

Program Name: **Programming Exercise2 - プログラミング演習2**

GR#0				
GR#1				
GR#2				
GR#3				
GR#4				
GR#5				
GR#6				
GR#7				
GR#8				
GR#9				
GR#A				
GR#B				
GR#C				
GR#D				
GR#E				
GR#F				

IA (next) 00

IR

CC

Step Count 0

OK. right answer.

User Comment

(このエリアへはこのシート上でコメントを直接記入する。inputCLRボタンでは消去されない)

印刷方法: 上部メニューのページレイアウトで以下のように設定する。  
 ・A4縦刷印刷の場合 余白: ユーザ設定, 上: 1.4cm, 下: 1.4cm, 左: 1.8cm, 右: 1.3cm, ヘッダ: 0.5cm, フッタ: 0.5cm  
 印刷の向き: 縦, サイズ: A4, 拡大/縮小: 63% (手入力)  
 (シート全体は3ページとなるので必要なページを印刷する)  
 ・A4横刷印刷の場合 余白: 標準, 印刷の向き: 横, サイズ: A4, 拡大/縮小: 82% (手入力)  
 (シート全体は6ページとなるので必要なページを印刷する)  
 注) 印刷前に上部メニューの「表示」, 「改ページプレビュー」で印刷領域を確認すること。  
 シートが左右に分割されている場合は境界の青い破線をマウスで右端へ移動する。

注: Print Bufferのシート名変更は不可(他のシートからProgram Copyボタンでペーストできなくなる)

Label	Op-code RM (Op1)	Op1 Op2 (Op2)	value sign	RR	RM	Op 1	Op 2	Operation	Branch Condition
MM-00									
MM-01									
MM-02									
MM-03	vn	1 1 1 1	v	--	--				
MM-04		2 2 2 2	v	--	--				
MM-05		3 3 3 3	v	--	--				
MM-06		4 4 4 4	v	--	--				
MM-07		5 5 5 5	v	--	--				
MM-08		A A A A	v	--	--				
MM-09	vs	0 0 0 1	v	--	--				
MM-0A		0 0 0 2	v	--	--				
MM-0B		0 0 0 3	v	--	--				
MM-0C		0 0 0 4	v	--	--				
MM-0D		0 0 0 5	v	--	--				
MM-0E	vi	1 2 3 4	v	--	--				
MM-0F									
MM-10									
MM-11									
MM-12		9 A 0 0	--	LA	GR#A	Imv-00	Load Address	--	
MM-13		9 B 0 1	--	LA	GR#B	Imv-01	Load Address	--	
MM-14		9 E 0 A	--	LA	GR#E	Imv-0A	Load Address	--	
MM-15		9 F 0 1	--	LA	GR#F	Imv-01	Load Address	--	
MM-16	g0	1 A A B	AR	--	GR#A	GR#B	Add R	--	
MM-17		1 A B F	AR	--	GR#B	GR#F	Add R	--	
MM-18		1 C B E	CR	--	GR#B	GR#E	Compare R	--	
MM-19		7 C g 0	--	BC	m=C	Lbl-g0	BranchCC=0or1	= or < 0 or	
MM-1A		D A F 0	--	ST	GR#A	MM-F0	Store	--	
MM-1B		0 1 0 0	HLT	--			End Stop	--	
MM-1C									
MM-1D									
MM-1E									
MM-1F		8 0 v i	--	L	GR#0	Lbl-vi	Load	--	
MM-20		1 8 1 0	LR	--	GR#1	GR#0	Load R	--	
MM-21		1 8 2 0	LR	--	GR#2	GR#0	Load R	--	
MM-22		9 C 0 8	--	LA	GR#C	Imv-08	Load Address	--	
MM-23		9 F 0 1	--	LA	GR#F	Imv-01	Load Address	--	
MM-24	g1	2 3 1 1	SD	--	GR#1	GR#1	1-b Shift down	--	
MM-25		1 B C F	SR	--	GR#C	GR#F	Subtract R	--	
MM-26		7 2 g 1	--	BC	m=2	Lbl-g1	Branch CC=2	> +	
MM-27		9 3 F F	--	LA	GR#3	Imv-FF	Load Address	--	
MM-28		2 A 2 3	AND	--	GR#2	GR#3	Logical AND	--	
MM-29		0 1 0 0	HLT	--			End Stop	--	
MM-2A									
MM-2B									
MM-2C		9 A 5 7	--	LA	GR#A	Imv-57	Load Address	--	
MM-2D		9 B 0 0	--	LA	GR#B	Imv-00	Load Address	--	
MM-2E		9 C 0 8	--	LA	GR#C	Imv-08	Load Address	--	
MM-2F		9 E 0 1	--	LA	GR#E	Imv-01	Load Address	--	
MM-30	g3	2 3 A A	SD	--	GR#A	GR#A	1-b Shift down	--	
MM-31		7 8 g 2	--	BC	m=8	Lbl-g2	Branch CC=0	= 0	
MM-32		1 A B E	AR	--	GR#B	GR#E	Add R	--	
MM-33	g2	1 B C E	SR	--	GR#C	GR#E	Subtract R	--	
MM-34		7 2 g 3	--	BC	m=2	Lbl-g3	Branch CC=2	> +	
MM-35		D B F 0	--	ST	GR#B	MM-F0	Store	--	
MM-36		0 1 0 0	HLT	--			End Stop	--	
MM-37									
MM-38									
MM-39									
MM-3A		9 A 0 0	--	LA	GR#A	Imv-00	Load Address	--	
MM-3B		9 B 0 5	--	LA	GR#B	Imv-05	Load Address	--	
MM-3C		1 8 C B	LR	--	GR#C	GR#B	Load R	--	
MM-3D		9 E 0 1	--	LA	GR#E	Imv-01	Load Address	--	
MM-3E	g4	1 A A B	AR	--	GR#A	GR#B	Add R	--	
MM-3F		1 B C E	SR	--	GR#C	GR#E	Subtract R	--	
MM-40		7 2 g 4	--	BC	m=2	Lbl-g4	Branch CC=2	> +	
MM-41		D A F 0	--	ST	GR#A	MM-F0	Store	--	
MM-42		0 1 0 0	HLT	--			End Stop	--	
MM-43									
MM-44									
MM-45									
MM-46									
MM-47									
MM-48									
MM-49									

Comment
(202行以下に理解度確認問題あり)
◎このシートの例題・問題で用いる数値サンプル
H列に ? がある行の命令ワードに空白がある。適切なコードを入力すること。 注1) 命令コードをキーワード検索するときは ? を適切な英小文字に変更しその行の番地セルを選択してからTranslate_barをクリックする 注2) プログラム全体のチェックは番地セル以外を選択しTranslate_barをクリックする 注3) キーワード q による検索, または[ctrl]+qキー入力で穴埋め問題の正解判定可 「例題・問題の空白のコメント欄に各自で命令の意味を書き込むことを推奨する。その際, 文の先頭に*を付けるなどして自分で書いたことが分かるようにしておくこと。」
16.6.3 (PE2) 繰り返し処理 (ループ) とその応用
例題 PE2.1 2進数16進表記で1からAまでを加算し[F0h]にストアする GR#Aを被加数用とし初期値0hを設定 GR#Bを加数用とし初期値1hを設定 GR#Eを加数の上限値用としAhを設定 GR#Fに加算用数値1hを設定 GR#Aの被加数にGR#Bの加数を加算 (ループ処理開始) GR#Bの加数を+1 GR#Bの加数とGR#Eの上限値とを比較 (= or <) ならラベルg0へ分岐 (ループを繰り返す) GR#Aの被加数を[F0h]へストア 終了停止, 答え: [F0h]=0037h, [ctrl]+sで確認
●問題 PE2.1 (1) GR#0の上位バイトと下位バイトをそれぞれGR#1, GR#2の下位バイトに分けて格納する (準備) GR#0 ← [vi] GR#1 ← GR#0 (上位バイト用にコピー) GR#2 ← GR#0 (下位バイト用にコピー) GR#Cにシフト回数 (カウント初期値) を設定 GR#Fに減算用数値1hを設定 GR#1 ← sd GR#1 (下位シフト, ループ処理開始) GR#Cのシフト回数を-1 残りのシフト回数が1以上 (>0の正数) ならラベルg1へ分岐 (ループを繰り返す) GR#3にマスク値を設定 GR#2 and GR#3 (マスク処理) 終了停止, 答え: GR#1=0012h, GR#2=0034h (MM-2A)
●問題 PE2.1 (2) 1バイトデータ57hに含まれる2値の'1'の数を数え[F0h]にストアする GR#A ← 57h GR#Bをゼロクリア GR#Cにシフト回数? (カウント初期値) を設定 GR#Eに加減算用数値1hを設定 GR#A ← sd GR#A (下位シフト, ループ処理開始) CC=0ならラベルg2へ分岐 GR#Bを+1 GR#Cのシフト回数を-1 残りのシフト回数が1以上 (>0の正数) ならラベルg3へ分岐 (ループを繰り返す) [F0h] ← GR#B 終了停止, 答え: [F0]=0005h, [ctrl]+sで確認 (MM-37)
●問題 PE2.1 (3) 数値5hの値を二乗し[F0h]にストアする 5hを5h回加算する。 GR#Aをゼロクリア GR#Bに数値5hを設定 GR#Cに加算回数5hを設定 (GR#Bをコピー) GR#Eに減算用数値1hを設定 GR#A + GR#B (ループ処理開始) GR#Cの加算回数 (カウント) を-1 残りの加算回数が1以上 (>0の正数) ならラベルg4へ分岐 (ループを繰り返す) [F0h] ← GR#A 終了停止, 答え: [F0]=0019h, [ctrl]+sで確認
6.6.4 (PE2) メインメモリへの連続アクセス Viplusは, この目的のためにレジスタ間接メモリアクセス命令 (38命令, 3D命令) を備えているがV1にはない。ここでは, それに替わる以下の方法でこの機能を実現する。 (MM-47) Inst_Advanceシートの[IA1]に同様のプログラムあり。 例題 PE2.2 (1) 09h番地からの5ワードの和を求め[F0h]にストアする GRワードに加算命令を設定してメモリアクセス番地y2を歩進しつつMM_RAM領域にコピーして

[illegible]

[illegible]