

本日の授業のテーマ

- (1) 数値データを文字データに変換
  - ・
  - ・
- (2) 文字データを数値データに変換
  - ・
  - ・

## 1 数値データを文字データに変換

### 1.1 プログラムの仕様

教科書の p.108 [例題 10] プログラムの仕様は次の通りです。

- ラベル A に入っている数値 (最大 5 桁) を文字列に変換して、OUT 命令で表示する。
- 表示は 6 桁 (カラム) 用意して、第 1 桁は符号、2~6 桁で絶対値をあらわす。ただし、上位の桁が 0 の場合、スペースを入れる。

数値	表示
-32768	- 3 2 7 6 8
-2693	-   2 6 9 3
-739	-   7 3 9
-12	-   1 2
-8	-   8
0	0
6	6
84	8 4
813	8 1 3
5173	5 1 7 3
32767	3 2 7 6 7

図 1 数値と表示の関係

ラベル A の数値が最大 5 桁というのは、ラベル A の値を符号付き整数と考えるからです。符号付整数の場合、1 ワードで表現できるのは、-32768~32767 の範囲です。これは 5 桁で、表示する場合、符号を合わせて 6 カラム必要です。

REI5A START

```

LAD GR1, 0 ;カウンタを0にする
LD GR0, A ;Aの値をGR0に入れる

JMI FUSUU ;負数が判断

LD GR3,=#0020 ;正のときはスペース
ST GR3, BUFF, GR1 ;文字を保存
JUMP KEISAN ;数値処理にジャンプ

FUSUU LD GR3, '-' ;負数の時はマイナス
ST GR3, BUFF, GR1 ;文字を保存
LAD GR0, 0 ;絶対値処理
SUBA GR0, A ;絶対値処理

KEISAN LAD GR1, 1, GR1 ;カウンタをインクリメント
LD GR2, WORK, GR1 ;除数をGR2に読み込み
CALL DIV ;割り算処理

PUSH 0, GR3 ;GR3を一時退避
OR GR3, WORK ;文字コードの判別
ST GR3, WORK ;結果をWORKに
POP GR3 ;GR3を復帰
JZE SPACE ;ORの結果が0のとき分岐
OR GR3, MOJI ;#0030を加算
JUMP HOZON ;文字コードの保存処理へ

SPACE LD GR3,=#0020 ;ORが0のときスペース
HOZON ST GR3, BUFF, GR1 ;文字コードの保存

CPA GR1, C4 ;ループ回数の比較
JMI KEISAN ;4回ループする

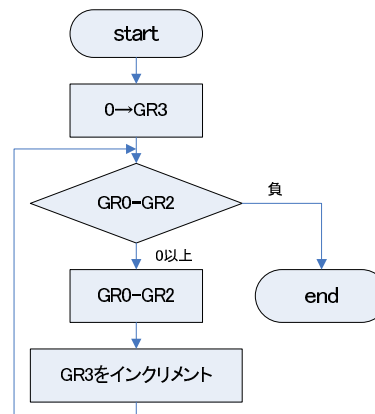
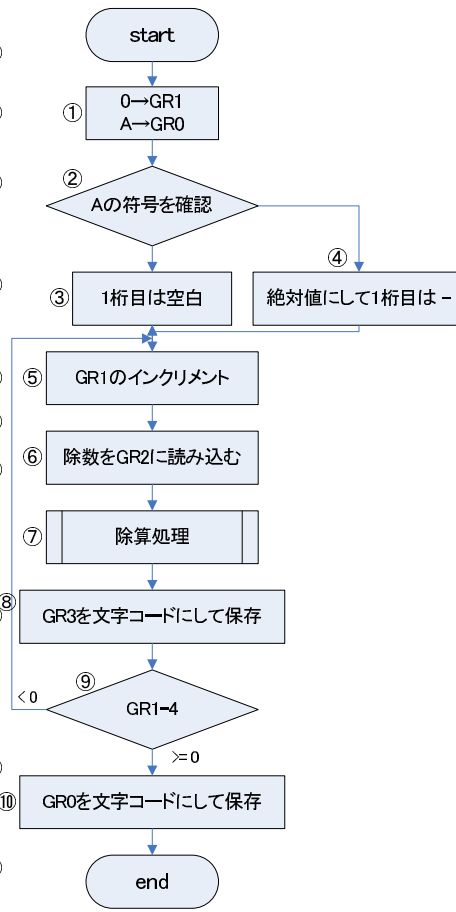
OR GR0, MOJI ;1の位を文字コードに
LAD GR1, 1, GR1 ;カウンタをインクリメント
ST GR0, BUFF, GR1 ;文字コードの保存
OUT BUFF, LEN ;結果の表示
RET
  
```

```

A DC -2050
BUFF DS 6 ;文字コードの保存領域
LEN DC 6 ;文字コードの長さ
WORK DC 0, 10000, 1000, 100, 10
MOJI DC #0030
C4 DC 4
END
  
```

```

DIV START
LAD GR3, 0 ;商を0にする
DLOOP CPA GR0, GR2 ;除数で引けるか判別
JMI FIN ;引けなければFINへ
SUBA GR0, GR2 ;被除数-除数
LAD GR3, 1, GR3 ;商をインクリメント
JUMP DLOOP ;繰り返し処理へ
FIN RET
END
  
```



## 2 文字データを数値データに変換

### 2.1 プログラムの仕様

教科書の p.115 [例題 11] プログラムの仕様は次の通りです。

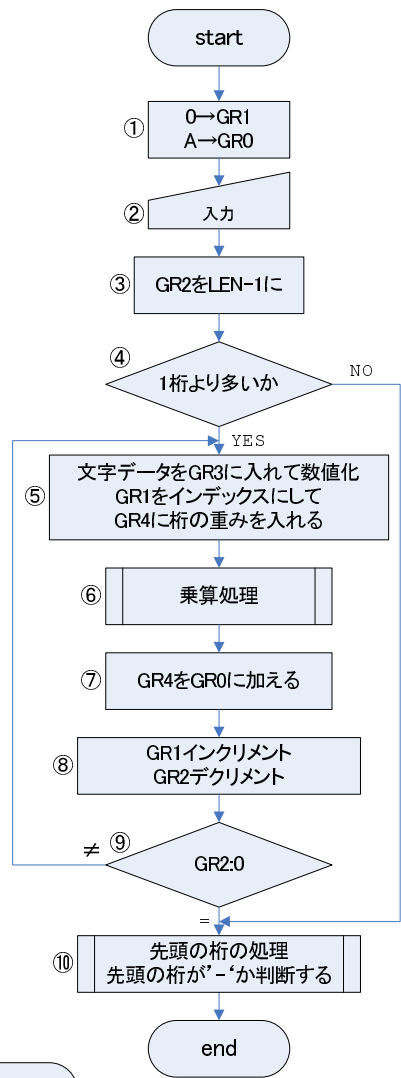
- IN 命令によりキーボードから入力した値を数値データに変換して、GR0 に入れる。
- キーボードから入力する数値は、0～9 と ' - ' のみで表す。
- 正の場合、最大 5 桁とする。負の場合は絶対値最大 5 桁、符号の 1 桁で合計 6 桁とする。

要するに、ここでキーボードから入力する数値の範囲は、1 ワードで表現できる -32768～32767 の整数とします。

```

REI5B  START
      LAD  GR0,0      ;各桁を加算するレジスタ
      LAD  GR1,0      ;重みを入れるレジスタ
      IN   BUFF,LEN   ;文字の入力
      LD   GR2,LEN    ;桁数を読み込む
      LAD  GR2,-1,GR2 ;制御変数をデクリメント
      CPA  GR2,=0     ;0と比較
      JZE  SAIGO      ;入力した文字が1桁の時
      YOMU LD  GR3,BUFF,GR2 ;文字の読み込み
           AND  GR3,#000F ;マスク処理(数値データへ)
           LD   GR4,WK,GR1 ;桁の重み
           CALL MULTI      ;乗算処理
           ADDA GR0,GR4    ;各桁を累算する
           LAD  GR1,1,GR1 ;インクリメント
           LAD  GR2,-1,GR2 ;制御変数のデクリメント
           SLA  GR2,0     ;フラグのセット
           JNZ  YOMU      ;GR2でないときは繰り返し
      SAIGO LD  GR3,BUFF ;先頭の桁の処理
           CPA  GR3,='- ' ;先頭の桁がマイナスか判断
           JZE  MINUS    ;マイナスの時は分岐
           AND  GR3,#000F ;プラスのときはマスク処理
           LD   GR4,WK,GR1 ;重みの読み込み
           CALL MULTI      ;乗算処理
           ADDA GR0,GR4    ;各桁を累算する
           JUMP OWARI     ;処理終了
      MINUS XOR  GR0,#FFFF ;マイナスのときの処理
           ADDA GR0,=1    ;1の補数にして1加算
      OWARI RET
      BUFF DS 10        ;文字データの格納場所
      LEN  DS 1
      WK   DC 1,10,100,1000,10000
      END

```



```

MULTI  START
      CPA  GR3,=0     ;数値データがゼロか判断
      JZE  ZERO      ;ゼロの時は分岐
      ST   GR4,TEMP   ;GR4を一時保存
      MLOOP LAD  GR3,-1,GR3 ;デクリメント
           CPA  GR3,=0     ;ゼロか判断
           JZE  FIN      ;ゼロの時は分岐
           ADDA GR4,TEMP   ;桁の重みを加算する
           JUMP MLOOP    ;繰り返しへ
      ZERO LAD  GR4,0   ;GR3が0のとき
      FIN  RET
      TEMP DS 1
      END

```

