enquanto espera pela .	INC interrup CMP	M[ni_random] ocao R6, 1	;incrementa o ni
quinto novembro 20, 2017			Mostor

nov 30, 17 15:19		MasterMind.as	Page 1
	BR.NZ DSI	ciclo_IA	
JOGO_NOVO:	PUSH PUSH CALL	R0 M[ni_random] random	; ge
digo	POP POP	M[ni_random] M[codigo]	, , ,
	CALL	clrscr	
	MOV PUSH CALL	<pre>M[n_jogadas], R0 M[n_jogadas] sete_seg</pre>	;ir
iza display 7 segment	os		
INICIO_JOGADA:	PUSH CALL POP	RO input_jogada M[jogada]	;re
jogada	CMP	M[jogada], R0	;verifica s
introduzida	JMP . Z	novo_jogo	; se
0 o tempo acabou acab		_3 3	,
	PUSH PUSH PUSH CALL	R0 M[codigo] M[jogada] verifica	;00
jogada e codigo	POP	M[resultado]	;recebe res
0		•	,
print_jogada: ;guarda espaco para c	PUSH	R0	
	PUSH PUSH PUSH	M[cursor] M[resultado] M[jogada]	;00
ado em 2 bits cada	CALL	print_resultado	;print joga
do resultado	POP	M[cursor]	
	INC MOV CMP	M[n_jogadas] R7, 0055h M[resultado], R7	;verifica s
hou	BR.Z PUSH CALL	atualiza_highscore M[n_jogadas] sete_seg	;at
a display	MOV CMP BR.Z	R7, M[n_jogadas] R7, 12 novo_jogo	;se
ez 12 jogadas		3 - 3 -	, 55
gada	MOV	M[jogada], R0	;reiniciali
sultado	MOV	M[resultado], R0	;reiniciali

nov 30, 17 15:19		MasterMind.as	Page 15/16
	JMP	INICIO_JOGADA	
atualiza_highscore:	INC MOV CMP BR.NN	M[ganhou] R7, M[n_jogadas] R7, M[highscore] novo_jogo	;se tive
r feito highscore			
ore	MOV	M[highscore], R7	;atualiza highsc
	PUSH CALL	R7 atualiza_LCD	;atualiza LCD
novo_jogo: a para menu final	CALL	clrscr	;continu
u?	CMP	M[ganhou], R0	;ganhou ou perde
<pre>perdeu: salta_ganhou:</pre>	BR.Z MOV MOV BR MOV PUSH PUSH	<pre>perdeu R7, ganhou_str M[ganhou], R0 salta_ganhou R7, perdeu_str R7</pre>	
GANHOU ou PERDEU no I	CALL	print_str	;imprime
GANHOU OU PERDEU NO 1			
1	PUSH PUSH CALL	menu_final 0100h print_str	;print menu fina
	MOV MOV ENI MOV	R7, 0C00h M[INT_ MASK_ADDR], R7 R6, R0	
ciclo_IA_IB: u IA ou IB?	CMP	R6, FFFFh	;escolhe
	BR.Z CMP BR.NZ DSI	fim_creditos R6, 0001h ciclo_IA_IB	
	MOV	M[jogada], R0	;repoe valores p
ara novo jogo	MOV MOV MOV	M[codigo], R0 M[resultado], R0 M[n_jogadas], R0 M[cursor], R0	
	JMP	JOGO_NOVO	
fim_creditos:	DSI CALL	clrscr	
	PUSH PUSH CALL	creditos RO print_str	;imprime
creditos e acabou	-	1	, implime
	PUSH PUSH	Andre 0100h	
quinto novembro 20, 2017			Maata

			1 miled by
nov 30, 17 15:19		MasterMind.as	Page 1
	CALL	print_str	
	PUSH PUSH CALL	Rafael 0200h print_str	
Fim:	BR	Fim	