コンピュータシステム基礎後編演習問題Ⅰ　クラス名列番号：　　氏名：

注意事項　命令一覧表（Instruction Table を印刷する）を参照して，各問題の空欄に適切な 16 進コードまたは数値を記入しなさい．なお各問題のプログラムでは，エミュレータのシートで記述してある MM 番地列と Operation 列は省略してある．なお，Lbl は Label 列，v-s は value-sign 列，RR または RM はアセンブラ表記の Mnemonic 列，Op1 は第 1 オペランド列，Op2 は第 2 オペランド列，Branch C は分岐命令での分岐条件列である．　また，GR ワード番号の指定がない場合は GR#6～GR#F を下記として使う．

GR#6: Op1上位ワード
GR#7: Op2上位ワード
GR#8:サブルーチン先頭番地
GR#9:サブルーチン復帰番地
GR#A: Op1下位ワード
GR#B: Op2下位ワード
GR#C:カウンタ
GR#D:ワークワード
GR#E:ワークワード
GR#F:ワークワード

手計算の問題：　[問 1]と[問 2]に答えなさい　（番地による数値ワード指定）

Lbl	Instruction				v-s	RR	RM	Op1	Op2	Branch C	Comment
v0	0	0	0	0	v	--	--	--	--	--	数値データ
v1	3	A	F	9	v	--	--	--	--	--	数値データ
v2	2	3	E	1	v	--	--	--	--	--	数値データ
	8	A	v	0		--	L	GR#A	Lbl-v0	--	プログラム開始
	B	A	v	1		--	S	GR#A	Lbl-v1	--	[問 1] この命令実行後の GR#A の値を書きなさい
	A	A	v	2		--	A	GR#A	Lbl-v2	--	[問 2] この命令実行後の GR#A の値を書きなさい
	0	1	0	0		HLT	--	--	--	--	終了停止

注：答えの数値は逆算で検算すること．

問 1 の答え				
問 2 の答え				

機械命令プログラミングの問題

● **問題 PE1.1（2）　GR#A ← (GR#B＋GR#C)×3－GR#D×2**

GR#B，GR#C，GR#D には数値データがロードしてあるものとする．×3 の計算は，加算結果を退避させてから式全体を×2 で計算し，退避してある値を 1 回加算して行う．ここではオーバーフローはないものとする．

	1	A	B	C		AR	--	GR#B	GR#C	--	プログラム開始
			A	B	?		--	GR#A	GR#B	--	退避
	1	B	B	D		SR	--	GR#B	GR#D	--	
			B	B	?		--	GR#B	GR#B	--	×2
	1	A	A	B		AR	--	GR#A	GR#B	--	
	0	1	0	0		HLT	--	--	--	--	終了停止

● **問題 PE1.1 (3)　[F0h] ← [0Ch]＋[0Dh]＋[0Eh]**　　（番地による数値ワード指定）

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.

					?	--				プログラム開始：第 1 項を GR#A へロード
	A	A			?	--	A	GR#A		
	A	A			?	--	A	GR#A		
			F	0	?	--			MM-F0	
	0	1	0	0		HLT	--	--	--	終了停止

● **問題 PE1.1 (4)　[F0h] ← [0Ch]×10**　　（番地による数値ワード指定）

×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.

	8	A	0	C		--	L	GR#A	MM-0C	プログラム開始
			A	A	?		--	GR#A	GR#A	×2
	1	8	B	A		LR	--	GR#B	GR#A	
			A	A	?		--	GR#A	GR#A	
			A	A	?		--	GR#A	GR#A	
			A	B	?		--	GR#A	GR#B	
	D	A	F	0		--	ST	GR#A	MM-F0	
	0	1	0	0		HLT	--	--	--	終了停止

● **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**

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）

	8	A	1	1		--	L	GR#A	MM-11	プログラム開始
	1	8	E	A		LR	--	GR#E	GR#A	
	9	F			?	--	LA	GR#F		マスク値を設定
			E	E	?		--	GR#E	GR#E	以下／8 の計算
			E	E	?		--	GR#E	GR#E	
			E	E	?		--	GR#E	GR#E	
	D	E	F	0		--	ST	GR#E	MM-F0	
			A	F	?		--	GR#A	GR#F	マスク処理
	D	A	F	1		--	ST	GR#A	MM-F1	
	0	1	0	0		HLT	--	--	--	終了停止

● **問題** PE1.3 (1) [F0h] ← max([0Fh], [10h], [11h]) (max は最大値を意味する. 数値は負の場合を含む2の補数形式). 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. 元の値に戻すには再度 8000h を加算する.

	9	E	0	1		--	LA	GR#E	Imv-01	プログラム開始 : (準備) GR#E に 0001h を設定
	2	5	E	E		ESD	--	GR#E	GR#E	(準備) 1 ビット転シフトして 8000h にする
	8	A	0	F		--	L	GR#A	MM-0F	
	1	A	A	E		AR	--	GR#A	GR#E	8000h を加算
	8	B	1	0		--	L	GR#B	MM-10	
	1	A	B	E		AR	--	GR#B	GR#E	8000h を加算
	1	C	A	B		CR	--	GR#A	GR#B	
			g	2	?	--			Lbl-g2	(> or =) なら分岐
					?		--			上書き
g2	8	B	1	1		--	L	GR#B	MM-11	
	1	A	B	E		AR	--	GR#B	GR#E	
	1	C	A	B		CR	--	GR#A	GR#B	
			g	3	?	--			Lbl-g3	(> or =) なら分岐
					?		--			上書き
g3					?		--			8000h を加算 (元に戻す)
	D	A	F	0		--	ST	GR#A	MM-F0	
	0	1	0	0		HLT	--	--	--	終了停止

● **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)

負数判定は差を求める演算 (減算) の CC で行う.

	8	A	1	1		--	L	GR#A	MM-11	プログラム開始
		A	0	C	?	--		GR#A	MM-0C	
			g	4	?	--			Lbl-g4	結果が正数またはゼロ (+ or 0) なら分岐
	9	F	0	0		--	LA	GR#F	Imv-00	以下絶対値変換
			F	A	?		--	GR#F	GR#A	
			A	F	?		--	GR#A	GR#F	
g4	D	A	F	0		--	ST	GR#A	MM-F0	
	0	1	0	0		HLT	--	--	--	終了停止

● 問題 PE1.3 (3)　[F0h] ← |[11h]|　(負数の場合は絶対値に変換する)

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.

	8	A	1	1		--	L	GR#A	MM-11	--	プログラム開始
			E	A	?		_	GR#E	GR#A	--	
			g	5	?	--			Lbl-g5		正数またはゼロ(符号ビットが'0':CC=0)なら分岐
	9	F	0	0		--	LA	GR#F	Imv-00	--	以下絶対値変換
			F	A	?		--	GR#F	GR#A	--	
			A	F	?		--	GR#A	GR#F	--	
g5	D	A	F	0		--	ST	GR#A	MM-F0	--	
	0	1	0	0		HLT	--	--	--	--	終了停止

● 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.

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.

	8	A	v	3		--	L	GR#A	Lbl-v3	--	プログラム開始：(準備) GR#A ← [v3]
			B	A	?		--	GR#B	GR#A	--	奇数判定
			g	6	?	--			Lbl-g6		奇数（CC=1）なら分岐
	D	B	F	0		--	ST	GR#B	MM-F0	--	
	7	F	g	7		--	BC	m=F	Lbl-g7	forced	
g6	D	A	F	0		--	ST	GR#A	MM-F0	--	
g7	0	1	0	0		HLT	--	--	--	--	終了停止

● 問題 PE1.3 (5)　GR#A ≠ GR#B のとき，大きい方を[F0h]にストアする

(数値は負の場合を含む2の補数形式). GR#A, GR#B の内容は保存する. ＝のときはストアしない.

	8	A	1	0		--	L	GR#A	MM-10	--	プログラム開始：(準備) GR#A ← [10h]
	8	B	1	1		--	L	GR#B	MM-11	--	(準備) GR#B ← [11h]
	9	E	0	1		--	LA	GR#E	Imv-01	--	(準備) GR#E に 0001h を設定
	2	5	E	E		ESD	--	GR#E	GR#E	--	(準備) 1 ビット下位回転シフトして 8000h にする
	1	8	6	A		LR	--	GR#6	GR#A	--	GR#A の内容を保存するため GR#6 にコピー
	1	8	7	B		LR	--	GR#7	GR#B	--	GR#B の内容を保存するため GR#7 にコピー
	1	A	6	E		AR	--	GR#6	GR#E	--	8000h を加算
	1	A	7	E		AR	--	GR#7	GR#E	--	8000h を加算
			6	7	?		--	GR#6	GR#7	--	比較
			g	9	?	--			Lbl-g9		(=)なら分岐
			g	A	?	--			Lbl-gA		(<)なら分岐
	D	A	F	0		--	ST	GR#A	MM-F0	--	
	7	F	g	9		--	BC	m=F	Lbl-g9	forced	
gA	D	B	F	0		--	ST	GR#B	MM-F0	--	
g9	0	1	0	0		HLT	--	--	--	--	終了停止

コンピュータシステム基礎後編演習問題Ⅱ　クラス名列番号：　　氏名：

● 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

	8	0	v	i		--	L	GR#0	Lbl-vi	プログラム開始：(準備) GR#0 ← [vi]
	1	8	1	0		LR	--	GR#1	GR#0	GR#1 ← GR#0 (上位バイト用にコピー)
	1	8	2	0		LR	--	GR#2	GR#0	GR#2 ← GR#0 (下位バイト用にコピー)
	9	C			?	--	LA	GR#C		GR#C にシフト回数 (カウント初期値) を設定
	9	F	0	1		--	LA	GR#F	Imv-01	GR#F に減算用 1h を設定
g1	2	3	1	1		SD	--	GR#1	GR#1	GR#1 ← sd GR#1 (下位シフト, ループ処理開始)
	1	B	C	F		SR	--	GR#C	GR#F	GR#C のシフト回数を-1
			g	1	?	--			Lbl-g1	残りのシフト回数が>0 の正数ならラベル g1 へ分岐
	9	3	F	F		--	LA	GR#3	Imv-FF	GR#3 にマスク値を設定
			2	3	?		--	GR#2	GR#3	GR#2 and GR#3 (マスク処理)
	0	1	0	0		HLT	--	--	--	終了停止

● 問題 PE2.1 (2)　1 バイトデータ 57h に含まれる2値の'1'の数を計数し[F0h]にストアする

注：あらゆる 1 バイトデータに対応できるプログラムにすること.

	9	A	5	7		--	LA	GR#A	Imv-57	プログラム開始： GR#A ← 57h
	9	B	0	0		--	LA	GR#B	Imv-00	GR#B をゼロクリア
	9	C			?	--	LA	GR#C		GR#C にシフト回数？(カウント初期値) を設定
	9	E	0	1		--	LA	GR#E	Imv-01	GR#E に加減算用数値 1h を設定
g3	2	3	A	A		SD	--	GR#A	GR#A	GR#A ← sd GR#A (下位シフト, ループ処理開始)
			g	2	?	--			Lbl-g2	CC=0 ならラベル g2 へ分岐
	1	A	B	E		AR	--	GR#B	GR#E	GR#B を+1
g2	1	B	C	E		SR	--	GR#C	GR#E	GR#C のシフト回数を-1
			g	3	?	--			Lbl-g3	残りのシフト回数が>0 の正数ならラベル g3 へ分岐
	D	B	F	0		--	ST	GR#B	MM-F0	[F0h] ← GR#B
	0	1	0	0		HLT	--	--	--	終了停止

● 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

	9	A	0	0		--	LA	GR#A	Imv-00	--	プログラム開始：GR#A をゼロクリア
	9	B	0	5		--	LA	GR#B	Imv-05	--	GR#B に数値 5h を設定
	1	8	C	B		LR	--	GR#C	GR#B	--	GR#C に加算回数 5h を設定（GR#B をコピー）
	9	E	0	1		--	LA	GR#E	Imv-01	--	GR#E に減算用数値 1h を設定
g4	1	A	A	B		AR	--	GR#A	GR#B	--	GR#A + GR#B （ループ処理開始）
			C	E	?		--	GR#C	GR#E	--	GR#C の加算回数（カウント）を-1
			g	4	?	--			Lbl-g4		残りの加算回数が>0 の正数ならラベル g4 へ分岐
	D	A	F	0		--	ST	GR#A	MM-F0	--	[F0h] ← GR#A
	0	1	0	0		HLT	--	--	--	--	終了停止

例題 PE3.1 (1)　[F1h][F0h] ← [05h][04h]＋[07h][06h]　（２ワード加算）

GR#A と GR#6，GR#B と GR#7 をペアレジスタとして用いる（後者が上位ワード用）

２ワード加算サブルーチンを用いる．（[IA2] 1)　参照）そのプログラムで，下位ワード加算前と加算後の第１オペランドの数値から上位ワードへのキャリー（桁上げ）の有無を判定する．キャリーがある場合は加算後の絶対値が加算前の絶対値より小さくなる．

	8	A	0	4		--	L	GR#A	MM-04	--	プログラム開始：GR#A に Op1 下位ワードをﾛｰﾄﾞ
	8	6	0	5		--	L	GR#6	MM-05	--	GR#6 に Op1 上位ワードをロード
	8	B	0	6		--	L	GR#B	MM-06	--	GR#B に Op2 下位ワードをロード
	8	7	0	7		--	L	GR#7	MM-07	--	GR#7 に Op2 上位ワードをロード
	9	D	0	1		--	LA	GR#D	Imv-01	--	GR#D にキャリー（桁上げ）用の 1h を設定
	9	8	s	0		--	LA	GR#8	Lbl-s0	--	GR#8 にサブルーチン先頭ラベル s0 を設定
	0	5	9	8		BALR	--	GR#9	GR#8	forced	サブルーチン s0 へ分岐
	D	A	F	0		--	ST	GR#A	MM-F0	--	Op1 下位ワードを F0h 番地へストア
	D	6	F	1		--	ST	GR#6	MM-F1	--	Op1 上位ワードを F1h 番地へストア
	0	1	0	0		HLT	--	--	--	--	終了停止
											以下，２ワード加算サブルーチン
s0			E	A	?		--	GR#E	GR#A	--	退避
	1	A	A	B		AR	--	GR#A	GR#B	--	下位ワード加算
			A	E	?		--	GR#A	GR#E	--	Op1 と退避してある Op1 とを比較
	7	A	g	4		--	BC	m=A	Lbl-g3	> or =, + or 0	(> or =)ならキャリー無しでラベル g4 へ分岐
			6	D	?		--	GR#6	GR#D	--	Op1 上位ワードにキャリーを加算
g4	1	A	6	7		AR	--	GR#6	GR#7	--	上位ワード加算
	0	7	F	9		BCR	--	m=F	GR#9	forced	GR#9 の復帰番地へ無条件分岐

例題 PE3.1 (2)　[F1h][F0h] ← [07h][06h]／4　（２ワード除算，小数点以下切捨て）

２ワード連結ｎビット下位シフトサブルーチンを用いる（[IA3] 2)　参照）.そのプログラムで，上位ワードの下位シフトでの LSB からのアンダーフロービットを下位シフト済み下位ワードの MSB へ合成する処理を行う.

	8	A	0	6		--	L	GR#A	MM-06	--	プログラム開始:R#A に 0p1 下位ワードをロード
	8	6	0	7		--	L	GR#6	MM-07	--	GR#6 に 0p1 上位ワードをロード
	9	D	0	1		--	LA	GR#D	Imv-01	--	GR#D に数値 1h を設定
	2		E	D	?		--	GR#E	GR#D	--	合成用数値 8000h を作成
	9	C			?	--	LA	GR#C		--	カウンタ用 GR#C に回数 ?h を設定
	9	8	s	1		--	LA	GR#8	Lbl-s1	--	GR#8 にサブルーチン先頭ラベル s1 を設定
	0	5	9	8		BALR	--	GR#9	GR#8	forced	サブルーチン s1 へ分岐
	D	A	F	0		--	ST	GR#A	MM-F0	--	0p1 下位ワードを F0h 番地へストア
	D	6	F	1		--	ST	GR#6	MM-F1	--	0p1 上位ワードを F1h 番地へストア
	0	1	0	0		HLT	--	--	--	--	終了停止
											以下 2 ワード連結 n ビット上位シフトｻﾌﾞﾙｰﾁﾝ
s1	2	3	A	A		SD	--	GR#A	GR#A	--	GR#A の 0p1 下位ワードを 1 ビット下位シフト
	2	3	6	6		SD	--	GR#6	GR#6	--	GR#6 の 0p1 上位ワードを 1 ビット下位シフト
			g	5	?	--			Lbl-g5		CC=0 ならラベル g5 へ分岐
			A	E	?		--	GR#A	GR#E	--	シフトした 0p1 下位ワードに 8000h を合成
g5	1	B	C	D		SR	--	GR#C	GR#D	--	カウンタ用 GR#C を-1
	7	2	s	1		--	BC	m=2	Lbl-s1	>, +	>0 の正数ならラベル s1 へ分岐
	0	7	F	9		BCR	--	m=F	GR#9	forced	GR#9 の復帰番地へ無条件分岐

● **問題 PE3.1 (1)　[F1h][F0h] ← [07h][06h]－[09h][08h]　（２ワード減算）**

下位ワード減算で絶対値が 0p1<0p2 のときは 0p1 上位ワードからボロー（借り）が発生するため前もって-1 する．サブルーチンは使用しない．（[IA2] 2)　参照）

	8	A	0	6		--	L	GR#A	MM-06	--	プログラム開始:GR#A に 0p1 下位ワードをロード
	8	6	0	7		--	L	GR#6	MM-07	--	GR#6 に 0p1 上位ワードをロード
	8	B	0	8		--	L	GR#B	MM-08	--	GR#B に 0p2 下位ワードをロード
	8	7	0	9		--	L	GR#7	MM-09	--	GR#7 に 0p2 上位ワードをロード
	9	D	0	1		--	LA	GR#D	Imv-01	--	GR#D にボロー用 0001h を設定
			A	B	?		--	GR#A	GR#B	--	下位ワードの 0p1 と 0p2 を比較
			g	4	?	--			Lbl-g4		0p1 が小でなければ（> or = なら）分岐
			6	D	?		--	GR#6	GR#D	--	上位ワードの 0p1 からボローを減算
g4	1	B	A	B		SR	--	GR#A	GR#B	--	下位ワード減算
	1	B	6	7		SR	--	GR#6	GR#7	--	上位ワード減算
	D	A	F	0		--	ST	GR#A	MM-F0	--	0p1 下位ワードを F0h 番地へストア
	D	6	F	1		--	ST	GR#6	MM-F1	--	0p1 上位ワードを F1h 番地へストア
	0	1	0	0		HLT	--	--	--	--	終了停止

● **問題 PE3.1 (2) [07h]の値を２ワード演算で 1024 倍し [F1h][F0h]にストアする**

２ワード連結ｎビット上位シフトサブルーチンを使用する **([IA3] 1) 参照)**

	8	A	0	7		--	L	GR#A	MM-07	--	プログラム開始:GR#A に Op1 下位ワードをロード
	9	6	0	0		--	LA	GR#6	Imv-00	--	Op1 上位ワードをゼロクリア
	9	D	0	1		--	LA	GR#D	Imv-01	--	GR#D に数値 1h を設定
	9	C			?	--	LA	GR#C		--	カウンタ用 GR#C にシフト回数を設定
	9	8	s	2		--	LA	GR#8	Lbl-s2	--	GR#8 にサブルーチン先頭ラベル s2 を設定
	0	5	9	8		BALR	--	GR#9	GR#8	forced	サブルーチン s1 へ分岐
	D	A	F	0		--	ST	GR#A	MM-F0	--	Op1 下位ワードを F0h 番地へストア
	D	6	F	1		--	ST	GR#6	MM-F1	--	Op1 上位ワードを F1h 番地へストア
	0	1	0	0		HLT	--	--	--	--	終了停止
											以下 2 ワード連結 n ビット上位シフトサブルーチン
s2			6	6	?		--	GR#6	GR#6	--	上位ワードを 1 ビット上位シフト
			A	A	?		--	GR#A	GR#A	--	下位ワードを 1 ビット上位シフト
	7	8	g	3		--	BC	m=8	Lbl-g3	=, 0	CC=0 ならラベル g3 へ分岐
			6	D	?		--	GR#6	GR#D	--	Overflow した'1'を上位ワードの LSB に合成
g3	1	B	C	D		SR	--	GR#C	GR#D	--	GR#C のシフト回数を-1
	7	2	s	2		--	BC	m=2	Lbl-s2	>, +	残りのシフト回数が>0 の正数ならラベル s2 へ分岐
	0	7	F	9		BCR	--	m=F	GR#9	forced	GR#9 の復帰番地へ無条件分岐

「以上」