

アセンブラ入門 (CASL II) 第3版

別冊

総合問題・演習問題 解答書

2012 年 9 月 1 日 発行 (第 1 版)

電子開発学園出版局

目 次

総合問題 1 (12 ページ)	1
演習問題 2-1 (14 ページ)	2
演習問題 2-4 (20 ページ)	2
演習問題 2-5 (22 ページ)	2
演習問題 2-6-1 (25 ページ)	2
演習問題 2-6-2 (26 ページ)	3
演習問題 2-6-3 (27 ページ)	3
演習問題 2-7-1(1) (29 ページ)	3
演習問題 2-7-1(2) (31 ページ)	3
演習問題 2-7-1(3) (31 ページ)	3
演習問題 2-7-2 (32 ページ)	3
演習問題 2-7-3 (32 ページ)	3
演習問題 2-7-4 (33 ページ)	3
演習問題 2-7-5 (34 ページ)	3
総合問題 2 (34 ページ)	3
[演習 2-1] (34 ページ)	3
[演習 2-2] (34 ページ)	3
[演習 2-3] (34 ページ)	4
[演習 2-4] (34 ページ)	4
[演習 2-5] (34 ページ)	4
演習問題 3-1 (40 ページ)	5
演習問題 3-2 (42 ページ)	5
演習問題 3-3 (47 ページ)	5
演習問題 3-4(a) (54 ページ)	5
演習問題 3-4(b) (58 ページ)	6
総合問題 3 (60 ページ)	6
[演習 3-1] (60 ページ)	6
[演習 3-2] (60 ページ)	6
[演習 3-3] (60 ページ)	6
[演習 3-4] (60 ページ)	6
演習問題 4-1 (62 ページ)	7
演習問題 4-2 (63 ページ)	7
演習問題 4-3 (65 ページ)	7
演習問題 4-5-1 (68 ページ)	7
演習問題 4-5-2 (69 ページ)	7
演習問題 4-5-3 (72 ページ)	7
演習問題 4-5-4 (74 ページ)	7
総合問題 4 (78 ページ)	8

[演習 4-1] (78 ページ)	8
[演習 4-2] (78 ページ)	9
演習問題 5-1-1 (88 ページ)	10
演習問題 5-1-2 (88 ページ)	10
演習問題 5-2-1 (90 ページ)	10
演習問題 5-2-2 (91 ページ)	11
演習問題 5-2-3 (93 ページ)	12
演習問題 5-2-4 (93 ページ)	12
演習問題 5-3-2 (101 ページ)	12
演習問題 5-4 (105 ページ)	13
演習問題 5-5 (107 ページ)	13
総合問題 5 (108 ページ)	13
[演習 5-1] (108 ページ)	13
[演習 5-2] (108 ページ)	13
[演習 5-3] (108 ページ)	14
[演習 5-4] (108 ページ)	14
[演習 5-5] (108 ページ)	14
総合問題 6 (146 ページ)	15
[演習 6-1] (146 ページ)	15
[演習 6-2] (146 ページ)	15
演習問題 7-1-1 (151 ページ)	16
演習問題 7-1-2 (153 ページ)	17
演習問題 7-2-1 (154 ページ)	17
演習問題 7-2-2 (156 ページ)	18
演習問題 7-2-3 (158 ページ)	18
演習問題 7-2-4 (158 ページ)	19
演習問題 7-3-1 (160 ページ)	19
演習問題 7-3-2 (162 ページ)	20
演習問題 7-3-3 (163 ページ)	20
総合問題 7 (164 ページ)	20
[演習 7-1] (164 ページ)	20
[演習 7-2] (164 ページ)	21
[演習 7-3] (164 ページ)	23
演習問題 8-1 (171 ページ)	24
演習問題 8-2-1 (176 ページ)	25
演習問題 8-2-2 (178 ページ)	25
総合問題 8 (180 ページ)	26
[演習 8-1] (180 ページ)	26
[演習 8-2] (180 ページ)	26
[演習 8-3] (180 ページ)	26

[演習 8-4] (180 ページ)	27
[演習 8-5] (180 ページ)	28
演習問題 9-2-1 (189 ページ)	29
演習問題 9-2-2 (191 ページ)	29
演習問題 9-2-3 (193 ページ)	29
演習問題 9-3-1(a) (194 ページ)	30
演習問題 9-3-1(b) (195 ページ)	31
演習問題 9-3-2 (196 ページ)	31
演習問題 9-4-1 (197 ページ)	32
演習問題 9-4-2 (199 ページ)	34
総合問題 9 (199 ページ)	36
[演習 9-1] (199 ページ)	36
[演習 9-2] (199 ページ)	36
[演習 9-3] (199 ページ)	36
[演習 9-4] (200 ページ)	37
[演習 9-5] (200 ページ)	39
[演習 9-6] (200 ページ)	40
[演習 9-7] (200 ページ)	41
[演習 9-8] (200 ページ)	42
演習問題 10-1 (202 ページ)	43
演習問題 10-2 (204 ページ)	43
演習問題 10-3 (210 ページ)	44
演習問題 10-4 (213 ページ)	44
演習問題 10-5 (217 ページ)	45
総合問題 10 (222 ページ)	45
[演習 10-1] (222 ページ)	45
[演習 10-2] (222 ページ)	47
[演習 10-3] (222 ページ)	48
[演習 10-4] (222 ページ)	48
演習問題 11-2-1 (228 ページ)	49
演習問題 11-2-2 (228 ページ)	49
演習問題 11-3 (231 ページ)	49
演習問題 11-4-1 (234 ページ)	49
演習問題 11-4-2 (235 ページ)	49
演習問題 11-4-3 (236 ページ)	49
演習問題 11-5-1 (245 ページ)	49
演習問題 11-5-2 (247 ページ)	50
総合問題 11 (248 ページ)	50
[演習 11-1] (248 ページ)	50
[演習 11-2] (248 ページ)	51

[演習 11-3] (248 ページ)	51
[演習 11-4] (248 ページ)	53
演習問題 12-1-1 (251 ページ)	55
演習問題 12-1-2 (252 ページ)	55
演習問題 12-1-3 (253 ページ)	55
演習問題 12-1-4 (256 ページ)	55
演習問題 12-2-1 (262 ページ)	55
演習問題 12-2-2 (263 ページ)	56
演習問題 12-3 (270 ページ)	56
演習問題 12-4 (274 ページ)	56
総合問題 12 (275 ページ)	56
[演習 12-1] (275 ページ)	56
[演習 12-2] (276 ページ)	56
索引	57

本書・シミュレータ、サンプルプログラムのダウンロード方法

本書および CASLDV シミュレータ、サンプルプログラムは、以下の URL からダウンロードしてください。

<http://www.officeuchida.com/CASL/>

総合問題 1 (12 ページ)

- (1)
- (2) × 機械語はそれぞれのコンピュータごとに異なる。JIS 規格も存在しない。
- (3) × アセンブラ言語で記述したプログラムは実行効率は確かによいが、プログラムの生産性が極度に低くなり、プログラムの生産コストを上昇させる。したがって、すべてをアセンブラ言語で開発することは得策ではない。
- (4)
- (5)
- (6) 基本的にアセンブラ言語は、そのコンピュータのすべての機械語を表現できる。しかし、機械語の中には特権命令と呼ばれる特殊な命令があり、この命令は特権を有したプログラムでないと実行できない。ただし、COMET II は単純化されたコンピュータなので特権命令はない。
- (7)
- (8) × 現在のコンピュータでもプログラム内蔵方式が採用されている。
- (9) × アセンブラ言語で書かれたプログラムはそのコンピュータの機械語に 1 対 1 に変換されるので、実行効率は低下しない。
- (10) × C 言語特有の処理のために、特定のコンピュータに対するチューニングができないので、「常に効率が良くなる」とは言えない。ただし、現在の C 言語コンパイラは大変良くできており、へたな人が作成したアセンブラ言語のプログラムよりも効率の良い機械語を生成する。
- (11) × Windows 系 OS は、パソコンメーカーが異なってもインテル社製の CPU を対象とした OS であるので、機械語はまったく同じである。したがってメーカーが異なっても同じ機械語が動作する。
- (12) × CPU が同じであれば機械語は同じである。しかし、OS が異なるので動作環境や実行時ライブラリ (入出力や GUI のプログラム) が異なる。実行時ライブラリが異なれば、その機械語のプログラムは残念ながら実行できない。

演習問題 2-1 (14 ページ)

時間の表記では、時刻に 12 進法、分・秒に 60 進法を用いている。また、完全な例ではないが、月は 12 進法、日にちは変則的な 30 進法を用いていると言えないこともない。さらに、算盤（そろばん）は、1～4 までの珠と 5 の珠の 2 種類の珠を用いている。これも変則的な 5 進法と言えないこともない。

演習問題 2-3-1 (16 ページ)

(1) (a) $1010_{(2)}=10$ (b) $111011_{(2)}=59$ (c) $10101011_{(2)}=171$

(2)

数値	10 進数	奇数	2 の倍数 (=偶数)	4 の倍数	8 の倍数
$11011_{(2)}$	27		×	×	×
$100000_{(2)}$	32	×			
$111110_{(2)}$	62	×		×	×
$110100_{(2)}$	52	×			×
$100011_{(2)}$	35		×	×	×

演習問題 2-3-2 (18 ページ)

(1) $11_{(2)}+10101_{(2)}=3+21=24_{(10)}=11000_{(2)}$ (2) $1001_{(2)}+110111_{(2)}=9+55=64_{(10)}=1000000_{(2)}$

演習問題 2-4 (20 ページ)

(1) $13_{(8)}=11_{(10)}=1011_{(2)}$

(2) $57_{(8)}=47_{(10)}=101111_{(2)}$

(3) $1234_{(8)}=668_{(10)}=1010011100_{(2)}$

演習問題 2-5 (22 ページ)

(1) $5CCA_{(16)}=101110011001010_{(2)}$

(2) $FDDA_{(16)}=1111110111011010_{(2)}$

(3) $1234_{(16)}=1001000110100_{(2)}$

(4) $4444_{(16)}=100010001000100_{(2)}$

(5) $CAFE_{(16)}=1100101011111110_{(2)}$

演習問題 2-6-1 (25 ページ)

	2 進数	符号無し	符号付き
(1)	$1111_{(2)}$	15	-1
(2)	$1010_{(2)}$	10	-6
(3)	$1110_{(2)}$	14	-2
(4)	$0001_{(2)}$	1	1
(5)	$0101_{(2)}$	5	5

演習問題 2-6-2 (26 ページ)

数値 2 の補数

- (1) 5 $111111111111011_{(2)}$
 (2) 10 $1111111111110110_{(2)}$
 (3) -5 $000000000000101_{(2)}$

演習問題 2-6-3 (27 ページ)

- (1) $5-3 = 000000000000101_{(2)} + 111111111111101_{(2)} = 000000000000010_{(2)}$
 (2) $2-10 = 000000000000010_{(2)} + 1111111111110110_{(2)} = 111111111111000_{(2)}$
 (3) $14-5 = 000000000001110_{(2)} + 111111111111011_{(2)} = 000000000001001_{(2)}$

演習問題 2-7-1(1) (29 ページ)

- (1) $159=10011111_{(2)}$
 (2) $2152=100001101000_{(2)}$
 (3) $10000=10011100010000_{(2)}$

演習問題 2-7-1(2) (31 ページ)

- (1) $159=237_{(8)}$ (2) $2152=4150_{(8)}$ (3) $10000=23420_{(8)}$

演習問題 2-7-1(3) (31 ページ)

- (1) $159=9F_{(16)}$ (2) $2152=868_{(16)}$ (3) $10000=2710_{(16)}$

演習問題 2-7-2 (32 ページ)

- (1) $1011101_{(2)}=135_{(8)} = 5D_{(16)}$
 (2) $11110000111100001111_{(2)}=3607417_{(8)} = F0F0F_{(16)}$
 (3) $10101101011010101_{(2)}=255325_{(8)} = 15AD5_{(16)}$

演習問題 2-7-3 (32 ページ)

- (1) $156_{(8)}=1101110_{(2)}$ (2) $FFAB_{(16)}=111111110101011_{(2)}$

演習問題 2-7-4 (33 ページ)

- (1) $1234_{(8)}=29C_{(16)}$ (2) $ABCD_{(16)}=125715_{(8)}$ (3) $7777_{(8)}=FFF_{(16)}$

演習問題 2-7-5 (34 ページ)

- (1) $110101011_{(2)}=427$ (2) $12345_{(8)}=5349$ (3) $FC_{(16)}=252$

総合問題 2 (34 ページ)

[演習 2-1] (34 ページ)

- (a) (1) $101_{(2)}$ (2) $1010_{(2)}$ (3) $101101_{(2)}$ (4) $1010001010_{(2)}$ (5) $10000000001_{(2)}$
 (b) (1) $7_{(8)}$ (2) $175_{(8)}$ (3) $6405_{(8)}$ (4) $23613_{(8)}$ (5) $303240_{(8)}$
 (c) (1) $64_{(16)}$ (2) $929_{(16)}$ (3) $401_{(16)}$ (4) $FA00_{(16)}$ (5) $186A0_{(16)}$

[演習 2-2] (34 ページ)

- (1) 725 (2) 470 (3) 8240817 (4) 926118497 (5) 51966 (6) 19645263

[演習 2-3] (34 ページ)

- (1) $1011100_{(2)}$
- (2) $101011001_{(2)}$
- (3) $372200_{(8)}$
- (4) $323713_{(8)}$
- (5) $AB013_{(16)}$
- (6) $FD01FFF_{(16)}$
- (7) $12ABD_{(16)}$
- (8) $1000000_{(16)}$

[演習 2-4] (34 ページ)

- (1) $1001100_{(2)}$ (2) $100011_{(2)}$ (3) $11101011_{(2)}$

[演習 2-5] (34 ページ)

80000 を 2 進数で表現すると、 $10011100010000000_{(2)}$ となる。ちなみに 16 進数では $13880_{(16)}$ である。この下位 16 ビットは、16 進数で 4 桁に相当するから $3880_{(16)} = 11100010000000_{(2)}$ である。この値は、10 進数では 14464 である。したがって、答は 14464 となる。

演習問題 3-1 (40 ページ)

- (1) 各汎用レジスタは 16 ビットであるから、符号付きの場合 $-32768 \sim 32767$ まで表現でき、符号無しの場合 $0 \sim 65535$ まで表現できる (表 2-9(24 ページ) 参照)。
- (2) PR は 16 ビットなので、 $0 \sim 65535$ までの数を表現できる。したがって、COMET II の記憶装置のすべての番地を PR で表現することができる。

演習問題 3-2 (42 ページ)

- (1) 「LD GR5,7936」あるいは「LD GR5,#1F00」(注: $7936=1F00_{(16)}$)
- (2) $101003E8_{(16)}$ (注: $1000=03E8_{(16)}$)

演習問題 3-3 (47 ページ)

- (1) PR は次に実行すべき機械語の格納された番地を保持している。つまり、実行中の機械語の命令が終了すると、PR の番地の機械語命令が CPU に取り込まれ解読される。その時に、PR の値は自動的に機械語の語長分だけ増加される。つまり、実行中の機械語命令が 1 語長であれば、PR は「現在実行中の機械語の格納された番地+1」になっている。実行中の機械語が 2 語長であれば、PR は「現在実行中の機械語の格納された番地+2」になっている。ただし、機械語の種類 (ジャンプ命令、CALL 命令、RET 命令) によっては PR を書き直すので注意すること。
- (2) ロードして実行するプログラムが 4000 バイト=4000 語である。もし、61536 番地からロードすると、プログラムは 61536 番地～65535 番地に格納される。しかし、61537 番地以降からロードすると、プログラムの最後の 1 語が格納できなくなる。したがって、0 番地から 61536 番地の間にロードすることができる。ちなみに $61536=65535-4000+1$ である。
- (3) (a) PR が指している記憶装置のデータを機械語命令として取り出す。そのとき、第 1 語目を見れば、その語長が分かるので、必要に応じて第 2 語を取り出す。
 (b) PR の値を語長分だけ増加させる。
 (c) 取り出してきた機械語命令を実行する。
 (d) ステップ (a) に戻る。

演習問題 3-4(a) (54 ページ)

1	SAMPLE	START	
2		LD	GR1,7
3		ADDA	GR1,7
4		ST	GR1,8
5		RET	
6		DC	5
7		DS	1
8		END	

3 行目の「ADDA GR1,7」は「SLA GR1,1」でも良い。

演習問題 3-4(b) (58 ページ)

- (1) (必要性 1) ラベルを使わずに絶対番地を使うと、プログラムの修正などに伴いアドレスがずれた時に絶対番地の修正が困難であり、大変面倒になる。また、絶対番地を使うと、プログラムがロードされるアドレスが固定されてしまう。これは多くのプログラムを同時に実行させるマルチタスクの実現の際などに管理が大変になる。
(必要性 2) 人間にとって番号で記憶するよりも名前で記憶した方が覚えやすいし、プログラムも分かりやすくなる。
- (2) (a) 10DATA 数字から始まっている
(b) X__1 アンダーラインは使えない
(c) Z-1 ハイフンは使えない
(d) ギリシア文字や漢字は使えない
(e) SAMPLEPGM ラベルは 8 文字までしか使えない

総合問題 3 (60 ページ)

[演習 3-1] (60 ページ)

CASL II アセンブラは、CASL II プログラムを入力して、COMET II の機械語を生成する。COMET II シミュレータは、その機械語を実行する。

[演習 3-2] (60 ページ)

CASLDV を起動して、右上のペイン (ウインドウ) にプログラムを入力する。メニューから「ファイル 名前を付けて保存」を選び (あるいは、書込みボタンを押す)、SAMPLE.casl というファイル名で保存する (ファイル名は ~.casl であれば何でもよい。メニューから「アセンブル アセンブル (CASL 完全準拠)」を選ぶ (あるいはアセンブルボタンを押す)。これでアセンブル結果が左上のペインに描画される。メニューから「実行 実行開始」を選ぶ (あるいは、実行ボタンを押す)。これで、プログラムが実行される。画面上の 8 番地の内容が「000A」になっているはずである。

[演習 3-3] (60 ページ)

- (1) × フラグレジスタは 3 ビットである。それ以外のレジスタは 16 ビットである。一般にレジスタというと PR や FR、SP も含んだ全てのレジスタを指すことに注意。
- (2) × GR0 は指標レジスタとしては使えない。
- (3) × プログラムをロードしてもそのまま記憶装置に残っている。
- (4) × 1 語の命令もある。
- (5)
- (6)
- (7)
- (8) × アンダーラインは使えない。

[演習 3-4] (60 ページ)

- (a) 0 (b) 65535 (c) 8 (d) 3 (e) 8 (f) オペレーティングシステム

演習問題 4-1 (62 ページ)

- (ア) ラベルは 1 カラム目から書き出す。
- (イ) プログラムは大文字で記述する。
- (ウ) 命令は 2 カラム以降から記述する。
- (エ) オペランドの間には空白を書かない。
- (オ) 1 行で記述する。
- (カ) オペランドがない場合、コメントは「;」の後に記述する。
- (キ) START 命令には必ずラベルを書く。
- (ク) END 命令にはラベルを書かない。
- (ケ) GR0 ~ GR7 はラベル名としては使えない。

演習問題 4-2 (63 ページ)

- (1) 100000 を 16 進数に直すと、186A0₍₁₆₎ である。この下位 4 バイトは 86A0₍₁₆₎ である。実際には、この数値が設定される。これを 10 進数で見ると、符号付きのデータの場合 -31072 となり、符号無しデータの場合 34464 となる。なお、これをチェックするプログラムが OVRCHK.casl に格納されている。
- (2) 30000 を 16 進数に直すと、7530₍₁₆₎ である。したがって、CASL II の 16 進数では、#7530 と表記される。
- (3) -1 を 16 進数に直すと、FFFF₍₁₆₎ である。したがって、CASL II の 16 進数では、#FFFF と表記される。

演習問題 4-3 (65 ページ)

- (1)
- (2) ただし、最後の「, GR3」がコメントになるが見た目に分かり難いだけでなく、GR3 を指標レジスタと勘違いする恐れがある。
- (3) × RET 命令はオペランドをとらない。また、コメントを書く場合には、「;」で開始する。
- (4)

演習問題 4-5-1 (68 ページ)

- (1) 最初のプログラムの名前は PRG3 で実行開始アドレスは 0 番地である。次のプログラムの名前は PRG4 で実行開始アドレスは 3 番地である。
- (2) START 命令に記述する実行開始アドレスはラベルでなければならない。この例では絶対番地を指定している。

演習問題 4-5-2 (69 ページ)

END 命令はアセンブラに対して、「ここでプログラムが終了しますよ」ということを伝えるための命令であり、これに対応する機械語はない。機械語命令がなければ、それに対するアドレスを決定することができない。

演習問題 4-5-3 (72 ページ)

- (1) 4 語 (2) 8 語 (3) 3 語 (アドレス定数は 1 語分占める)

演習問題 4-5-4 (74 ページ)

DS 命令のあとに記述できるのは 0 以上の 10 進定数だけである。

- (1) 文字定数は記述できない
- (2) 16 進定数は記述できない
- (3) 負の数は記述できない

総合問題 4 (78 ページ)

[演習 4-1] (78 ページ)

以下の説明の中で数値はすべて 16 進数とする。

START 命令は、対応する機械語がないので、何もしない。

「LD GR1,A」は、主記憶装置の内容 (A の値) を汎用レジスタに転送する命令であり、この機械語は「10」で始まる。次に汎用レジスタの番号 1 と (後で学習する指標レジスタを使用しないという意味の)0 が続き、第 0 語目は「1010」となる。次にラベル A のアドレスを記述するが、この時点ではまだ A が配置されるアドレスが分からないので第 1 語目は保留にしておく。

「ADDA GR1,B」は、主記憶装置の内容 (B の値) を汎用レジスタに算術加算する命令であり、この機械語は「20」で始まる。次に汎用レジスタの番号 1 と (後で学習する指標レジスタを使用しないという意味の)0 が続き、第 2 語目は「2010」となる。次にラベル B のアドレスを記述するが、この時点ではまだ B が配置されるアドレスが分からないので保留にしておく。

「ST GR1,C」は、汎用レジスタの内容を主記憶装置 (C) に転送する命令であり、これは「11」で始まる。次に汎用レジスタの番号 1 と (後で学習する指標レジスタを使用しないという意味の)0 となり、第 4 語目は「1110」となる。次にラベル C のアドレスを記述するが、この時点ではまだ C が配置されるアドレスが分からないので保留にしておく。

RET 命令は 1 語の命令であり、これは「8100」である。これが第 6 語目に配置される。

次にラベル A の定義があるが、ここで初めてラベル A が第 7 語目に配置されることがわかる。そこで、最初の命令「LD GR1,A」は「1010 0007」となり、これが第 0 語目～第 1 語目となる。また、「A DC 5」は値 5 がそのままデータとして残るので第 7 語目は「0005」となる。

同様に「B DC 3」は、「0003」となり、これが第 8 語目に配置される。

B の相対アドレスが 8 となるので、この時点で「ADDA GR1,B」は「2010 0008」となる。

「C DS 1」によって C のアドレスが 9 であり、ここから 1 語分が確保されることがわかる。ただし、DS 命令なので、第 9 語目に配置されるデータは不定となる (つまり、何がセットされるかは分からない)。しかし、C のアドレスは分かったので、「ST GR1,C」は「1110 0009」となる。

以上をまとめると、以下 (表 4-1 と同じ) に示す機械語が生成できる。ただし、ここで生成した A、B、C のアドレスは相対アドレスであり、さらにロードによって実際のアドレスに変換されることに注意されたい。

したがって、以下の機械語が生成される。「????」の部分には何が入るか分からないことを示す。

1010 0007 2010 0008 1110 0009 8100 0005 0003 ?????

[演習 4-2] (78 ページ)

- (1) × ラベルは必ず第 1 カラムから記述する
- (2) × 最大 8 文字まで
- (3) ラベル名として使えないのは、汎用レジスタの名前だけである
- (4) × を必ずしも付ける必要はない
- (5) × を必ず付ける
- (6) × DS 命令には、このような書き方は許されていない
- (7)
- (8)
- (9) × オペランドがある場合には、セミコロン ; を付けずに直接コメントを書くことができる
- (10) × 「DC 'AB' 'C' 」と書くべきである

演習問題 5-1-1 (88 ページ)

(1)

1	EX511A	START	
2		LD	GR1,X
3		SUBA	GR1,Y
4		ADDA	GR1,Z
5		ST	GR1,W
6		RET	
7	X	DC	10
8	Y	DC	11
9	Z	DC	12
10	W	DS	1
11		END	

(2)

1	EX511B	START	
2		LD	GR1,X
3		SUBA	GR1,Y
4		SUBA	GR1,Z
5		ST	GR1,W
6		RET	
7	X	DC	10
8	Y	DC	11
9	Z	DC	12
10	W	DS	1
11		END	

演習問題 5-1-2 (88 ページ)

1	EX512	START	
2		LD	GR1,A
3		ADDA	GR1,B GR1 A+B
4		LD	GR2,C
5		ADDA	GR2,D GR2 C+D
6		LD	GR3,GR1 GR3 GR1(A+B)
7		ADDA	GR3,GR1 GR3 $2 \times (A+B)$
8		SUBA	GR3,GR2 GR3 $2 \times (A+B) - (C+D)$
9		SUBA	GR3,GR2 GR3 $2 \times (A+B) - 2 \times (C+D)$
10		SUBa	GR3,GR2 GR3 $2 \times (A+B) - 3 \times (C+D)$
11		ST	GR3,E
12		RET	
13	A	DC	10
14	B	DC	11
15	C	DC	12
16	D	DC	13
17	E	DS	1
18		END	

演習問題 5-2-1 (90 ページ)

たとえば、4 ビットのデータで考える。 $b_3b_2b_1b_0$ という 2 進数 ($b_i = 1$ か 0) は $b_3 \times 2^3 + b_2 \times 2^2 + b_1 \times 2^1 + b_0 \times 2^0$ である。この各ビットを左に 1 ビットシフトするということは $b_3b_2b_1b_00$ という 2 進数になり、これは 10 進数では $b_3 \times 2^4 + b_2 \times 2^3 + b_1 \times 2^2 + b_0 \times 2^1$ になり、全体を 2 で括ると $2 \times (b_3 \times 2^3 + b_2 \times 2^2 + b_1 \times 2^1 + b_0 \times 2^0)$ となるので、元の数の 2 倍になっていることがわかる。また、 $b_3b_2b_1b_0$ を 1 ビット右にシフトするということは $b_3b_2b_1$ という 2 進数になり、これは 10 進数では $b_3 \times 2^2 + b_2 \times 2^1 + b_1 \times 2^0$ になり、元の数の半分になっている。

演習問題 5-2-2 (91 ページ)

(1)

1	EX522A	START	
2		LD	GR1,N
3		SLA	GR1,2
4		ST	GR1,M
5		RET	
6	N	DC	10
7	M	DS	1
8		END	

(2)

1	EX522B	START	
2		LD	GR1,N
3		SLA	GR1,3
4		ST	GR1,M
5		RET	
6	N	DC	10
7	M	DS	1
8		END	

(3)

1	EX522C	START	
2		LD	GR1,A
3		ADDA	GR1,B
4		SLA	GR1,3
5		ST	GR1,C
6		RET	
7	A	DC	3
8	B	DC	7
9	C	DS	1
10		END	

演習問題 5-2-3 (93 ページ)

(1)

1	EX523A	START		
2		LD	GR1,N	
3		LD	GR2,GR1	GR2 N
4		SLA	GR2,1	GR2 2N
5		LD	GR3,GR1	GR3 N
6		SLA	GR3,3	GR3 8N
7		ADDA	GR1,GR2	GR1 $N+2N=3N$
8		ADDA	GR1,GR3	GR1 $3N+8N=11N$
9		ST	GR1,M	
10		RET		
11	N	DC	10	
12	M	DS	1	
13		END		

(2)

1	EX523B	START		
2		LD	GR1,A	
3		ADDA	GR1,B	
4		LD	GR2,GR1	GR2 (A+B)
5		SLA	GR2,1	GR2 $2(A+B)$
6		LD	GR3,GR1	GR3 (A+B)
7		SLA	GR3,3	GR3 $8(A+B)$
8		ADDA	GR1,GR2	GR1 $(A+B)+2(A+B)=3(A+B)$
9		ADDA	GR1,GR3	GR1 $3(A+B)+8(A+B)=11(A+B)$
10		ST	GR1,C	
11		RET		
12	A	DC	3	
13	B	DC	7	
14	C	DS	1	
15		END		

演習問題 5-2-4 (93 ページ)

(1)

1	EX524A	START	
2		LD	GR1,N
3		SRA	GR1,3
4		ST	GR1,M
5		RET	
6	N	DC	80
7	M	DS	1
8		END	

(2)

1	EX524B	START	
2		LD	GR1,A
3		ADDA	GR1,B
4		SRA	GR1,4
5		ST	GR1,C
6		RET	
7	A	DC	80
8	B	DC	1520
9	C	DS	1
10		END	

演習問題 5-3-2 (101 ページ)

(1) 100 (2) 200 (3) 102 (4) 201 (5) 204 (6) 105

ただし、設問 (4) と (6) では、指標レジスタの値が目的としているテーブルの範囲を超えている。

たまたま、TABLE1 と TABLE2 が連続して配置されていたので、このような結果となったのであり、本来はこのようなプログラムは書くべきではない。しかし、TABLE1 と TABLE2 が初めから連続して配置してあり、一つの連続したテーブルであることを想定している場合は、問題はない(が、よほどうまくロジックを組まないとプログラムが分かり難くなる)。

演習問題 5-4 (105 ページ)

「LD GR1,X」は実行されない。JUMP 命令は無条件にジャンプしてしまうため、その直後の命令は実行されない。CASLDV にこのプログラムをかけると警告メッセージが表示される。

演習問題 5-5 (107 ページ)

- (1) (a) LD GR0,='a' (b) LD GR1,=-45 (c) LD GR2,#FFFF
 (2) 「LD GR2,'CASL II」は、CASL II アセンブラによって次のように変換される。

1		LD	GR2,TEMPDATA
2		:	
3	TEMPDATA	DC	'C'
4		DC	'A'
5		DC	'S'
6		DC	'L'
7		DC	' '
8		DC	'I'
9		DC	'I'

したがって、GR2 には 'C' の文字コード $43_{(16)}$ が入る。

総合問題 5 (108 ページ)

[演習 5-1] (108 ページ)

- (1) X が $9_{(16)}$ 番地、Y が $A_{(16)}$ 番地、Z が $B_{(16)}$ 番地、W が $C_{(16)}$ 番地
 (2) X が $3F1_{(16)}$ 番地、Y が $3F2_{(16)}$ 番地、Z が $3F3_{(16)}$ 番地、W が $3F4_{(16)}$ 番地
 (3) X が $2009_{(16)}$ 番地、Y が $200A_{(16)}$ 番地、Z が $200B_{(16)}$ 番地、W が $200C_{(16)}$ 番地

[演習 5-2] (108 ページ)

へたに (第 9 章で学習する) ループを使ってプログラムを組むよりも、次のプログラムの方が単純で高速である。

1		LD	GR0,X
2		LD	GR0,X,GR1
3		LD	GR0,X,GR2
4		LD	GR0,X,GR3
5		LD	GR0,X,GR4

[演習 5-3] (108 ページ)

- (1) × たとえば、レジスタ GR0 の値を 32 倍するには「SLA GR0,5」とすれば 1 命令でよい。
- (2)
- (3) ただし、「DC 1,2」は 2 語確保し、そこにデータ 1、2 をセットするのに対し、「DS 2」は 2 語データを確保するが、そこに設定される値は不定である。
- (4) × 指標レジスタとして使えるのは、GR1 から GR7 までの 7 個の汎用レジスタである。GR0 は指標レジスタとしては使えない。
- (5) × 「LD GR1,X,GR1」ができないという記述は、COMET II の定義のどこにも書かれていない。たとえば、LD 命令の例ではないが、実際に「LAD GR1,1,GR1」という記述はよく用いる。
- (6) × この記述もどこにも書かれていない。
- (7)
- (8)
- (9) × 「LD GR1,150」によって生成される実効アドレスは 150 である。
- (10)

[演習 5-4] (108 ページ)

表の先頭のアドレスを「0 番地」とするのは、指標レジスタの計算を単純化するためである。もし、表の先頭を「1」とすると、実効アドレスの計算は「アドレス + 指標レジスタ + 1」となって余分な「+1」の計算が必要となる。

[演習 5-5] (108 ページ)

プログラムの命令の個数をまとめると次の表のようになる。

	MUL10	MUL10A
ロード (LD)	1 回	1 回
加算 (ADDA)	4 回	3 回
シフト (SLA)	1 回	2 回
保存 (ST)	2 回	2 回

上記の表に示す通り、MUL10の方が加算が 1 回多く、シフトが 1 回少ない。シフトより加算の方が実行時間がかかると推定されるので、MUL10Aの方が若干効率が良いと考えられる。

総合問題 6 (146 ページ)

[演習 6-1] (146 ページ)

- (1) 30000
- (2) -30001
- (3) -25536 符号無しでは 40000
- (4) 1250
- (5) 9488
- (6) 30512
- (7) 21024

[演習 6-2] (146 ページ)

効率よく 100 倍するには、シフト命令を用いてべき乗の結果を足せばよい。100 倍の場合、元の数を x とすると、 $x^4 + x^{32} + x^{64}$ で求められる。

参考となるプログラムを以下に示す。

1	P62	START	
2		LAD	GR0,100
3		LD	GR1,GR0
4		LD	GR2,GR0
5		LD	GR3,GR0
6		SLA	GR1,2
7		SLA	GR2,5
8		SLA	GR3,6
9		ADDA	GR1,GR2
10		ADDA	GR1,GR3
11		ST	GR1,ANS
12		RET	
13	ANS	DS	1
14		END	

演習問題 7-1-1 (151 ページ)

(1) (a)

1	LAD	GR7,12345
---	-----	-----------

(b)

1	LD	GR7,CONST
2	CONST	DC 12345

(c)

1	LD	GR7,=12345
---	----	------------

(2) (方法 1)

1	LD	GR0,VALUE
2	LD	GR1,GR0
3	LD	GR2,GR0
4	LD	GR3,GR0
5	LD	GR4,GR0
6	LD	GR5,GR0
7	LD	GR6,GR0
8	LD	GR7,GR0

(方法 2)

1	LD	GR0,VALUE
2	LD	GR1,VALUE
3	LD	GR2,VALUE
4	LD	GR3,VALUE
5	LD	GR4,VALUE
6	LD	GR5,VALUE
7	LD	GR6,VALUE
8	LD	GR7,VALUE

方法 1 では、最初の命令 (1 行目) だけが 2 語であとはすべて 1 語 (2 行目 ~ 8 行目) であるので、プログラム全体の語数は 9 語である。また、主記憶装置とのアクセスは最初の 1 命令だけであとは全部 CPU 内のデータ複写となっている。

これに対し、方法 2 では、すべての命令 (1 行目から 8 行目) が 2 語なので、プログラム全体の語数は 16 語となっている。さらに、すべての命令で主記憶装置とのアクセスが発生している。

このことから、方法 1 の方がかなり実行効率および記憶効率が良いと言える。

(3)

プログラム 1 では 3 つの LD 命令が TEN という領域のデータを汎用レジスタに転送している。データが定義されている領域 (TEN の領域) は 1 語分しかない。

しかし、プログラム 2 では、それぞれの LD 命令で別個にリテラルを用いている。これらの値はすべて 10 という同じ数であるが、最適化機能がないということから、これらは次に示すように別々の領域に確保される可能性が高い。

LD	GR1,=10	CASL II アセンブラ	LD	GR1,TEN1	} 重複したデータ
LD	GR2,=10		LD	GR2,TEN2	
LD	GR3,=10		LD	GR3,TEN3	
		TEN1	DC	10	
		TEN2	DC	10	
		TEN3	DC	10	

このため、重複したデータが主記憶装置内に存在してしまうことになる。

(4) 仮に最適化機能があれば、次のように同じ値を用いているリテラルを 1 つの領域に統合することも考えられる。

LD	GR1,=10	CASL II アセンブラ	LD	GR1,TEN1	
LD	GR2,=10		LD	GR2,TEN2	
LD	GR3,=10		LD	GR3,TEN3	
			TEN	DC	10

このようにすれば、データ 10 のための記憶装置は 1 語で済む。

演習問題 7-1-2 (153 ページ)

(1)

(方法 1)

1	XOR	GR0,GR0
2	XOR	GR1,GR1
3	XOR	GR2,GR2
4	XOR	GR3,GR3
5	XOR	GR4,GR4
6	XOR	GR5,GR5
7	XOR	GR6,GR6
8	XOR	GR7,GR7

(方法 2)

1	XOR	GR0,GR0
2	LD	GR1,GR0
3	LD	GR2,GR0
4	LD	GR3,GR0
5	LD	GR4,GR0
6	LD	GR5,GR0
7	LD	GR6,GR0
8	LD	GR7,GR0

COMET II の仕様書には詳しく書いていないが、汎用レジスタにゼロを作り出すには XOR 命令が最も効率が良いと考えられる。しかし、その後、すでにあるゼロ (GR0) を LD 命令で他の汎用レジスタに複写する方が効率が良いのか、あるいは XOR 命令ですべての汎用レジスタをゼロにすれば良いのかは、仕様書からは読み取れないので、この 2 つを解として示す。

(2) 「AND GR3,=1」

解説：最下位ビットに注目する。偶数なら最下位ビットが 0 で、奇数なら 1 である。そこで、1 と論理積 (AND) を取ればよい。

(3) 「XOR GR4,=1」

解説：1 と排他的論理和を取ると問題文の処理になる。

演習問題 7-2-1 (154 ページ)

(1)

1	LD	GR1,GR0
2	LD	GR2,GR0
3	LD	GR3,GR0
4	LD	GR4,GR0
5	LD	GR5,GR0
6	LD	GR6,GR0
7	LD	GR7,GR0

(2) 「LAD GR5,0,GR3」によって生成される実効アドレスは「0+GR3 の値」であるから、GR3 の値が実効アドレスになり、この値がそのまま GR5 に複写される。

「LD GR5,0,GR3」の場合、生成される実効アドレスは同じく「0+GR3 の値」である。LD 命令の場合には、実効アドレス (すなわち、GR3 の値) を番地として持つ主記憶装置の語の値が GR5 に格納される。

(3) この問題は、演習問題 7-1-1(2) と同じ問題である。プログラム 1 は、語長が 4 語になり、主記憶装置とのアクセスが 2 回になる。プログラム 2 は、語長が 3 語になり、主記憶装置とのアクセスが 1 回になる。

演習問題 7-2-2 (156 ページ)

(1)

- (a) LAD GR1, -1, GR1
- (b) LAD GR2, 12, GR2
- (c) LAD GR3, #00AB, GR3
- (d) LAD GR4, #FD03, GR4 (02FD₍₁₆₎ の 2 の補数は FD03₍₁₆₎)

(2)

- (a) 指標レジスタに GR0 を用いている。
- (b) LAD 命令はフラグレジスタを変化させないので、JZE 命令が正しく働かない。

(3)

1	LAD	GR1, -1, GR1
2	LAD	GR2, -1, GR2
3	LAD	GR3, -1, GR3
4	LAD	GR4, -1, GR4
5	LAD	GR5, -1, GR5
6	LAD	GR6, -1, GR6
7	LAD	GR7, -1, GR7

演習問題 7-2-3 (158 ページ)

(1)

処理内容	表 7.6(1 つ目の方法)	表 7.6(2 つ目の方法)	表 7.6(3 つ目の方法)	表 7.6(4 つ目の方法)
(a) GR3 GR2+5	LAD GR3, 5, GR2	LAD GR1, 5 LD GR3, GR2 ADDA GR3, GR1	LD GR3, GR2 ADDA GR3, FIVE FIVE DC 5	LD GR3, GR2 ADDA GR3, =5
(b) GR4 GR1-5	LAD GR4, -5, GR1	LAD GR2, 5 LD GR4, GR1 SUBA GR4, GR2	LD GR4, GR1 SUBA GR4, FIVE FIVE DC 5	LD GR4, GR1 SUBA GR4, =5
(b) GR4 GR1-5 (別解)		LAD GR2, -5 LD GR4, GR1 ADDA GR4, GR2	LD GR4, GR1 ADDA GR4, MFIVE MFIVE DC -5	LD GR4, GR1 ADDA GR4, =-5
(c) GR7 GR6-1	LAD GR7, -1, GR6	LAD GR1, 1 LD GR7, GR6 SUBA GR7, GR1	LD GR7, GR6 SUBA GR7, ONE ONE DC 1	LD GR7, GR6 SUBA GR7, =1
(c) GR7 GR6-1 (別解)		LAD GR1, -1 LD GR7, GR6 ADDA GR7, GR1	LD GR7, GR6 ADDA GR7, MONE MONE DC 1	LD GR7, GR6 ADDA GR7, =-1

なお、「表 7.6(157 ページ)(2 つ目の方法)」では、他の汎用レジスタ ((a) では GR1、(b) では GR2) の値を破壊しているのを、それを保存しておきたい場合には、次のようにスタックに退避する (7.3.2 項 (160 ページ) 参照)。

PUSH	0, GR1
LAD	GR1, 5
LD	GR3, GR2
ADDA	GR3, GR1
POP	GR1

(2)

GR0 が指標レジスタとして使えないことに注意してプログラムを組む。したがって、「LAD GR7, 6, GR0」とすることはできない。

LAD GR1, 6	LD GR7, GR0	LD GR7, GR0
LD GR7, GR0	ADDA GR7, SIX	ADDA GR7, =6
ADDA GR7, GR1	SIX DC 6	

(3)

表 7・6 の 2 番目の方法は汎用レジスタを 1 つ余分に使うが、主記憶装置とのアクセスが発生しない。しかし、3 番目と 4 番目の方法は主記憶装置とのアクセスが発生し、その分、実行速度が低下すると考えられる。ループを使う場合、その内部は何千、何万回も同じ処理が繰り返される可能性があるので、少しでも効率の良い方がよい。

(4)

(d) が異なる。(d) を実行すると GR3 には $800A_{(16)}$ が格納される。これは「符号付き」で見ると -32758 となり、「符号無し」で見ると 32778 となる。

(b) は一見すると -1 (カウントダウン) しないように見える。しかし、 65535 は符号無し数値の中で最大のもので、これを 16 進数で表現すると $FFFF_{(16)}$ となる。これを加算するとオーバーフローするがその結果は -1 した値と同じになる。なお、これをチェックするプログラムが解答プログラム集の chap7 フォルダの中の EX723D.casl に格納されている。

演習問題 7-2-4 (158 ページ)

(1)

1	ADDA	GR2,GR0
---	------	---------

(2)

1	ADDA	GR0,GR7
2	ADDA	GR1,GR7
3	ADDA	GR2,GR7
4	ADDA	GR3,GR7
5	ADDA	GR4,GR7
6	ADDA	GR5,GR7
7	ADDA	GR6,GR7

(3)

シフト命令を使う。ビット操作の方が加算回路を使うより効率が良いと考えられる。

1	SLA	GR3,1
---	-----	-------

演習問題 7-3-1 (160 ページ)

(a)

1	PUSH	0,GR1
2	LD	GR1,GR7
3	LD	GR7,GR6
4	LD	GR6,GR1
5	POP	GR1

(b)

1	ST	GR7,TEMP
2	LD	GR7,GR6
3	LD	GR6,TEMP
4		
5	TEMP	DS 1

(c)

1	PUSH	0,GR7
2	PUSH	0,GR6
3	POP	GR7
4	POP	GR6

演習問題 7-3-2 (162 ページ)

(1)

1	PUSH	0,GR1
2	PUSH	0,GR2
3	PUSH	0,GR3
4	PUSH	0,GR4
5	PUSH	0,GR5
6	PUSH	0,GR6
7	PUSH	0,GR7
8	; 何かの処理	
9	POP	GR7
10	POP	GR6
11	POP	GR5
12	POP	GR4
13	POP	GR3
14	POP	GR2
15	POP	GR1

(2)

1	LD	GR7,GR0
2	PUSH	0,GR7
3	; 何かの処理	
4	POP	GR0

(3)

1. PUSH 0, 2. 0,GR0 3. GR0

演習問題 7-3-3 (163 ページ)

1	PUSH	0,GR4
2	PUSH	0,GR4
3	PUSH	0,GR4
4	; 何かの処理	
5	POP	GR1
6	POP	GR2
7	POP	GR3

総合問題 7 (164 ページ)

[演習 7-1] (164 ページ)

$20 = 2^2 + 2^4$, $5 = 1 + 2^2$ の関係からシフト命令を組み合わせれば良い。

1	TE71	START			
2		LD	GR1,A		
3		LD	GR2,GR1		
4		SLA	GR1,2	GR1	4A
5		SLA	GR2,4	GR2	16A
6		ADDA	GR1,GR2	GR1	4A+16A=20A
7		LD	GR3,B		
8		LD	GR4,GR3		
9		SLA	GR4,2	GR4	4B
10		ADDA	GR3,GR4	GR3	B+4B=5B
11		ADDA	GR1,GR3	GR1	20A+5B
12		ST	GR1,C		
13		RET			
14	A	DC	15		
15	B	DC	10		
16	C	DS	1		
17		END			

[演習 7-2] (164 ページ)

以下のプログラムは、TE72.casl に含まれている。プログラムの行番号は、TE72.casl のものである。

(方法 1)

```

10 ; 方法 1
11 ADDA GR0,=1
12 LAD GR1,1,GR1
13 LAD GR2,1,GR2
14 LAD GR3,1,GR3
15 LAD GR4,1,GR4
16 LAD GR5,1,GR5
17 LAD GR6,1,GR6
18 LAD GR7,1,GR7

```

(方法 2)

```

19 ; 方法 2
20 ADDA GR0,=1
21 ST GR0,TEMP1
22 LAD GR0,1
23 ADDA GR1,GR0
24 ADDA GR2,GR0
25 ADDA GR3,GR0
26 ADDA GR4,GR0
27 ADDA GR5,GR0
28 ADDA GR6,GR0
29 ADDA GR7,GR0
30 LD GR0,TEMP1
44 ;
45 TEMP1 DS 1

```

(方法 3)

```

31 ; 方法 3
32 ADDA GR0,TEMP2
33 ST GR0,TEMP2
34 LAD GR0,1
35 ADDA GR1,GR0
36 ADDA GR2,GR0
37 ADDA GR3,GR0
38 ADDA GR4,GR0
39 ADDA GR5,GR0
40 ADDA GR6,GR0
41 ADDA GR7,GR0
42 LD GR0,TEMP2
44 ;
46 TEMP2 DC 1

```

方法 3 では TEMP に最初 1 を入れておき、その値が不要になった時点で汎用レジスタ GR0 の退避用に用いている。なお、これを確認するプログラムが前述のように TE72.casl に格納されている。参考のために各方法の性能を以下の表に示す。

方法 1 と方法 2 の比較

	プログラムの語数	主記憶装置とのアクセス
方法 1	16 語 (命令)+1 語 (データ)	1 回
方法 2	15 語 (命令)+2 語 (データ)	3 回
方法 3	15 語 (命令)+1 語 (データ)	3 回

参考のために、TE72.casl のプログラム全体を示す。

1	TE72	START	
2		XOR	GRO,GRO
3		XOR	GR1,GR1
4		XOR	GR2,GR2
5		XOR	GR3,GR3
6		XOR	GR4,GR4
7		XOR	GR5,GR5
8		XOR	GR6,GR6
9		XOR	GR7,GR7
10	; 方法 1		
11		ADDA	GRO,=1
12		LAD	GR1,1,GR1
13		LAD	GR2,1,GR2
14		LAD	GR3,1,GR3
15		LAD	GR4,1,GR4
16		LAD	GR5,1,GR5
17		LAD	GR6,1,GR6
18		LAD	GR7,1,GR7
19	; 方法 2		
20		ADDA	GRO,=1
21		ST	GRO,TEMP1
22		LAD	GRO,1
23		ADDA	GR1,GRO
24		ADDA	GR2,GRO
25		ADDA	GR3,GRO
26		ADDA	GR4,GRO
27		ADDA	GR5,GRO
28		ADDA	GR6,GRO
29		ADDA	GR7,GRO
30		LD	GRO,TEMP1
31	; 方法 3		
32		ADDA	GRO,TEMP2
33		ST	GRO,TEMP2
34		LAD	GRO,1
35		ADDA	GR1,GRO
36		ADDA	GR2,GRO
37		ADDA	GR3,GRO
38		ADDA	GR4,GRO
39		ADDA	GR5,GRO
40		ADDA	GR6,GRO
41		ADDA	GR7,GRO
42		LD	GRO,TEMP2
43		RET	
44			
45	TEMP1	DS	1
46	TEMP2	DC	1
47		END	

[演習 7-3] (164 ページ)

11	PUSH	0,GR7
12	ADDA	GR7,GR6
13	ADDA	GR7,GR5
14	ADDA	GR7,GR4
15	ADDA	GR7,GR3
16	ADDA	GR7,GR2
17	ADDA	GR7,GR1
18	ADDA	GR7,GR0
19	ST	GR7,SUM
20	POP	GR7

これを確認するプログラムは、TE73.casl に格納されている。以下に、TE73.casl の全体のプログラムを示す。

1	TE73	START	
2		LAD	GR0,1
3		LAD	GR1,2
4		LAD	GR2,3
5		LAD	GR3,4
6		LAD	GR4,5
7		LAD	GR5,6
8		LAD	GR6,7
9		LAD	GR7,8
10		;	
11		PUSH	0,GR7
12		ADDA	GR7,GR6
13		ADDA	GR7,GR5
14		ADDA	GR7,GR4
15		ADDA	GR7,GR3
16		ADDA	GR7,GR2
17		ADDA	GR7,GR1
18		ADDA	GR7,GR0
19		ST	GR7,SUM
20		POP	GR7
21		RET	
22	SUM	DS	1
23		END	

演習問題 8-1 (171 ページ)

- 1 GR5
- 2 JNZ
- 3 SKIP
- 4 JZE LB2
- 5 JZE
- 6 JZE
- 7 JZE
- 8 JZE
- 9 JPL
- 10 JZE
- 11 JUMP (JMI でもよいが多少効率が落ちる。無条件 JUMP の方が高速であると考えられる)
- 12 JZE
- 13 JUMP

演習問題 8-2-1 (176 ページ)

(1)

1	ABS3	START	
2		LD	GR1,ORGDATA
3		JMI	GETABS
4	SVPROC	ST	GR1,ANSWER
5		RET	
6	GETABS	XOR	GR1,#FFFF
7		ADDA	GR1,=1
8		JUMP	SVPROC
9	ORGDATA	DC	-20
10	ANSWER	DS	1
11		END	

(2)

(a) 「if A > 0 then N ←A else N ←A」とすればよい。

1	EX821A	START	
2		LD	GR1,A
3		JPL	ABSPROC
4	SAVE	ST	GR1,N
5		RET	
6	ABSPROC	XOR	GR1,#FFFF
7		ADDA	GR1,=1
8		JUMP	SAVE
9	A	DC	16
10	N	DS	1
11		END	

(b)

1	EX821B	START	
2		LD	GR1,A
3		ADDA	GR1,B
4		SUBA	GR1,C
5		JPL	ABSPROC
6	SAVE	ST	GR1,N
7		RET	
8	ABSPROC	XOR	GR1,#FFFF
9		ADDA	GR1,=1
10		JUMP	SAVE
11	A	DC	16
12	B	DC	4
13	C	DC	19
14	N	DS	1
15		END	

演習問題 8-2-2 (178 ページ)

- (1) 「CPA GR1,601」の実効アドレスは「60」であるから、これはGR1の値と60番地に格納された値とを比較し、フラグレジスタを変更するという命令になる。60という値と比較したいのならリテラルを用いて「CPA GR1,=60」とすべきである。
- (2) 'A'の文字コードは、表 4-4(66 ページ) から $41_{(16)}=65$ であることがわかる。したがって、GR2 には $65=41_{(16)}$ が格納される。
- (3) LAD 命令の一般形は、「LAD r,adr[x]」である。adr に記述できるのは、ラベル、10進定数、16進定数、リテラルだけである。したがって、文字定数は記述できないので CASL II の構文エラーになる。

総合問題 8 (180 ページ)

[演習 8-1] (180 ページ)

1	TE81	START	
2		LD	GR1,MOJI
3		CPA	GR1,=32
4		JMI	CONTROL
5		JZE	SPACE
6		CPA	GR1,=48
7		JMI	SEPARATR
8		CPA	GR1,=58
9		JMI	NUMBER
10		CPA	GR1,=65
11		JMI	SEPARATR
12		CPA	GR1,=91
13		JMI	UPPER
14		CPA	GR1,=97
15		JMI	SEPARATR
16		CPA	GR1,=123
17		JMI	LOWER
18	SEPARATR	OUT	= 'Separator',=9
19		JUMP	FINISH
20	CONTROL	OUT	= 'Control Key',=11
21		JUMP	FINISH
22	SPACE	OUT	= 'Space',=5
23		JUMP	FINISH
24	NUMBER	OUT	= 'Number',=6
25		JUMP	FINISH
26	UPPER	OUT	= 'Upper',=5
27		JUMP	FINISH
28	LOWER	OUT	= 'Lower',=5
29	FINISH	RET	
30	MOJI	DC	'z'
31		END	

[演習 8-2] (180 ページ)

1	TE82	START	
2		LD	GR1,MOJI
3		CPA	GR1,=65
4		JMI	SET
5		CPA	GR1,=91
6		JMI	LOWER
7	SET	ST	GR1,MOJI
8		RET	
9	LOWER	XOR	GR1,=#0020
10		JUMP	SET
11	MOJI	DC	'Z'
12		END	

[演習 8-3] (180 ページ)

1	TE83	START	
2		LD	GR1,A
3		ADDL	GR1,B
4		JOV	OVER
5		XOR	GRO,GRO
6	SET	ST	GRO,OVF
7		RET	
8	OVER	LAD	GRO,1
9		JUMP	SET
10	A	DC	#8000
11	B	DC	#7000
12	OVF	DS	1
13		END	

[演習 8-4] (180 ページ)

次の手順で作ればよいでしょう。

- (1) GR1 YEAR
- (2) GR1 GR1-400
- (3) GR1 が 0 なら (16) へ
- (4) GR1 が正なら (2) へ
- (5) GR1 YEAR
- (6) GR1 GR1-100
- (7) GR1 が 0 なら (13) へ
- (8) GR1 が正なら (6) へ
- (9) GR1 YEAR
- (10) GR1 GR1-4
- (11) GR1 が 0 なら (16) へ
- (12) GR1 が正なら (10) へ
- (13) GR2 0
- (14) LYEAR GR2
- (15) (18) へジャンプ
- (16) GR2 1
- (17) LYEAR GR2
- (18) OS へ戻る

上記のロジックをプログラムにすると次のようになります。

1	LEAP	START	
2		LD	GR1, YEAR
3	LP1	SUBA	GR1, =400
4		JZE	LEAPYEAR
5		JPL	LP1
6		LD	GR1, YEAR
7	LP2	SUBA	GR1, =100
8		JZE	NOTL
9		JPL	LP2
10		LD	GR1, YEAR
11	LP3	SUBA	GR1, =4
12		JZE	LEAPYEAR
13		JPL	LP3
14	NOTL	XOR	GR2, GR2
15		ST	GR2, LYEAR
16		JUMP	FINISH
17	LEAPYEAR	LD	GR2, =1
18		ST	GR2, LYEAR
19	FINISH	RET	
20	YEAR	DC	2104
21	LYEAR	DS	1
22		END	

[演習 8-5] (180 ページ)

1	TR	START	
2		LD	GR1, A
3		LD	GR2, B
4		LD	GR3, C
5		LD	GR4, GR2
6		ADDA	GR4, GR3
7		CPA	GR1, GR4
8		JMI	TR1
9		JUMP	NOTTR
10	TR1	LD	GR4, GR3
11		ADDA	GR4, GR1
12		CPA	GR2, GR4
13		JMI	TR2
14		JUMP	NOTTR
15	TR2	LD	GR4, GR1
16		ADDA	GR4, GR2
17		CPA	GR3, GR4
18		JMI	TRIA
19	NOTTR	LD	GR4, =0
20		JUMP	FIN
21	TRIA	LD	GR4, =1
22	FIN	ST	GR4, TRI
23		RET	
24	A	DC	3
25	B	DC	4
26	C	DC	5
27	TRI	DS	1
28		END	

演習問題 9-2-1 (189 ページ)

(1)

- 1 2
- 2 -1
- 3 GR4
- 4 JMI (これを確認するプログラムが EX921A.casli に格納されている)

なお、解答 3 は「=9」も可であるが効率が落ちる。

(2)

1	LOOP1A	START		
2		XOR	GR1,GR1	GR1 0,
3	LOOP	ADDA	GR1,=1	GR1(カウンタ)をカウントアップ
4		NOP		;
5		CPA	GR1,=5	GR1:GR4(=5) 比較
6		JNZ	LOOP	if
7		RET		
8		END		

1	LOOP3A	START		
2		LAD	GR1,1	GR1 1
3	LOOP	NOP		;
4		ADDA	GR1,=1	GR1(カウンタ)をカウントアップ
5		CPA	GR1,=6	GR1:6
6		JNZ	LOOP	if
7		RET		
8		END		

(3)

1	LOOP3B	START		
2		LAD	GR1,1	GR1 1
3		LAD	GR2,1	GR2 1
4		LAD	GR3,6	GR3 6
5	LOOP	NOP		;
6		ADDA	GR1,GR2	GR1(カウンタ)をカウントアップ
7		CPA	GR1,GR3	GR1:6
8		JMI	LOOP	if
9		RET		
10		END		

演習問題 9-2-2 (191 ページ)

1. GR1
2. JZE

演習問題 9-2-3 (193 ページ)

LAD 命令では、フラグレジスタの値を変更しないので、JNZ 命令が有効に働かない。ゆえに「LAD GR1,-1,GR1」としたのではだめである。

演習問題 9-3-1(a) (194 ページ)

(1)

汎用レジスタの値をゼロにするには XOR 命令が最も効率が良いと考えられるが、この他に次の命令でも汎用レジスタをゼロクリアできる。表 7-3(152 ページ) に詳細に記載されている。

方法 1	LAD	GR1,0
方法 2	SUBA	GR1,GR1
方法 3	LD	GR1,=0

(2)

以下に示すように SUM1A の方がプログラムの行数は短い、ループ内での主記憶装置へのアクセスが多くなるので、実行効率が悪くなる。

1	SUM1A	START		
2		XOR	GR1,GR1	GR1 0,
3		XOR	GR3,GR3	GR3 0(合計値)
4	LOOP	ADDA	GR1,=1	GR1(カウンタ)をカウントアップ
5		ADDA	GR3,GR1	GR3
6		CPA	GR1,N	GR1(汎用レジスタ)
7		JMI	LOOP	if
8		ST	GR3,S	GR3 S
9		RET		
10	S	DS	1	
11	N	DC	10	
12		END		

解説 「ADDA GR1,GR4」は汎用レジスタ同士の計算で処理速度が速いし、命令語長も 1 語である。これに対し、「ADDA GR1,=1」は、値 1 を主記憶装置から取り出す分だけ処理速度が遅くなり、命令語長も 2 語になる。CPA 命令についても同じ事が言える。SUM1A では、ループの準備の部分で、2 つのロード命令が無くなっているが、そのことよりも何度も繰り返されるループの本体の命令語長や主記憶装置とのアクセス回数が増えていて、トータルの効率がかかなり落ちている。

(3)

1	SUM1B	START		
2		LAD	GR1,1	GR1 1,
3		LAD	GR2,1	GR2 1(定数)
4		LD	GR4,N	GR4 N,
5		ADDA	GR4,GR2	GR4 N+1(「リミット+1」にする)
6		XOR	GR3,GR3	GR3 0(合計値)
7	LOOP	ADDA	GR3,GR1	GR3
8		ADDA	GR1,GR2	GR1(カウンタ)をカウントアップ
9		CPA	GR1,GR4	GR1:(N+1)
10		JMI	LOOP	if
11		ST	GR3,S	GR3 S
12		RET		
13	S	DS	1	
14	N	DC	10	
15		END		

1	SUM1C	START		
2		XOR	GR1,GR1	GR1 0,
3		LD	GR4,N	GR4 N,
4		LAD	GR2,1	GR2 1(定数)
5		XOR	GR3,GR3	GR3 0(合計値)
6	LOOP	CPA	GR1,GR4	
7		JPL	EXITLOOP	
8		ADDA	GR3,GR1	GR3
9		ADDA	GR1,GR2	GR1(カウンタ)をカウントアップ
10		JUMP	LOOP	
11	EXITLOOP	ST	GR3,S	
12		RET		
13	S	DS	1	
14	N	DC	10	
15		END		

演習問題 9-3-1(b) (195 ページ)

N が奇数かどうか判断するには、 $0001_{(16)}$ と論理積 (AND) を取って、最下位ビットを抽出し、それが 1 であれば奇数、0 であれば偶数と判断できる。最初の命令で N の値は汎用レジスタ GR1 へロードされているが、この値と $0001_{(16)}$ の論理積を取ると GR1 の値が N の値ではなくなってしまふ。そこで、GR1 の値をいったん GR2 にロードして、GR2 の値と $0001_{(16)}$ の値の論理積を取る。その結果、1 となれば奇数であるから、これを判定する命令は JPL あるいは JNZ ということになる。もし、N の値が奇数ならラベル ODDVALUE へジャンプする。ここでは、汎用レジスタ GR3 の値をゼロクリアして、ラベル SETV へジャンプする。

1. GR2
2. JPL あるいは JNZ
3. GR3
4. JUMP SETV

演習問題 9-3-2 (196 ページ)

以下のプログラムを実行すると N には 255 が格納される。1 ~ 255 までの和が 32640 であり、1 ~ 256 までの和が 32896 であるので、N=255 が正しいことが分かる。

1	SUM4A	START		
2		XOR	GR1,GR1	GR1 0,
3		LAD	GR2,1	GR2 1(定数)
4		XOR	GR3,GR3	GR3 0(合計値)
5	LOOP	ADDA	GR1,GR2	GR1(カウンタ)をカウントアップ
6		ADDA	GR3,GR1	GR3
7		JOV	EXITLOOP	オーバーフローならループを終了
8		JUMP	LOOP	LOOP ヘジャンプ
9	EXITLOOP	SUBA	GR1,GR2	
10		ST	GR1,N	GR3 N
11		RET		
12	N	DS	1	
13		END		

演習問題 9-4-1 (197 ページ)

(1)

1	MULTA	START		
2		LD	GR2,B	GR2 B、B 回ループするカウンタ
3		LD	GR3,A	GR3 A
4		CPA	GR2,GR3	B:A
5		JMI	SKIP	B<A なら SKIP ヘジャンプ
6		LD	GR1,GR2	B と A を
7		LD	GR2,GR3	格納しているレジスタの値を
8		LD	GR3,GR1	交換する
9	SKIP	XOR	GR1,GR1	GR1 0、解を格納する汎用レジスタ
10		LD	GR2,GR2	GR2(ループ回数)とゼロを比較
11		JZE	FINISH	ループ回数=0 なら答えもゼロ
12		LAD	GR4,1	GR4 1、定数、カウントダウン用
13	MLTLOOP	ADDA	GR1,GR3	GR1 GR1+GR3、積を求める
14		SUBA	GR2,GR4	カウンタのカウントダウン
15		JPL	MLTLOOP	カウンタが 1 以上なら MLTLOOP ヘ
16	FINISH	ST	GR1,C	GR1(解) C
17		RET		
18	A	DC	0	
19	B	DC	148	
20	C	DS	1	
21		END		

プログラム MULTA では、B A のとき、A、B の値が入った汎用レジスタの値を交換しているが、次のように、再ロードする方法も考えられる (MULTA1.casl)。

1	MULTA1	START		
2		LD	GR2,B	GR2 B、B 回ループするカウンタ
3		LD	GR3,A	GR3 A
4		CPA	GR2,GR3	B:A
5		JMI	SKIP	B<A なら SKIP ヘジャンプ
6		LD	GR2,A	B と A を
7		LD	GR3,B	格納しているレジスタの値を交換する
8	SKIP	XOR	GR1,GR1	GR1 0、解を格納する汎用レジスタ
9		LD	GR2,GR2	GR2(ループ回数)とゼロを比較
10		JZE	FINISH	ループ回数=0 なら答えもゼロ
11		LAD	GR4,1	GR4 1、定数、カウントダウン用
12	MLTLOOP	ADDA	GR1,GR3	GR1 GR1+GR3、積を求める
13		SUBA	GR2,GR4	カウンタのカウントダウン
14		JPL	MLTLOOP	カウンタが 1 以上なら MLTLOOP ヘ
15	FINISH	ST	GR1,C	GR1(解) C
16		RET		
17	A	DC	3	
18	B	DC	148	
19	C	DS	1	
20		END		

(2)

1	MULTB	START		
2		XOR	GR5,GR5	GR5 0(オーバーフローした印)
3		LD	GR2,B	GR2 B、B 回ループするカウンタ
4		LD	GR3,A	GR3 A
5		CPA	GR2,GR3	B:A
6		JMI	SKIP	B<A なら SKIP ヘジャンプ
7		LD	GR1,GR2	B と A を
8		LD	GR2,GR3	格納しているレジスタの値を
9		LD	GR3,GR1	交換する
10	SKIP	XOR	GR1,GR1	GR1 0、解を格納する汎用レジスタ
11		LD	GR2,GR2	GR2(ループ回数)とゼロを比較
12		JZE	FINISH	ループ回数=0 なら答えもゼロ
13		LAD	GR4,1	GR4 1、定数、カウントダウン用
14	MLTLOOP	ADDA	GR1,GR3	GR1 GR1+GR3、積を求める
15		JOV	OVERFLOW	
16		SUBA	GR2,GR4	カウンタのカウントダウン
17		JPL	MLTLOOP	カウンタが 1 以上なら MLTLOOP ヘ
18	FINISH	ST	GR1,C	GR1(解) C
19		ST	GR5,OVF	GR5(オーバーフローの印) OVF
20		RET		
21	OVERFLOW	LAD	GR1,#7FFF	GR1 正の最大値
22		LAD	GR5,1	GR5 1(オーバーフローした印)
23		JUMP	FINISH	
24	A	DC	100	
25	B	DC	200	
26	C	DS	1	
27	OVF	DS	1	
28		END		

演習問題 9-4-2 (199 ページ)

(1)

プログラム CNVTIME の最初で GR3 を 0 に初期化している。GR3 には、分を求めたときの余りが格納されるので初期化の必要性がないと感じるかも知れない。しかし、SECOND が 0 のときには、「JNZ FINISH」命令で FINISH ヘジャンプする。そのときに GR3 が 0 になっていないと都合が悪いのである。

1	;	秒数 (SECOND) を時 (HOUR)、分 (MINUTE)、秒 (SECOND) に変換するプログラム
2	;	
3	CNVTIME	START
4	XOR	GR1, GR1 時
5	XOR	GR2, GR2 分
6	XOR	GR3, GR3 秒
7	LD	GR4, TIME (ここからどんどん引いていく)
8	JZE	FINISH もし GR3=0 なら FINISH ヘジャンプ
9	LAD	GR5, 1 GR5 1
10	;	時間を求める
11	LAD	GR6, 3600 1 時間は 3600 秒
12	HLOOP	CPA GR4, GR6 GR4 と GR6 (=3600) を比較
13	JMI	MSTEP GR4 から GR6 (=3600) が引けなくなったら MSTEP
14	SUBA	GR4, GR6 GR4 GR4+GR6 (=3600)
15	ADDA	GR1, GR5 GR1 (時間) GR1+GR5 (=1)
16	JUMP	HLOOP HLOOP ヘジャンプしてループを形成
17	;	分を求める
18	MSTEP	LAD GR6, 60 1 分間は 60 秒
19	MLOOP	CPA GR4, GR6 GR4 と GR6 (=60) を比較
20	JMI	SSTEP GR4 から GR6 (=60) が引けなくなったら SSTEP
21	SUBA	GR4, GR6 GR4 GR4+GR6 (=3600)
22	ADDA	GR2, GR5 GR2 (分) GR2+GR5 (=1)
23	JUMP	MLOOP MLOOP ヘジャンプしてループを形成
24	;	秒を求める
25	SSTEP	LD GR3, GR4 余り (GR4) を秒 (GR3) にセット
26	FINISH	ST GR1, HOUR GR1 HOUR
27		ST GR2, MINUTE GR2 MINUTE
28		ST GR3, SECOND GR3 SECOND
29	RET	;
30	TIME	DC 12530
31	HOUR	DS 1
32	MINUTE	DS 1
33	SECOND	DS 1
34	END	

(2)

1	DIV5	START		
2		LAD	GR3,5	GR3 5
3		XOR	GR1,GR1	GR1(商) 0
4		LD	GR2,A	GR2 A(ここからどんどん引いていく)
5		LAD	GR4,1	GR4 1
6	DIVLOOP	CPA	GR2,GR3	GR2 と GR3(=B) を比較
7		JMI	FINISH	GR2 から GR3(=B) が引けなくなったら FINISH
8		SUBA	GR2,GR3	GR2 GR2+GR3(=B)
9		ADDA	GR1,GR4	GR1 GR1+GR4(=1)
10		JUMP	DIVLOOP	DIVLOOP ヘジャンプしてループを形成
11	FINISH	ST	GR1,C	GR1 C
12		SLA	GR2,1	余りを 2 倍にして小数点以下第 1 桁を求める
13		ST	GR2,P	GR2 P
14		RET		;
15	A	DC	1253	
16	C	DS	1	商(整数部)
17	P	DS	1	商(小数点以下第 1 桁)
18		END		

総合問題 9 (199 ページ)

[演習 9-1] (199 ページ)

1	; フィボナッチ数列の第 20 項目を求める			
2	FIB20	START		
3		LAD	GR0,0	F0 0
4		LAD	GR1,1	F1 1
5		LAD	GR3,1	GR3:カウンタ
6		LAD	GR4,1	定数
7		LAD	GR5,20	定数
8	LOOP	ADDA	GR3,GR4	GR3 カウントアップ
9		LD	GR2,GR0	
10		ADDA	GR2,GR1	GR2 GR0+GR1
11		CPA	GR3,GR5	
12		JZE	EXITLOOP	
13		LD	GR0,GR1	
14		LD	GR1,GR2	
15		JUMP	LOOP	
16	EXITLOOP	ST	GR2,F20	
17		RET		
18	F20	DS	1	
19		END		

[演習 9-2] (199 ページ)

1	LPCHK	START		
2		LD	GR1,N	
3		LAD	GR2,1	定数
4	LOOP	CPA	GR1,GR2	
5		JZE	FINISH	もし GR1=1 ならおしまい
6		LD	GR3,GR1	GR1 の奇偶判断
7		AND	GR3,GR2	(直接 GR1 と AND を取ると GR1 の値が破壊される)
8		JZE	EVEN	
9	; 奇数の時の処理			
10		LD	GR3,GR1	
11		ADDA	GR1,GR3	
12		ADDA	GR1,GR3	
13		ADDA	GR1,GR2	GR1 3GR1+1
14		JUMP	LOOP	
15	; 偶数の時の処理			
16	EVEN	SRA	GR1,1	GR1 GR1 ÷ 2
17		JUMP	LOOP	
18	FINISH	RET		
19	N	DC	6	
20		END		

[演習 9-3] (199 ページ)

IN 命令では、データが無くなると、-1 の長さを返す。

以下のプログラムは、次々にデータを読み込み、長さが負になったら終了している。

1	FILEPR	START	
2	LOOP	IN	TEXT,LEN
3		LD	GR1,LEN
4		JMI	EXIT
5		OUT	TEXT,LEN
6		JUMP	LOOP
7	EXIT	RET	
8	TEXT	DS	255
9	LEN	DS	1
10		END	

[演習 9-4] (200 ページ)

問題文に示されたゼラーの公式通り丁寧にプログラムを組んでいけばよい。
参考のために、いくつかのデータを示す。

例 1 : 2011 年 12 月 31 日の例

$Y=2011$ 、 $M=12$ なので、 $M \geq 3$ に該当し、 $YY=Y=2011$ 、 $MM=M-2=12-2=10$ となる。

$$\begin{aligned} X &= 2011 + \left\lfloor \frac{2011}{4} \right\rfloor - \left\lfloor \frac{2011}{100} \right\rfloor + \left\lfloor \frac{2011}{400} \right\rfloor + \left\lfloor \frac{13 \times 10 - 1}{5} \right\rfloor + 31 \\ &= 2011 + 502 - 20 + 5 + 25 + 31 = 2554 \\ 2554 &\text{ を } 7 \text{ で割った余りは、} 6 \text{ であるので「土曜日」である。} \end{aligned}$$

例 1 : 2012 年 2 月 2 日の例

$Y=2012$ 、 $M=2$ なので、 $M < 3$ に該当し、 $YY=Y-1=2011$ 、 $MM=M+2=2+2=4$ となる。

$$\begin{aligned} X &= 2011 + \left\lfloor \frac{2011}{4} \right\rfloor - \left\lfloor \frac{2011}{100} \right\rfloor + \left\lfloor \frac{2011}{400} \right\rfloor + \left\lfloor \frac{13 \times 4 - 1}{5} \right\rfloor + 2 \\ &= 2011 + 502 - 20 + 5 + 10 + 2 = 2510 \\ 2510 &\text{ を } 7 \text{ で割った余りは、} 4 \text{ であるので「木曜日」である。} \end{aligned}$$

例 1 : 2012 年 6 月 25 日の例

$Y=2012$ 、 $M=6$ なので、 $M \geq 3$ に該当し、 $YY=Y=2012$ 、 $MM=M-2=6-2=4$ となる。

$$\begin{aligned} X &= 2012 + \left\lfloor \frac{2012}{4} \right\rfloor - \left\lfloor \frac{2012}{100} \right\rfloor + \left\lfloor \frac{2012}{400} \right\rfloor + \left\lfloor \frac{13 \times 4 - 1}{5} \right\rfloor + 25 \\ &= 2012 + 503 - 20 + 5 + 10 + 25 = 2535 \\ 2535 &\text{ を } 7 \text{ で割った余りは、} 1 \text{ であるので「月曜日」である。} \end{aligned}$$

次ページにプログラムを示す。

```

1 LEAPYAER      START
2              LD      GRO,Y      YY  Y
3              LD      GR1,M      MM  M
4              CPA     GR1,=3
5              JMI     LOWER
6 ; M 3 のとき
7              SUBA    GR1,=2      MM  MM-2
8 MAIN01        LD      GR3,GRO    X(=GR3)  0
9 ; YY/4 を加える
10             XOR     GR4,GR4     GR4  0(解=YY/4)
11             LD      GR5,GRO     GR5  YY(これを 4 で割る)
12 LP4          ADDA    GR4,=1
13             SUBA    GR5,=4
14             JZE     NEXT100
15             JPL     LP4
16             SUBA    GR4,=1      解を足しすぎたので+1 する
17 NEXT100      ADDA    GR3,GR4    X  X+YY/4
18 ; YY/100 を引く
19             XOR     GR4,GR4     GR4  0(解=YY/100)
20             LD      GR5,GRO     GR5  YY
21 LP100        ADDA    GR4,=1
22             SUBA    GR5,=100
23             JZE     NEXT400
24             JPL     LP100
25             SUBA    GR4,=1      解を足しすぎたので+1 する
26 NEXT400      SUBA    GR3,GR4
27 ; YY/400 を加える
28             XOR     GR4,GR4
29             LD      GR5,GRO     GR5  YY
30 LP400        ADDA    GR4,=1
31             SUBA    GR5,=400
32             JZE     NEXTSP
33             JPL     LP400
34             SUBA    GR4,=1
35 NEXTSP       ADDA    GR3,GR4
36 ; (13 × MM-1)/5 の計算
37 ; 13 倍は、8+4+1 で計算
38             LD      GR5,GR1     GR5  MM
39             LD      GR6,GR1     GR6  MM
40             SLA     GR5,3      GR5  MM の 8 乗
41             SLA     GR6,2      GR6  MM の 4 乗
42             ADDA    GR5,GR6     GR5  MM の 8 乗+MM の 4 乗=MM の 12 乗
43             ADDA    GR5,GR1     GR5  MM の 12 乗+MM=MM の 13 乗
44             SUBA    GR5,=1      GR5  GR5-1
45             XOR     GR4,GR4
46 LPLAST       ADDA    GR4,=1
47             SUBA    GR5,=5
48             JZE     LASTSP
49             JPL     LPLAST
50             SUBA    GR4,=1
51 LASTSP       ADDA    GR3,GR4
52             ADDA    GR3,D      X に D を加える
53 ; GR3(=X) を 7 で割った時の余りを計算
54 CONT        SUBA    GR3,=7
55             JZE     FIN
56             JPL     CONT
57             ADDA    GR3,=7
58 FIN         ST      GR3,WEEK
59             RET
60 ; M=1,2 のとき
61 LOWER       SUBA    GRO,=1      YY  YY-1
62             ADDA    GR1,=2      MM  MM+1
63             JUMP    MAIN01
64 Y           DC      2012
65 M           DC      6
66 D           DC      25
67 WEEK        DS      1
68             END

```

[演習 9-5] (200 ページ)

第 12 章で学習するサブルーチンを使えばもっとコンパクトにできるが、まだ学習していないので、ここでは加算による方法を使っている。ただし、 2×3 の計算は 3 を 2 回加えれば計算可能であるが、この解答プログラムでは汎用性を示し、ループによる計算を行っている。

1	FACT	START	
2		LD	GR4,=1
3	; 2 × 3=6 を求める		
4		LD	GR1,=2
5		LD	GR2,=3
6		XOR	GR3,GR3
7	M2B3	ADDA	GR3,GR2
8		SUBA	GR1,GR4
9		JPL	M2B3
10	; 3! × 4=24		
11		LD	GR1,GR3
12		LD	GR2,=4
13		XOR	GR3,GR3
14	MB4	ADDA	GR3,GR2
15		SUBA	GR1,GR4
16		JPL	MB4
17	; 4! × 5=120		
18		LD	GR1,GR3
19		LD	GR2,=5
20		XOR	GR3,GR3
21	MB5	ADDA	GR3,GR2
22		SUBA	GR1,GR4
23		JPL	MB5
24	; 5! × 6=720		
25		LD	GR1,GR3
26		LD	GR2,=6
27		XOR	GR3,GR3
28	MB6	ADDA	GR3,GR2
29		SUBA	GR1,GR4
30		JPL	MB6
31	; 6! × 7=5040		
32		LD	GR1,GR3
33		LD	GR2,=7
34		XOR	GR3,GR3
35	MB7	ADDA	GR3,GR2
36		SUBA	GR1,GR4
37		JPL	MB7
38	; 解を格納		
39		ST	GR3,ANS
40		RET	
41	ANS	DS	1
42		END	

[演習 9-6] (200 ページ)

100 までの素数は以下のように全部で 25 個である。「(10 進数、2 進数)」という形式で示している。

(2, $2_{(2)}$)、(3, $3_{(2)}$)、(5, $5_{(2)}$)、(7, $7_{(2)}$)、(11, $B_{(2)}$)、(13, $D_{(2)}$)、(17, $11_{(2)}$)、(19, $13_{(2)}$)、
 (23, $17_{(2)}$)、(29, $1D_{(2)}$)、(31, $1F_{(2)}$)、(37, $25_{(2)}$)、(41, $29_{(2)}$)、(43, $2B_{(2)}$)、(47, $2F_{(2)}$)、
 (53, $35_{(2)}$)、(59, $3B_{(2)}$)、(61, $3D_{(2)}$)、(67, $43_{(2)}$)、(71, $47_{(2)}$)、(73, $49_{(2)}$)、(79, $4F_{(2)}$)、
 (83, $53_{(2)}$)、(89, $59_{(2)}$)、(97, $61_{(2)}$)

1	PRIME	START
2	; 2 は素数とする	
3	LD	GR0,=2
4	ST	GR0,PN
5	; GR2 を 3 から 100 まで 2 個ずつ動かす	
6	LD	GR1,=1 PN の添字
7	LD	GR2,=1
8	MLOOP	ADDA GR2,=2
9	; GR2 が素数かどうかチェックする	
10	; GR3 を 3 から GR2-1 まで動かす	
11	LD	GR3,=3
12	LOOP	CPA GR2,GR3
13		JZE FOUND
14	; 実際の検査	
15	; GR4(=GR2) を GR3 で割り切れるか	
16	LD	GR4,GR2
17	; GR4 が 0 なら割り切れた (素数ではない)--->GR2 を次にする	
18	; GR4 が負--->割り切れない--->GR3 を次にする	
19	CHECK	CPA GR4,=0
20		JZE MLOOP
21		JMI NEXT3
22		SUBA GR4,GR3
23		JUMP CHECK
24	NEXT3	ADDA GR3,=2
25		JUMP LOOP
26	FOUND	ST GR2,PN,GR1
27		ADDA GR1,=1
28		CPA GR2,=99
29		JMI MLOOP
30		RET
31	PN	DS 25
32		END

[演習 9-7] (200 ページ)

このプログラムは、演習 9-6 の結果を利用すればよい。

100 までの素数の和は、 $1159=487_{(2)}$ となる。

```

1  PSUM      START
2  ; GR6 に合計を加える
3          XOR      GR6,GR6
4  ; 2 は素数とする
5          LD        GR0,=2
6  ; GR2 を 3 から 100 まで 2 個ずつ動かす
7          LD        GR2,=1
8  MLOOP    ADDA     GR2,=2
9  ; GR2 が素数かどうかチェックする
10 ; GR3 を 3 から GR2-1 まで動かす
11          LD        GR3,=3
12 LOOP     CPA      GR2,GR3
13          JZE       FOUND
14 ; 実際の検査
15 ; GR4(=GR2) を GR3 で割り切れるか
16          LD        GR4,GR2
17 ; GR4 が 0 なら割り切れた (素数ではない)--->GR2 を次にする
18 ; GR4 が負--->割り切れない--->GR3 を次にする
19 CHECK    CPA      GR4,=0
20          JZE       MLOOP
21          JMI       NEXT3
22          SUBA      GR4,GR3
23          JUMP      CHECK
24 NEXT3    ADDA      GR3,=2
25          JUMP      LOOP
26 FOUND    ADDA      GR6,GR2
27          CPA      GR2,=99
28          JMI       MLOOP
29          ST        GR6,SUM
30          RET
31 SUM      DS        1
32          END

```

[演習 9-8] (200 ページ)

1000 以下の完全数は、6、28、496 です。なお、このプログラムは CASLDV でトレースを行って実行すると大変時間がかかります。メニューの「オプション」「実行時オプションの指定」をクリックし、「実行時トレースを出力する」のチェックボックスをオフにしてください。この方法でトレースなしで実行できます。

なお、このプログラム PERNUM では、完全数は表 PERNUMBR に格納しています。

1	PERNUM	START	
2	; GR1=3---	>1000	
3		LD	GR0,=1 定数 1
4		LD	GR1,=2
5		XOR	GR5,GR5 PERNUMBR のカウンタ
6	MAINLP	ADDA	GR1,GR0 GR1 GR1+1
7	; GR2=1---	>GR1-1	
8		XOR	GR2,GR2
9		XOR	GR3,GR3 ここに約数の合計を求める
10	CLOOP	ADDA	GR2,GR0 GR2 GR2+1
11		CPA	GR1,GR2
12		JZE	NEXT GR1=GR2 なら GR2 のループ終了
13	; GR1 が GR2 で割りきれ	るならば、GR2 を GR3 に加算	
14		LD	GR4,GR1
15	CHECK	SUBA	GR4,GR2
16		JZE	DIVZERO
17		JPL	CHECK GR4>0 なら CHECK へ
18	; GR4<0 なの	で割り切れない。次の GR2 へ	
19		JUMP	CLOOP
20	; GR1 が GR2 で割り	きれたので、GR3 に GR2 を加算し、次の GR2 へ	
21	DIVZERO	ADDA	GR3,GR2
22		JUMP	CLOOP
23	; GR1=GR3 なら	完全数	
24	NEXT	CPA	GR1,GR3
25		JZE	PERN
26	CONT	CPA	GR1,=1000
27		JMI	MAINLP
28		NOP	
29		RET	
30	PERN	ST	GR1,PERNUMBR,GR5
31		ADDA	GR5,GR0
32		JUMP	CONT
33	PERNUMBR	DS	10
34		END	

演習問題 10-1 (202 ページ)

(1)

1. 3 2. 8 3. GR7 4. 3 5. JZE 6. 4 7. SRA 8. 1
 9. XOR 10. FFFF 11. 00FF 12. XOR 13. F000 14. XOR 15. E000 16. AND
 17. FF7F 18. AND 19. FEFF 20. OR 21. 0008

注:問題番号 1、2、4、6、8 は 16 進表記でも可。

(2) 0,GR3

(3) 3,GR3

演習問題 10-2 (204 ページ)

(1)

1	BCOUNT1	START	CSTART	
2	BITON	ADDA	GR2,GR4	GR2(1 のビット数) をカウントアップ
3		JUMP	CHECK	CHECK に戻ってループ続行
4	CSTART	LD	GR1,A	
5		XOR	GR2,GR2	GR2:1 のビット数
6		LAD	GR3,16	ループ回数(16 回)
7		LAD	GR4,1	1(減算用・加算用の定数)
8	LOOP	SRL	GR1,1	最下位ビットを押し出す
9		JOV	BITON	ビットが立っていれば BITON へ
10	CHECK	SUBA	GR3,GR4	GR3(カウンタ) をカウントダウン
11		JNZ	LOOP	16 回回り切ってなければ LOOP へジャンプ
12		ST	GR2,COUNT	ループ終了、求められたビット数を COUNT へ
13		RET		; プログラムの終了
14	A	DC	2263	
15	COUNT	DS	1	
16		END		

(2) 1. SLL 2. ADDA GR1,GR2 3. JZE

(3) 1. JOV 2. OR (あるいは、ADDL、ADDA でも動作するが、効率は多少落ちると考えられる)

(4) 1. AND 2. 8 3. 1 4. GR4 5. JNZ 6. AND 7. ADDA 8. 0100

演習問題 10-3 (210 ページ)

(1) ~ (3) を確認するプログラムが EX10P3.casl に格納されています。下記の行番号は、その EX10P3.casl 内の番号です。

(1)

3	AND	GR4,=#00F0
4	SRL	GR4,4

(2)

7	AND	GR4,=#1FE0
8	SRL	GR4,5

(3)

11	AND	GR4,=#7FFE
12	SRL	GR4,1

参考のために、EX10P3.casl の全ソースプログラムを示します。

1	EX10P3	START
2		LD GR4,N
3		AND GR4,=#00F0
4		SRL GR4,4
5		ST GR4,ANS1
6		LD GR4,N
7		AND GR4,=#1FE0
8		SRL GR4,5
9		ST GR4,ANS2
10		LD GR4,N
11		AND GR4,=#7FFE
12		SRL GR4,1
13		ST GR4,ANS3
14		RET
15	N	DC #9A56
16	ANS1	DS 1
17	ANS2	DS 1
18	ANS3	DS 1
19		END

(4) 1. XOR 2. GR1 3. JZE 4. GR5 5. SRL 6. 0F00 7. SUBA 8. 12 9. FINAL

プログラム (DECBIN) は、ループを使わずに各桁ごとに計算しているので多少冗長である。そこで、ループを使ってこのプログラムを簡略化したプログラムが、第 11 章テーブル操作に載っているので、参考にしてください。

演習問題 10-4 (213 ページ)

1. JOV 2. SLL GR3,1

演習問題 10-5 (217 ページ)

(1) 1. -1 2. SRL GR0,1 3. JZE 4. ADDA GR5,GR4 5. CPA 6. -1

(2) このような問題は、端の部分の処理で間違える可能性が高いから、次のようなデータの組み合わせが考えられる。

A	#FFFF	#F000	#0007	#7FFF	#FFFE
N	16	4	3	15	15

演習問題 10-6 (220 ページ)

1. SETC 2. XOR 3. JZE 4. SETC

総合問題 10 (222 ページ)

[演習 10-1] (222 ページ)

(1)

特定領域を抜き取るマスクデータを作成し、対象データと論理積 (AND) を取って、そのビット列だけを抽出する。シフト命令を用いて、抜き取られたビット列の一番低いビット番号のビットが最下位ビットになるようにシフトする。特定領域の対象となるビット数だけループを繰り返し、1 ビットずつ右シフトで押し出して、そのオーバーフローした数をカウントすれば、それが対象のビット数になっている。

(2)

「XOR GR2,=1」などとして、GR2 と 1 の排他的論理和を取る。こうすれば、もし GR2=1 ならば GR2 の値は 0 になり、GR2=0 ならば GR2 の値は 1 になる。

(3)

特定領域を含む語が GR1 にあるとする。「XOR GR1,=#FFFF」で 16 ビットすべてのビットを反転させる。対象領域の最下位ビット (問題中の図ならばビット番号 6) だけが 1 のデータを作る。これが GR2 にセットされているとする。「ADDA GR1,GR2」を行う。その後、特定領域を抜き取るマスクデータを作成し、対象データと論理積 (AND) を取って、そのビットだけを抽出する。

(4)

ループを 16 回繰り返す。この中で、対象となる語を 1 ビットずつシフトし、オーバーフローをチェックする。フラグを使って連続した状態が 12 回続いたかどうかチェックする。

(5)

2つの語のビットがすべて一致しているということは、その語が等しいということであるから、このチェックは簡単である。そこで、GR1、GR2、GR3の各汎用レジスタのすべてのビットが等しいかどうかは、次のプログラムで検査できる。

```
CPA    GR1,GR2
JNZ    NO
CPA    GR1,GR3
JNZ    NO
; 3つとも等しい
NO ; 等しくない
```

(6)

GR1に $ABCD_{(16)}$ が格納されていて、これを $BDCA_{(16)}$ にしてGR2に格納することを考える。これには次のステップが必要になる。

- (a) GR2をゼロクリアする。
- (b) GR1をGR3にコピーし、GR3の先頭4ビット(ビット番号(12~15))を抽出する。それを12ビット右に論理シフトしてGR2に加える。
- (c) GR1をGR3にコピーし、GR3のビット番号(8~11)を抽出する。それを4ビット左に論理シフトしてGR2に加える。
- (d) GR1をGR3にコピーし、GR3のビット番号(4~7)を抽出する。それをGR2に加える。
- (e) GR1をGR3にコピーし、GR3のビット番号(0~3)を抽出する。それを8ビット左に論理シフトしてGR2に加える。
- (f) GR1に $ABCD_{(16)}$ が格納されていて、これを $DDDD_{(16)}$ にしてGR2に格納することを考える。これには次のステップが必要になる。
 - (f-1) GR2をゼロクリアする。
 - (f-2) GR1をGR3にコピーし、GR3のビット番号(0~3)を抽出する。
それをGR2に加える。
 - (f-3) GR3を4ビット左に論理シフトしてGR2に加える。
 - (f-4) GR3を8ビット左に論理シフトしてGR2に加える。
 - (f-5) GR3を12ビット左に論理シフトしてGR2に加える。

[演習 10-2] (222 ページ)

(1)

1	SUMPM1	START		
2		XOR	GR1,GR1	GR1 0、GR1 をカウンタリセット
3		LD	GR4,N	GR4 N、GR4 はリミット
4		LAD	GR2,1	GR2 1(定数)
5		XOR	GR3,GR3	GR3 0(合計値)
6		LAD	GR5,0	フラグ: 1 なら+, 0 なら-
7	LOOP	ADDA	GR1,GR2	GR1(カウンタ) をカウントアップ
8		XOR	GR5,GR2	フラグを反転
9		JPL	PLUS	+なら PLUS へ
10		SUBA	GR3,GR1	GR3 = GR3 - GR1(カウンタ)
11		JUMP	SKIP	
12	PLUS	ADDA	GR3,GR1	GR3 = GR3 + GR1(カウンタ)
13	SKIP	CPA	GR1,GR4	GR1(汎用レジスタ)
14		JMI	LOOP	if GR1 < N then goto LOOP
15		ST	GR3,S	GR3 S
16		RET		
17	S	DS	1	
18	N	DC	100	
19		END		

(2)

1	SUMPM2	START		
2		XOR	GR1,GR1	GR1 0、GR1 をカウンタリセット
3		LAD	GR4,9	GR4 9、GR4 はリミット (ループ回数)
4		LAD	GR2,1	GR2 1(定数)
5		XOR	GR3,GR3	GR3 0(合計値)
6		LAD	GR5,0	フラグ: 1 なら+, 0 なら-
7		LAD	GR6,1	GR6 1、加える値
8	LOOP	ADDA	GR1,GR2	GR1(カウンタ) をカウントアップ
9		XOR	GR5,GR2	フラグを反転
10		JPL	PLUS	+なら PLUS へ
11		SUBA	GR3,GR6	GR3 = GR3 - GR1(カウンタ)
12		JUMP	SKIP	
13	PLUS	ADDA	GR3,GR6	GR3 = GR3 + GR1(カウンタ)
14	SKIP	SLA	GR6,1	GR6
15		CPA	GR1,GR4	GR1(汎用レジスタ)
16		JMI	LOOP	if GR1 < N then goto LOOP
17		ST	GR3,S	GR3 S
18		RET		
19	S	DS	1	
20		END		

[演習 10-3] (222 ページ)

次に示すプログラムのように、処理内容の A をビットの 1、B をビットの 0 に見立てて、データ C5CB₍₁₆₎ を作成する。このデータをシフト命令で左側 (上位ビット) から押し出し、オーバーフローしていれば A の処理を行い、オーバーフローしていなければ B の処理を行う。以下のプログラムでは、A の処理では 'A' と表示し、B の処理では 'B' と表示している。

1	ABCHECK	START	
2		LD	GR1,COM
3		LAD	GR2,1
4	LOOP	SLL	GR1,1
5		JOV	A
6	B	OUT	= 'B', =1
7		JUMP	EXITCHK
8	A	OUT	= 'A', =1
9	EXITCHK	SLL	GR2,1
10		JOV	FINISH
11		JUMP	LOOP
12	FINISH	RET	
13	COM	DC	#C5CB
14		END	

[演習 10-4] (222 ページ)

1	BCOUNT	START		
2		LD	GR1,A	
3		XOR	GR2,GR2	GR2:1 のビット数
4		LAD	GR3,16	ループ回数 (16 回)
5		LAD	GR4,1	1 (減算用・加算用の定数)
6	LOOP	LD	GR1,GR1	GR1 の値でフラグレジスタをセット
7		JZE	FINISH	もし GR1 がゼロならば終了
8		SRL	GR1,1	最下位ビットを押し出す
9		JOV	BITON	ビットが立っていれば BITON へ
10	CHECK	SUBA	GR3,GR4	GR3 (カウンタ) をカウントダウン
11		JNZ	LOOP	16 回回り切ってなければ LOOP へジャンプ
12	FINISH	ST	GR2,COUNT	ループ終了、求められたビット数を COUNT へ
13		RET		; プログラムの終了
14	BITON	ADDA	GR2,GR4	GR2 (1 のビット数) をカウントアップ
15		JUMP	CHECK	CHECK に戻ってループ続行
16	A	DC	2263	
17	COUNT	DS	1	
18		END		

演習問題 11-2-1 (228 ページ)

テーブル中の各要素を参照する場合には、多くの場合、「テーブルの先頭アドレス+指標レジスタの値」という実効アドレスを生成し、このアドレスによってテーブル中のデータをアクセスする。そのため、最初の要素をアクセスするには、指標レジスタの値は 0 にならざるを得ないのである。

演習問題 11-2-2 (228 ページ)

- (1) 1. 25 2. JMI 3. GR1,SEQ,GR1

(注:1. の回答を 24 としないこと。SEQ の 25 番目がセットされなくなる)

- (2) 1. SUBA GR2,GR3 2. GR2

演習問題 11-3 (231 ページ)

1. XOR 2. LD GR5,HDATA,GR1 3. JZE 4. GR5 5. GR3

演習問題 11-4-1 (234 ページ)

1. JZE FOUND 2. LD GR5,GR1 3. SETV

演習問題 11-4-2 (235 ページ)

COUNTT プログラムでは、テーブル MTABLE のデータをループのたびごとに汎用レジスタ GR5 にロードして、その GR5 のデータと定数 1 が格納されている汎用レジスタ GR3 の論理積を取って、奇遇の判断をしている。これでは、効率が悪いと考えて、汎用レジスタ GR3 とテーブル MTABLE の値を直接的に論理積を取っているのが、問題文に示された命令「AND GR3,MTABLE,GR1」である。

一見うまく行くように見えるが、実は、これではうまく動作しない。なぜなら、定数 1 を格納した汎用レジスタ GR3 の値が、対象となる MTABLE の要素が偶数のときに、上記の AND 命令の実行によって 0 になってしまうからである。

演習問題 11-4-3 (236 ページ)

- (1) 「LD GR2,MTABLE」によって生成される実効アドレスは「MTABLE」そのものであり、これはテーブル MTABLE の先頭アドレスである。その値が GR2 にロードされる。
- (2) 1. JMI 2. JZE
- (3) 答えは求められる。しかし、無駄な比較が最初に 1 回行われる。つまり、最初のデータを取りあえずの最大値としているのに、その値と (同じ) 最初の値を比較することになる。

演習問題 11-5-1 (245 ページ)

- (1) 1. JMI 2. GR3 3. XOR GR5,GR5 4. 9 5. ADDA 6. FFFF 7. SETDIGIT 8. 1,16,256,4096
- (2) '7' の文字コードは、表 4-5 の文字コード表によると、 $37_{(16)}$ である。これに 10 を加えると $41_{(16)}$ となり、これは 'A' の文字コードになる。また、 $37_{(16)}$ に 15 を加えると $46_{(16)}$ となり、これは 'F' の文字コードになる。

演習問題 11-5-2 (247 ページ)

- (1) 1. JPL 2. 4 3. 'A'
- (2) うまくいかない。SLA では、最上位ビットを符号と見なすため、この場合には、右側からシフトしているので、必ず 0 になる。

総合問題 11 (248 ページ)

[演習 11-1] (248 ページ)

1	TOCHAR	START		
2		XOR	GR1,GR1	テーブル MESSAGE のインデックス
3		XOR	GR2,GR2	文字数
4		LAD	GR3,1	1:定数
5	LOOP	LD	GR4,MESSAGE,GR1	
6		JZE	FINISH	0 ならおしまい
7		CPL	GR4,='a'	
8		JMI	NEXT	'a' より小さければ次へ
9		CPL	GR4,='z'	
10		JPL	NEXT	'z' より大きければ次へ
11		AND	GR4,=#FFDF	小文字 大文字
12	NEXT	ST	GR4,OUTAREA,GR1	
13		ADDA	GR1,GR3	インデックス+1
14		ADDA	GR2,GR3	文字数+1
15		JUMP	LOOP	
16	FINISH	ST	GR2,OUTLEN	
17		OUT	OUTAREA,OUTLEN	
18		RET		
19	MESSAGE	DC	'This is a long long message'	
20		DC	0	
21	OUTAREA	DS	255	
22	OUTLEN	DS	1	
23		END		

[演習 11-2] (248 ページ)

1	BINSRCH	START		
2		LD	GR0,A	GR0 A
3		LAD	GR1,1	1:定数
4		XOR	GR2,GR2	GR2 0、先頭
5		LD	GR3,N	GR3 N、個数
6		SUBA	GR3,GR1	GR3 N-1、最後の要素
7	LOOP	LD	GR4,GR3	
8		SUBA	GR4,GR2	GR4 (最後-先頭)
9		SRA	GR4,1	GR4 (最後-先頭)/2
10		LD	GR5,GR2	
11		ADDA	GR5,GR4	GR5:チェックする位置
12		LD	GR6,WTABLE,GR5	
13		CPA	GR6,GR0	
14		JZE	FOUND	見つかったか
15		CPA	GR2,GR3	インデックスのチェック
16		JZE	NOTFOUND	見つからなかったか
17		CPA	GR6,GR0	右か左のどちらかにある
18		JPL	SMALL	
19	LARGE	ADDA	GR5,GR1	左にある
20		LD	GR2,GR5	
21		JUMP	LOOP	
22	SMALL	SUBA	GR5,GR1	右にある
23		LD	GR3,GR5	
24		JUMP	LOOP	
25	FOUND	ST	GR5,FPOS	
26		RET		
27	NOTFOUND	LAD	GR5,-1	
28		RET		
29	FPOS	DS	1	
30	WTABLE	DC	-1000,-233,-58,-34,-12,0,12,122,125,456,3021,4404,9800	
31	N	DC	13	
32	A	DC	-58	
33		END		

[演習 11-3] (248 ページ)

次ページに解答プログラムを載せます。

1	KEYSUM	START		
2		LAD	GRO,1	1:定数
3	; 出力領域のクリア			
4		LD	GR1,OUTLEN	
5		LD	GR2,='	,
6	CLEARLP	SUBA	GR1,GRO	
7		JMI	STARTPGM	
8		ST	GR2,OUTAREA,GR1	
9		JUMP	CLEARLP	
10	STARTPGM	LD	GR7,OUTLEN	出力領域の長さ
11		SUBA	GR7,GRO	
12		IN	N1,N1LEN	入力データの取得
13		IN	N2,N2LEN	
14		XOR	GR6,GR6	桁上がりをゼロにセット
15		LD	GR1,N1LEN	
16		LD	GR2,N2LEN	
17		SUBA	GR1,GRO	インデックスなので
18		SUBA	GR2,GRO	1つ減じる
19	LOOP	LD	GR3,='0'	数字のゼロをセット
20		LD	GR4,='0'	数字のゼロをセット
21		LD	GR1,GR1	
22		JMI	FIN1	N1の領域が無くなったか
23		LD	GR3,N1,GR1	N1の桁をセット
24		JUMP	NEXTCHK	
25	FIN1	LD	GR2,GR2	N2の桁が無くなったか
26		JMI	FINISH	両方の桁が無かったら終了
27	NEXTCHK	LD	GR2,GR2	
28		JMI	FIN2	N2の桁が無くなったか
29		LD	GR4,N2,GR2	N2の桁をセット
30	FIN2	SUBA	GR3,='0'	数字を数値に変換
31		SUBA	GR4,='0'	数字を数値に変換
32		ADDA	GR3,GR6	前の桁上りを加える
33		XOR	GR6,GR6	次のために桁上りをクリアする
34		ADDA	GR3,GR4	桁を加える
35		CPA	GR3,=10	10以下か
36		JMI	SKIP	
37		LAD	GR6,1	桁上がりあり!
38		SUBA	GR3,=10	
39	SKIP	ADDA	GR3,='0'	数値を数字に戻す
40		ST	GR3,OUTAREA,GR7	出力領域に転送
41		SUBA	GR1,GRO	
42		SUBA	GR2,GRO	
43		SUBA	GR7,GRO	
44		JUMP	LOOP	
45	FINISH	LD	GR6,GR6	桁上がりがあるか
46		JZE	NO	なければNOへ
47		ADDA	GR6,='0'	桁上りを文字へ
48		ST	GR6,OUTAREA,GR7	出力領域へ転送
49	NO	LD	GR1,N1LEN	メッセージを付けて
50		ADDA	GR1,MESLEN	データを表示する
51		ST	GR1,N1LEN	
52		OUT	N1MES,N1LEN	
53		LD	GR2,N2LEN	
54		ADDA	GR2,MESLEN	
55		ST	GR2,N2LEN	
56		OUT	N2MES,N2LEN	
57		LD	GR3,OUTLEN	
58		ADDA	GR3,MESLEN	
59		ST	GR3,OUTLEN	
60		OUT	OUTMES,OUTLEN	
61		RET		
62	MESLEN	DC	8	
63	N1MES	DC	'DATA1	=
64	N1	DS	10	
65	N1LEN	DS	1	
66	N2MES	DC	'DATA2	=
67	N2	DS	10	
68	N2LEN	DS	1	
69	OUTMES	DC	'ANSWER=	,
70	OUTAREA	DS	255	
71	OUTLEN	DC	10	
72		END		

[演習 11-4] (248 ページ)

BINDECR.casl に答えがある。ゼロ抑制する部分は 41 行目から最後までである。その部分は次ページに示す。

1	BINDECR	START	
2	; 初期設定 =====		
3	XOR	GR1,GR1	GR1:編集格納領域のインデックス
4	LAD	GR2,1	GR2:定数 1(計算用)
5	LAD	GR3,4	GR3:重みテーブル (WTABLE) のインデックス
6	LD	GR0,A	GR0:変換するデータ
7	; 負符号の処理 =====		
8	JMI	MINUS	もし、変換するデータが負ならば '-' を付ける
9	; 数値編集のメインループ =====		
10	; 重み 10000, 1000, 100, 10 で割った数値を文字に変えて DIGIT にセット		
11	CNVLOOP	SUBA	GR3,GR2 重みテーブルのインデックスをカウントダウン
12	JMI	EXITLOOP	もし、すべての重みを計算したら終了し、1 桁目を処理
13	LD	GR4,WTABLE,GR3	GR4 重み (GR3 番目)
14	; -----		
15	; GR0 を GR4 で割った値 (GR5) と余り (GR0) を求める		
16	XOR	GR5,GR5	除算の商をゼロにしておく
17	CPA	GR0,GR4	GR0 と GR4(重み) を比較
18	JMI	DIVFIN	GR0 から GR4(重み) が引けなくなったら除算処理終わり
19	SUBA	GR0,GR4	GR0 から GR4(重み) を引く
20	ADDA	GR5,GR2	GR5(商) を 1 つカウントアップ
21	JUMP	DIVLOOP	さらに除算処理を続ける
22	; -----		
23	; 求めた桁 (GR5) を文字コードに変換して、編集領域にセット		
24	DIVFIN	ADDA	GR5,='0' 文字コード '0' を加えて、GR5 を文字コード化する
25	ST	GR5,DIGIT,GR1	文字に変換した桁を編集領域にセット
26	ADDA	GR1,GR2	GR1(編集領域インデックス) をカウントアップ
27	JUMP	CNVLOOP	次の桁の処理に移る
28	; 負符号の実際の処理 =====		
29	MINUS	LD	GR7, '-' GR7 '-'
30	ST	GR7,DIGIT,GR1	'-' を編集領域にセット
31	ADDA	GR1,GR2	GR1(編集領域インデックス) をカウントアップ
32	; GR0 の 2 の補数を求めて、符号を反転する		
33	XOR	GR0,#FFFF	ビット反転
34	ADDA	GR0,GR2	+1 する
35	JUMP	CNVLOOP	メインの処理に復帰する
36	; 最後の桁の処理 =====		
37	EXITLOOP	ADDA	GR0,='0' '0' を加えて最後の桁を文字の数値にする
38	ST	GR0,DIGIT,GR1	それを編集領域にセット
39	ADDA	GR1,GR2	GR1(編集領域インデックス) をカウントアップ (これで編集領域の
40	ST	GR1,OUTLEN	GR1 を文字数として、OUTLEN にストア 文字数になる)

41	; 先頭の空白を削除する処理			
42		LAD	GR2,5	
43		ST	GR2,DLEN	
44		XOR	GR1,GR1	DIGIT のインデックス
45		LD	GR0,DIGIT,GR1	先頭の文字を取り出す
46		CPA	GR0,='-'	負記号か
47		JNZ	NOTMINUS	
48		LAD	GR2,6	
49		ST	GR2,DLEN	
50		ADDA	GR1,=1	インデックスを次に進める
51		LD	GR0,DIGIT,GR1	次の文字を取る
52	NOTMINUS	CPA	GR0,='0'	文字のゼロか
53		JNZ	PRINT	ゼロでなければ先行するゼロはない
54		LD	GR7,GR1	GR7:詰めるゼロの先頭インデックス
55	LOOP	ADDA	GR1,=1	
56		LD	GR0,DIGIT,GR1	次の文字を取る
57		CPA	GR0,='0'	
58		JNZ	FOUND DG	0 以外の数字発見
59		CPA	GR1,=5	
60		JZE	LOOPFIN	全部 0 か
61		JUMP	LOOP	
62	FOUND DG	ST	GR0,DIGIT,GR7	数字発見, 詰める
63		ADDA	GR1,=1	
64		ADDA	GR7,=1	
65		LD	GR0,DIGIT,GR1	
66		CPA	GR1,DLEN	
67		JMI	FOUND DG	
68		ST	GR7,OUTLEN	長さなので 1 つ余分にする
69		JUMP	PRINT	
70	LOOPFIN	ST	GR0,DIGIT	
71		LAD	GR1,1	
72		ST	GR1,OUTLEN	
73	PRINT	OUT	DIGIT,OUTLEN	編集領域を出力
74		RET		;
75	A	DC	22315	変換する数値
76	WTABLE	DC	10,100,1000,10000	10 進数の重み
77	DIGIT	DS	6	編集領域
78	OUTLEN	DS	1	編集領域の長さ
79	DLEN	DS	1	数字の長さ (-のとき 6, 正のとき 5)
80		END		

演習問題 12-1-1 (251 ページ)

(1) プログラム中に記述されているのに一度も実行されないプログラムがあっても、その部分は他の部分と同様にアセンブラによって機械語に変換される。そして、それがプログラムの一部として毎回、主記憶装置にロードされてしまう。実行そのものには影響を与えないが、プログラムサイズが無意味に大きくなり、記憶効率が悪くなると言える。

(2) 条件判断を用いてある条件の時だけサブルーチンを実行させるようにすると、問題で指摘されているプログラムを作ることができる。

演習問題 12-1-2 (252 ページ)

(1)

			SP	752D ₍₁₆₎
7	752D ₍₁₆₎	サブルーチン S2 の戻り番地		
3	752E ₍₁₆₎	PUSH 3 によって積まれた値		
2	752F ₍₁₆₎	ブルーチン S1 の戻り番地		
???	7530 ₍₁₆₎	この領域は実際には使われないので何があるか不明 (OS への戻り番地の可能性もある)		

(2) サブルーチンの戻り番地はスタック上に置かれる。したがって、スタックが利用できなくなるほど多くの戻り番地をスタックに積み上げるほど CALL 命令を実行すれば、利用不可能になる。なお、スタックは、使用中のプログラムの領域に達するか 0 番地に達すると利用不可能になる。しかし、COMET II では、このチェックを自分のプログラムの中でかけることは不可能である (が、本書に付属の CASLDV ではチェックしている)。情報処理技術者試験の問題の中では、「十分な量のスタック領域があるもの」と仮定されている。

演習問題 12-1-3 (253 ページ)

(1) 「CALL SUB1」で呼び出されたサブルーチン SUB1 は、サブルーチンとして実行される。その後、「JUMP SUB1」でラベル SUB1 のある命令に制御が移るが、その中にある RET 命令によってオペレーティングシステムに制御が戻る。つまり、プログラムの実行が終了する。その意味で、「JUMP SUB1」の下の RET 命令は決して実行されることのない命令になる。

(2) CASLDV にかけるとサブルーチンの中でスタックの PUSH と POP の対応が合っていないという実行時エラーが報告される。チェック機能ははずす (メニューのオプション 実行時オプションの指定 「サブルーチン内のスタックの使用状況をチェックする」のチェックをはずす) と「PUSH 0,GR1」でスタックに PUSH した GR1 の値をサブルーチンの戻り番地として RET してしまう。

演習問題 12-1-4 (256 ページ)

演習問題 12-1 にも示したとおり、一つのサブルーチンの中で PUSH と POP の個数が合っていないと、RET 命令で正しくサブルーチンの戻り番地に戻れなくなる。

演習問題 12-2-1 (262 ページ)

1. PUSH 0,GR3
2. POP GR3

演習問題 12-2-2 (263 ページ)

サブルーチンの中で汎用レジスタの値を変更しないようにするには、スタックを用いる。

1	PUSH	0,GR3
2	PUSH	0,GR2
3	PUSH	0,GR1
4	LD	GR1,GR0 GR0 を退避
5	; サブルーチン MULTS2 の実行	
6	POP	GR0
7	POP	GR1
8	POP	GR2
9	POP	GR3

演習問題 12-3 (270 ページ)

1. JPL SHIFTL 2. JMI NEXTBIT

演習問題 12-4 (274 ページ)

1	; 再帰呼び出しで N の階乗を求めるプログラム		
2	; ただし、N の値は 7 までしかできない		
3	;		
4	; メインルーチン		
5	RFACT	START	
6		LD	GR1,N
7		CALL	FACT
8		ST	GR1,F
9		RET	
10	N	DC	7
11	F	DS	1
12	;		
13	; 1 から N までの階乗を求めるサブルーチン		
14	; 入力 GR1:N, 出力 GR1:階乗		
15	FACT	CPA	GR1,=0 もし、N(GR1) がゼロかどうか
16		JNZ	CAL ゼロでなければ CAL へ
17		LAD	GR1,1 GR1 1
18		RET	
19	CAL	PUSH	0,GR1 N
20		SUBA	GR1,=1 GR1
21		CALL	FACT FACT を呼ぶ。すなわち、
22		POP	GR2 N
23	; GR3 GR1 x GR2 = S(N-1) x N, GR2 はゼロではない		
24		XOR	GR3,GR3 GR3 0
25	MULTLOOP	ADDA	GR3,GR1 GR3 GR3+GR1
26		SUBA	GR2,=1 GR2 GR2-1, カウントダウン
27		JNZ	MULTLOOP
28		LD	GR1,GR3
29		RET	
30		END	

総合問題 12 (275 ページ)

[演習 12-1] (275 ページ)

1. LAD 2. LD GR6,0,GR1 3. LD 4. ADDA GR1,GR3

[演習 12-2] (276 ページ)

1. LD GR5,='¥' 2. SUBA GR5,GR6 3. ADDA GR4,GR7 4. LD GR6,='.' '

索引

IN 命令, 36

PR, 5

オーバーフロー, 45

階乗, 39

完全数, 42

奇数, 31

偶数, 31

時間の表記, 2

実行開始アドレス, 7

実行時トレース, 42

実行時ライブラリ, 1

スタック, 55

絶対番地, 6

ゼラーの公式, 37

ゼロクリア, 30

素数, 40, 41

算盤, 2

特権命令, 1

ビットの抽出, 45

編集, 53

文字コード, 49

戻り番地, 55

ラベル, 6