クラス番号: 1EP1-99　　学籍番号: 1000000　　提出日:2018/12/18

氏名: Prog_Exercise1シート(解答)　　Print_Buffer

Program Name: **Programming_Exercise1 ー プログラミング演習1**

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版縦印刷の場合 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版横印刷の場合 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	Comment
MM-00												**(284行以下に理解度確認問題あり)**
MM-01												
MM-02												**◎例題・問題の表記について**
MM-03												**・汎用レジスタGRの操作:** 正式には GR#A ← c(GR#A) + c(GR#B) とその内容を表すc( )を用いて
MM-04												書くが, ここではc( )を省略して GR#A ← GR#A + GR#B と記述する.
MM-05												
MM-06												**・メインメモリMMの操作:** 正式には F0h ← c(0Ch) + c(0Dh) と番地のみ, またはその番地の
MM-07												ワード内容を表すc( )を付けて書くが, ここではMMのワードを示す[ ]を付けて
MM-08												[F0h] ← [0Ch] + [0Dh] と記述する.
MM-09												ラベルを用いる場合は [x0] ← [v1] + [v2] と記述する.
MM-0A												
MM-0B												
MM-0C	v1	1	1	1	1	v	--	--				**◎このシートの例題・問題で用いる数値サンプル**
MM-0D	v2	2	2	2	2	v	--	--				
MM-0E	v3	3	3	3	3	v	--	--				**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.**
MM-0F	v4	4	4	4	4	v	--	--				**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**
MM-10		5	5	5	5	v	--	--				**番地セルを選択してからTranslate_barをクリックする**
MM-11		A	A	A	A	v	--	--				**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**
MM-12		0	0	0	1	v	--	--				**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**
MM-13		0	0	0	2	v	--	--				**「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,**
MM-14		0	0	0	3	v	--	--				**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. 」**
MM-15		0	0	0	4	v	--	--				
MM-16												**6.6.1 (PE1) 基本的な数値計算プログラム (1ワード演算)**
MM-17												以下の例題PE1.1と問題PE1.1では数値は2バイト符号なし絶対値形式とし, 桁あふれは考慮しない
MM-18												**例題 PE1.1 (1) GR#AとGR#Bの内容交換**
MM-19												一方のGRワード内容を他のGRワードに退避させてから上書きする.
MM-1A		8	A	v	1		--	L	GR#A	Lbl-v1 Load	--	(準備) GR#A ← [v1] (ラベルによる数値ワード指定)
MM-1B		8	B	v	2		--	L	GR#B	Lbl-v2 Load	--	(準備) GR#B ← [v2]
MM-1C		1	8	E	B		LR	--	GR#E	GR#B Load R.	--	
MM-1D		1	8	B	A		LR	--	GR#B	GR#A Load R.	--	
MM-1E		1	8	A	E		LR	--	GR#A	GR#E Load R.	--	
MM-1F		0	1	0	0		HLT	--	--	-- End Stop	--	終了停止
MM-20												**例題 PE1.1 (2) GR#E ← GR#A×2−GR#B**
MM-21												2のベキ乗の乗算はレジスタ加算命令で行える. GRワードを1ビット上位シフトしても×2になる.
MM-22		8	A	v	3		--	L	GR#A	Lbl-v3 Load	--	(準備) GR#A ← [v3] (ラベルによる数値ワード指定)
MM-23		8	B	v	4		--	L	GR#B	Lbl-v4 Load	--	(準備) GR#B ← [v4]
MM-24		1	A	A	A		AR	--	GR#A	GR#A Add R.	--	×2
MM-25		1	B	A	B		SR	--	GR#A	GR#B Subtract R.	--	
MM-26		1	8	E	A		LR	--	GR#E	GR#A Load R.	--	
MM-27		0	1	0	0		HLT	--	--	-- End Stop	--	終了停止
MM-28												**例題 PE1.1 (3) [F0h] ← [0Ch]×4** (番地による数値ワード指定)
MM-29												同上
MM-2A		8	A	0	C		--	L	GR#A	MM-0C Load	--	
MM-2B		1	A	A	A		AR	--	GR#A	GR#A Add R.	--	×2
MM-2C		1	A	A	A		AR	--	GR#A	GR#A Add R.	--	×2
MM-2D		D	A	F	0		--	ST	GR#A	MM-F0 Store	--	
MM-2E		0	1	0	0		HLT	--	--	-- End Stop	--	終了停止, 答え:[F0h]=4444h, [ctrl]+sで確認
MM-2F												**例題 PE1.1 (4) [11h]／4を計算し, 商を[F0h]に剰余 (余り) を[F1h]にストアする**
MM-30												商は下位シフト, 剰余は被除数へのマスク処理で求める. (番地による数値ワード指定)
MM-31		8	A	1	1		--	L	GR#A	MM-11 Load	--	
MM-32		1	8	E	A		LR	--	GR#E	GR#A Load R.	--	
MM-33		9	F	0	3		--	LA	GR#F	Imv-03 Load Address	--	マスク値の設定:商を2ビット下位シフトで求めるため 0000 0000 0000 0011b を設定
MM-34		2	3	E	E		SD	--	GR#E	GR#E 1-b Shift down	--	／2
MM-35		2	3	E	E		SD	--	GR#E	GR#E 1-b Shift down	--	／2
MM-36		D	E	F	0		--	ST	GR#E	MM-F0 Store	--	
MM-37		2	A	A	F		AND	--	GR#A	GR#F Logical AND	--	マスク処理
MM-38		D	A	F	1		--	ST	GR#A	MM-F1 Store	--	
MM-39		0	1	0	0		HLT	--	--	-- End Stop	--	終了停止, 答え:[F0h]=2AAAh, [F1h]=0002h, [ctrl]+sで確認
MM-3A												(MM-3A)
MM-3B												**● 問題 PE1.1 (1) [F0h]と[F1h]の内容を交換する**
MM-3C												[F0h]と[F1h]はMMのRAM領域のワードのため, 準備としてサンプルデータをストアしておく.
MM-3D		8	0	v	1		--	L	GR#0	Lbl-v1 Load	--	(準備) GR#0 ← [v1]
MM-3E		8	1	v	2		--	L	GR#1	Lbl-v2 Load	--	(準備) GR#1 ← [v2]
MM-3F		D	0	F	0		--	ST	GR#0	MM-F0 Store	--	(準備) [F0h] ← GR#0
MM-40		D	1	F	1		--	ST	GR#1	MM-F1 Store	--	(準備) [F1h] ← GR#1
MM-41		8	A	F	0	?	--	L	GR#A	MM-F0 Load	--	はじめに2つのデータワードをGRにロードし, 次にこれらが入れ替わるようにストアする
MM-42		8	B	F	1	?	--	L	GR#B	MM-F1 Load	--	
MM-43		D	A	F	1	?	--	ST	GR#A	MM-F1 Store	--	
MM-44		D	B	F	0	?	--	ST	GR#B	MM-F0 Store	--	
MM-45		0	1	0	0		HLT	--	--	-- End Stop	--	終了停止, [ctrl]+sで確認
MM-46												(MM-46)
MM-47												**● 問題 PE1.1 (2) GR#A ← (GR#B+GR#C)×2−GR#D×3**
MM-48												×3の計算は, 式全体を×2で計算し, さらに被乗数(GR#D)を1回減算して行う.
MM-49												または, ×2の項を計算してから3回減算してもよい. (ここでは上の方法で行う)

```
MM-4A        8  B  v  3        --   L   GR#B  Lbl-v3  Load          --   (準備) GR#B ← [v3]  (ラベルによる数値ワード指定)
MM-4B        8  C  v  2        --   L   GR#C  Lbl-v2  Load          --   (準備) GR#C ← [v2]
MM-4C        8  D  v  1        --   L   GR#D  Lbl-v1  Load          --   (準備) GR#D ← [v1]
MM-4D        1  A  B  C        AR   --  GR#B  GR#C   Add R.         --
MM-4E        1  B  B  D        SR   --  GR#B  GR#D   Subtract R.    --
MM-4F        1  A  B  B  ?     AR   --  GR#B  GR#B   Add R.         --      ×2
MM-50        1  B  B  D  ?     SR   --  GR#B  GR#D   Subtract R.    --
MM-51        1  8  A  B        LR   --  GR#A  GR#B   Load R.        --
MM-52        0  1  0  0        HLT  --  --    --     End Stop      --   終了停止, 答え : GR#A=7777h
MM-53
MM-54                                                                  (MM-54)
MM-55                                                                  ● 問題 PE1.1 (3)  [F0h] ← [0Ch]+[0Dh]+[0Eh]     (番地による数値ワード指定)
MM-56                                                                    第1項をGRワードにロードしてから残りの項をレジスタ・メモリ加算する.
MM-57        8  A  0  C  ?     --   L   GR#A  MM-0C  Load          --     第1項をGR#Aへロード
MM-58        A  A  0  D        --   A   GR#A  MM-0D  Add           --
MM-59        A  A  0  E        --   A   GR#A  MM-0E  Add           --
MM-5A        D  A  F  0  ?     --   ST  GR#A  MM-F0  Store         --
MM-5B        0  1  0  0        HLT  --  --    --     End Stop      --   終了停止, 答え : [F0h]=6666h, [ctrl]+sで確認
MM-5C
MM-5D                                                                  (MM-5D)
MM-5E                                                                  ● 問題 PE1.1 (4)  [F0h] ← [0Ch]×10     (番地による数値ワード指定)
MM-5F                                                                    ×2まで計算したらGRワードに退避しておき, ×8を計算したらそれに加算する.
MM-60        8  A  0  C        --   L   GR#A  MM-0C  Load          --
MM-61        1  A  A  A  ?     AR   --  GR#A  GR#A   Add R.         --      ×2
MM-62        1  8  B  A        LR   --  GR#B  GR#A   Load R.        --
MM-63        1  A  A  A  ?     AR   --  GR#A  GR#A   Add R.         --
MM-64        1  A  A  A  ?     AR   --  GR#A  GR#A   Add R.         --
MM-65        1  A  A  B  ?     AR   --  GR#A  GR#B   Add R.         --
MM-66        D  A  F  0        --   ST  GR#A  MM-F0  Store         --
MM-67        0  1  0  0        HLT  --  --    --     End Stop      --   終了停止, 答え : [F0h]=AAAAh, [ctrl]+sで確認
MM-68
MM-69                                                                  (MM-69)
MM-6A                                                                  ● 問題 PE1.1 (5)  [11h]／8を計算し, 商を[F0h]に剰余 (余り) を[F1h]にストアする
MM-6B                                                                    例題 PE1.1 (4) のシフトビット数とマスク処理のビット数を増やす(番地による数値ワード指定)
MM-6C        8  A  1  1        --   L   GR#A  MM-11  Load          --
MM-6D        1  8  E  A        LR   --  GR#E  GR#A   Load R.        --
MM-6E        9  F  0  7        --   LA  GR#F  Imv-07  Load Address  --     マスク値を設定
MM-6F        2  3  E  E  ?     SD   --  GR#E  GR#E   1-b Shift down --     以下／8の計算
MM-70        2  3  E  E  ?     SD   --  GR#E  GR#E   1-b Shift down --
MM-71        2  3  E  E  ?     SD   --  GR#E  GR#E   1-b Shift down --
MM-72        D  E  F  0        --   ST  GR#E  MM-F0  Store         --
MM-73        2  A  A  F  ?     AND  --  GR#A  GR#F   Logical AND    --     マスク処理
MM-74        D  A  F  1        --   ST  GR#A  MM-F1  Store         --
MM-75        0  1  0  0        HLT  --  --    --     End Stop      --   終了停止, 答え : [F0h]=1555h, [F1h]=0002h, [ctrl]+sで確認
MM-76
MM-77                                                                  (MM-77)
MM-78                                                                  6.6.2 (PE1) 条件分岐命令とその応用
MM-79                                                                    以降の問題では, 特に指定がない場合は, 数値は1ワード(2バイト)2の補数形式(正／負あり)
MM-7A                                                                    とする. また, 桁あふれは考慮しない
MM-7B        1  3  7  3  v     --   --                                  ● 問題 PE1.2  次の1)～13)のアセンブラ表記の演算をそれぞれ独立に実行するプログ
MM-7C        B  A  B  A  v     --   --                                      ラムを書き, 実行後のコンディションコードCCの値を調べて書きなさい
MM-7D                                                                      ただし実行前のレジスタの値はGR#A=1373h, GR#B=BABAhとする
MM-7E        8  A  7  B        --   L   GR#A  MM-7B  Load          --   (準備) GR#A ← [7Bh]
MM-7F        8  B  7  C        --   L   GR#B  MM-7C  Load          --   (準備) GR#B ← [7Ch]
MM-80        1  A  A  A  ?     AR   --  GR#A  GR#A   Add R.         --   1) AR A, A      CC=
MM-81        8  A  7  B        --   L   GR#A  MM-7B  Load          --   (準備) GR#A ← [7Bh]
MM-82        1  A  A  B  ?     AR   --  GR#A  GR#B   Add R.         --   2) AR A, B      CC=
MM-83        8  A  7  B        --   L   GR#A  MM-7B  Load          --   (準備) GR#A ← [7Bh]
MM-84        1  B  A  A  ?     SR   --  GR#A  GR#A   Subtract R.    --   3) SR A, A      CC=
MM-85        8  A  7  B        --   L   GR#A  MM-7B  Load          --   (準備) GR#A ← [7Bh]
MM-86        1  B  A  B  ?     SR   --  GR#A  GR#B   Subtract R.    --   4) SR A, B      CC=
MM-87        8  A  7  B        --   L   GR#A  MM-7B  Load          --   (準備) GR#A ← [7Bh]
MM-88        1  C  A  A  ?     CR   --  GR#A  GR#A   Compare R.     --   5) CR A, A      CC=
MM-89        1  C  A  B  ?     CR   --  GR#A  GR#B   Compare R.     --   6) CR A, B      CC=
MM-8A        2  2  0  A  ?     SU   --  GR#0  GR#A   1-b Shift up   --   7) SU 0, A      CC=
MM-8B        2  2  1  B  ?     SU   --  GR#1  GR#B   1-b Shift up   --   8) SU 1, B      CC=
MM-8C        2  3  0  A  ?     SD   --  GR#0  GR#A   1-b Shift down --   9) SD 0, A      CC=
MM-8D        2  3  1  B  ?     SD   --  GR#1  GR#B   1-b Shift down --   10) SD 1, B     CC=
MM-8E        8  B  7  C        --   L   GR#B  MM-7C  Load          --   (準備) GR#B ← [7Ch]
MM-8F        A  B  7  B  ?     --   A   GR#B  MM-7B  Add           --   11) A B, 7B     CC=
MM-90        8  B  7  C        --   L   GR#B  MM-7C  Load          --   (準備) GR#B ← [7Ch]
MM-91        B  B  7  B  ?     --   S   GR#B  MM-7B  Subtract      --   12) S B, 7B     CC=
MM-92        8  B  7  C        --   L   GR#B  MM-7C  Load          --   (準備) GR#B ← [7Ch]
MM-93        C  B  7  B  ?     --   C   GR#B  MM-7B  Compare       --   13) C B, 7B     CC=
MM-94        0  1  0  0        HLT  --  --    --     End Stop      --   終了停止
MM-95                                                                  (MM-95)
MM-96                                                                  例題 PE1.2   [F0h] ← max([0Dh], [0Eh], [0Fh])
MM-97                                                                    (maxは最大値を意味する. ここでは数値は符号なし絶対値形式とする)
MM-98                                                                    第1項をGRワードにロードし, 残りの項と比較して大なら上書きする.
MM-99        8  A  0  D        --   L   GR#A  MM-0D  Load          --     [0Dh]の2222hをGR#Aへロード
MM-9A        C  A  0  E        --   C   GR#A  MM-0E  Compare       --     GR#Aと[0Eh]の3333hとを比較
MM-9B        7  A  g  0        --   BC  m=A   Lbl-g0  BranchCC=2or0  (> or =)ならラベルg0へ分岐
MM-9C        8  A  0  E        --   L   GR#A  MM-0E  Load          --     GR#Aへ[0Eh]をロード
MM-9D  g0    C  A  0  F        --   C   GR#A  MM-0F  Compare       --     GR#Aと[0Fh]の4444hとを比較
MM-9E        7  A  g  1        --   BC  m=A   Lbl-g1  BranchCC=2or0  (> or =)ならラベルg1へ分岐
MM-9F        8  A  0  F        --   L   GR#A  MM-0F  Load          --     GR#Aへ[0Fh]をロード
MM-A0  g1    D  A  F  0        --   ST  GR#A  MM-F0  Store         --     [F0h]へGR#Aのワードをストア
MM-A1        0  1  0  0        HLT  --  --    --     End Stop      --   終了停止, 答え : [F0h]=4444h, [ctrl]+sで確認
MM-A2                                                                  (MM-A2)
MM-A3                                                                  ● 問題 PE1.3 (1) [F0h] ← max([0Fh],[10h],[11h]) (数値は負の場合を含む2の補数形式)
MM-A4                                                                  負の場合を含む2の補数形式の数値の大小比較は, 正数はMSBを'1'に負数はMSBを'0'にしてから
MM-A5                                                                  比較命令1C(またはC)を用いる. これには数値に8000hを加算してから比較すればよい. 元の値
MM-A6                                                                  に戻すには再度8000hを加算する. 7命令(分岐命令)の分岐条件はキーワード m で検索できる.
MM-A7        9  E  0  1        --   LA  GR#E  Imv-01  Load Address  --   (準備) GR#Eに0001hを設定
MM-A8        2  5  E  E        ESD  --  GR#E  GR#E   E-shift down  --   (準備)  1 ビット下位回転シフトして8000hにする
MM-A9        8  A  0  F        --   L   GR#A  MM-0F  Load          --
MM-AA        1  A  A  E        AR   --  GR#A  GR#E   Add R.         --     8000hを加算
MM-AB        8  B  1  0        --   L   GR#B  MM-10  Load          --
MM-AC        1  A  B  E        AR   --  GR#B  GR#E   Add R.         --     8000hを加算
```

MM-AD		1	C	A	B		CR	--	GR#A	GR#B	Compare R.	--	
MM-AE		7	A	g	2	?	--	BC	m=A	Lbl-g2	BranchCC=2or0	> or =, + or 0	(> or =)なら分岐
MM-AF		1	8	A	B	?	LR	--	GR#A	GR#B	Load R.	--	上書き(GR#A <- GR#B)
MM-B0	g2	8	B	1	1		--	L	GR#B	MM-11	Load	--	
MM-B1		1	A	B	E		AR	--	GR#B	GR#E	Add R.	--	8000hを加算
MM-B2		1	C	A	B		CR	--	GR#A	GR#B	Compare R.	--	
MM-B3		7	A	g	3	?	--	BC	m=A	Lbl-g3	BranchCC=2or0	> or =, + or 0	(> or =)なら分岐
MM-B4		1	8	A	B	?	LR	--	GR#A	GR#B	Load R.	--	上書き(GR#A <- GR#B)
MM-B5	g3	1	A	A	E	?	AR	--	GR#A	GR#E	Add R.	--	8000hを加算（元に戻す）
MM-B6		D	A	F	0		--	ST	GR#A	MM-F0	Store	--	
MM-B7		0	1	0	0		HLT	--	--	--	End Stop	--	終了停止，答え：[F0h]=5555h，[ctrl]+sで確認
MM-B8													(MM-B8)
MM-B9													● 問題 PE1.3 (2) [F0h] ← |[11h]−[0Ch]| （差を求め，負数の場合は絶対値に変換する）
MM-BA													負数判定は差を求める演算（減算）のCCで行う．
MM-BB		8	A	1	1		--	L	GR#A	MM-11	Load	--	
MM-BC		B	A	0	C	?	--	S	GR#A	MM-0C	Subtract	--	
MM-BD		7	A	g	4	?	--	BC	m=A	Lbl-g4	BranchCC=2or0	> or =, + or 0	結果が正数またはゼロなら分岐
MM-BE		9	F	0	0		--	LA	GR#F	Imv-00	Load Address	--	以下絶対値変換
MM-BF		1	B	F	A	?	SR	--	GR#F	GR#A	Subtract R.	--	
MM-C0		1	8	A	F	?	LR	--	GR#A	GR#F	Load R.	--	
MM-C1	g4	D	A	F	0		--	ST	GR#A	MM-F0	Store	--	
MM-C2		0	1	0	0		HLT	--	--	--	End Stop	--	終了停止，答え：[F0h]=6667h，[ctrl]+sで確認
MM-C3													(MM-C3)
MM-C4													● 問題 PE1.3 (3) [F0h] ← |[11h]| （負数の場合は絶対値に変換する）
MM-C5													加減算結果ではないデータワードの負数判定は1ビット上位シフト(元のデータは保存)で行う．
MM-C6		8	A	1	1		--	L	GR#A	MM-11	Load	--	
MM-C7		2	2	E	A	?	SU	--	GR#E	GR#A	1-b Shift up	--	符号ビット（最上位ビット）検出
MM-C8		7	8	g	5	?	--	BC	m=8	Lbl-g5	Branch CC=0	=, 0	正数またはゼロ（符号ビットが'0'：CC=0）なら分岐
MM-C9		9	F	0	0		--	LA	GR#F	Imv-00	Load Address	--	以下絶対値変換
MM-CA		1	B	F	A	?	SR	--	GR#F	GR#A	Subtract R.	--	
MM-CB		1	8	A	F	?	LR	--	GR#A	GR#F	Load R.	--	
MM-CC	g5	D	A	F	0		--	ST	GR#A	MM-F0	Store	--	
MM-CD		0	1	0	0		HLT	--	--	--	End Stop	--	終了停止，答え：[F0h]=5556h，[ctrl]+sで確認
MM-CE													(MM-CE)
MM-CF													● 問題 PE1.3 (4) GR#Aの値が奇数ならそのまま，偶数なら2で割って[F0h]にストアする．
MM-D0													奇数判定のためのLSBのCCセットは１ビット下位シフト（元のデータは保存）で行う．
MM-D1		8	A	v	3		--	L	GR#A	Lbl-v3	Load	--	（準備）奇数の場合の例：GR#A ← [v3]
MM-D2	g8	2	3	B	A	?	SD	--	GR#B	GR#A	1-b Shift down	--	奇数判定
MM-D3		7	4	g	6	?	--	BC	m=4	Lbl-g6	Branch CC=1	<, -	奇数（CC=1）なら分岐
MM-D4		D	B	F	0		--	ST	GR#B	MM-F0	Store	--	
MM-D5		7	F	g	7		--	BC	m=F	Lbl-g7	Branch	forced	
MM-D6	g6	D	A	F	0		--	ST	GR#A	MM-F0	Store	--	
MM-D7	g7	0	1	0	0		HLT	--	--	--	End Stop	--	答え：奇数の場合[F0h]=3333h，偶数の場合[F0h]=2222h，[ctrl]+sで確認
MM-D8													
MM-D9		8	A	v	4		--	L	GR#A	Lbl-v4	Load	--	（準備）偶数の場合の例：GR#A ← [v4]
MM-DA		7	F	g	8		--	BC	m=F	Lbl-g8	Branch	forced	
MM-DB													(MM-DB)
MM-DC													● 問題 PE1.3 (5) GR#A ≠ GR#B のとき，大きい方を[F0h]にストアする
MM-DD													（負数の場合あり，GR#A，GR#Bの内容は保存する．＝のときはストアしない）
MM-DE		8	A	1	0		--	L	GR#A	MM-10	Load	--	（準備）GR#A ← [10h]
MM-DF		8	B	1	1		--	L	GR#B	MM-11	Load	--	（準備）GR#B ← [11h]
MM-E0		9	E	0	1		--	LA	GR#E	Imv-01	Load Address	--	（準備）GR#Eに0001hを設定
MM-E1		2	5	E	E		ESD	--	GR#E	GR#E	E-shift down	--	（準備）１ビット下位回転シフトして8000hにする
MM-E2		1	8	6	A		LR	--	GR#6	GR#A	Load R.	--	GR#Aの内容を保存するためGR#6にコピー
MM-E3		1	8	7	B		LR	--	GR#7	GR#B	Load R.	--	GR#Bの内容を保存するためGR#7にコピー
MM-E4		1	A	6	E		AR	--	GR#6	GR#E	Add R.	--	8000hを加算
MM-E5		1	A	7	E		AR	--	GR#7	GR#E	Add R.	--	8000hを加算
MM-E6		1	C	6	7	?	CR	--	GR#6	GR#7	Compare R.	--	比較
MM-E7		7	8	g	9	?	--	BC	m=8	Lbl-g9	Branch CC=0	=, 0	(=)なら分岐
MM-E8		7	4	g	A	?	--	BC	m=4	Lbl-gA	Branch CC=1	<, -	(<)なら分岐
MM-E9		D	A	F	0		--	ST	GR#A	MM-F0	Store	--	
MM-EA		7	F	g	9		--	BC	m=F	Lbl-g9	Branch	forced	
MM-EB	gA	D	B	F	0		--	ST	GR#B	MM-F0	Store	--	
MM-EC	g9	0	1	0	0		HLT	--	--	--	End Stop	--	終了停止，答え：[F0h]=5555h，[ctrl]+sで確認
MM-ED													
MM-EE													<<End of Prog_Exercise1>>
MM-EF													理解度確認問題 (Prog_exercise1)　２８４行以下に示す．
MM-F0													
MM-F1													
MM-F2													
MM-F3													
MM-F4													
MM-F5													
MM-F6													
MM-F7													
MM-F8													
MM-F9													
MM-FA													
MM-FB													
MM-FC													
MM-FD													
MM-FE													
MM-FF													