



プログラミング演習 1

課題4：アセンブリ言語

鶴亀算

奇数の合計



本日の授業内容

1. 例題7: 鶴亀算
2. 例題8: 奇数の合計
3. レポートについて
4. 練習問題

鶴亀算

□前提

- ツルの羽数とカメの匹数の合計はわかっている $= A$
- 足の数もわかっている $= B$

□解法

- カメの匹数 $= (B - 2A)/2$
- ツルの羽数 $= A - \text{カメの匹数}$

□例: $A = 10, B = 26$

- カメの匹数 $= (26 - 2 \times 10)/2 = 3$
- ツルの羽数 $= 10 - 3 = 7$

鶴亀算

□ カメの匹数: $(\text{足数} - \text{頭数} \times 2) / 2$

- GR0: 足数
- GR0から頭数を2回引く
(足数 - 頭数 × 2)
- GR0を算術右1ビットシフト
(GR0 ÷ 2) → なぜ?
- GR0をKAMEへストア
- KAMEを10進数に変換 (サブルーチン)

; 例7 鶴亀算 (サブルーチンの使い方)

EX7 START

```
LD   GR0,ASI
SUBA GR0,ATAMA
SUBA GR0,ATAMA ;足の総数-頭の総数×2
SRA  GR0,1     ;÷2
ST   GR0,KAME
LAD  GR1,6     ;亀桁位置設定
CALL DECCNV    ;10進数に変換
```

```
LD   GR0,ATAMA
SUBA GR0,KAME
LAD  GR1,23    ;鶴桁位置設定
CALL DECCNV    ;10進数に変換
OUT  OBUF,OLEN
RET
```

```
ATAMA DC 10
ASI   DC 26
KAME  DS 1
OBUF  DC 'KAME   HIKI:TURU   WA'
OLEN  DC 31
```

; SUBROUTINE (10進数に変換)

```
DECCNV LAD GR3,0 ;文字列を数えるループカウンタ
LOOP1  LAD GR2,-1 ;商のカウンタ
LOOP2  LAD GR2,1,GR2 ;商のカウンタ+1
SUBA   GR0,CONST,GR3
JPL    LOOP2
JZE    LOOP2
ADDA   GR0,CONST,GR3 ;GR0←(GR0)+(CONST+GR3)
ADDA   GR2,ZONE      ;ゾーン部付加
ST     GR2,OBUF,GR1
LAD    GR1,1,GR1     ;桁位置のポインタ+1
LAD    GR3,1,GR3     ;ループのカウンタ+1
CPA    GR3,F5
JMI    LOOP1         ;終り?
RET
```

```
ZONE DC '0'
CONST DC 10000,1000,100,10,1
F5    DC 5
END
```

論理シフト(SLL,SRL)と算術シフト(SLA,SRA)

□論理シフト

➤ [ラベル] SRL(SLL) r, adr[,x]

rの内容をadr[,x]ビット右（左）にシフト（符号ビット含む）

□算術シフト

➤ [ラベル] SRA(SLA) r, adr[,x]

rの内容をadr[,x]ビット右（左）にシフト（符号ビット除く）

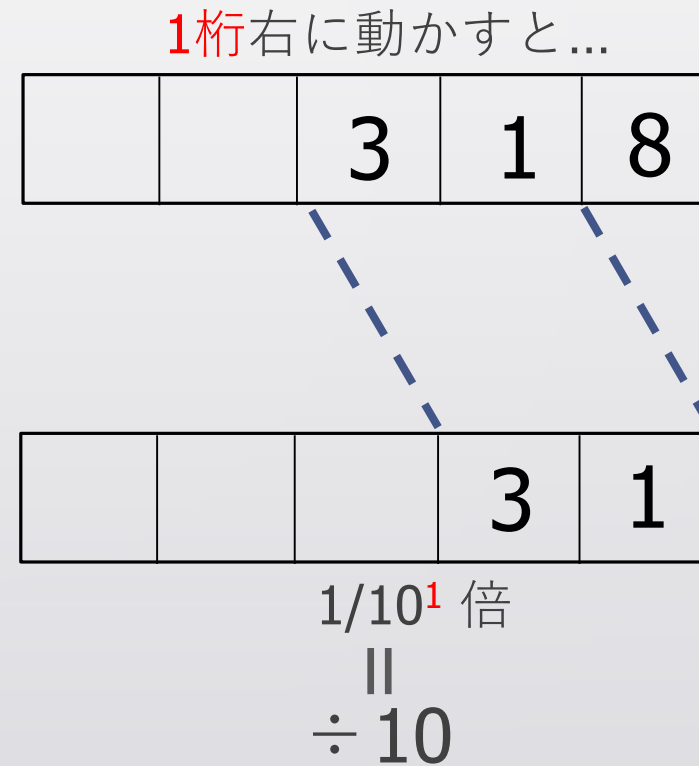
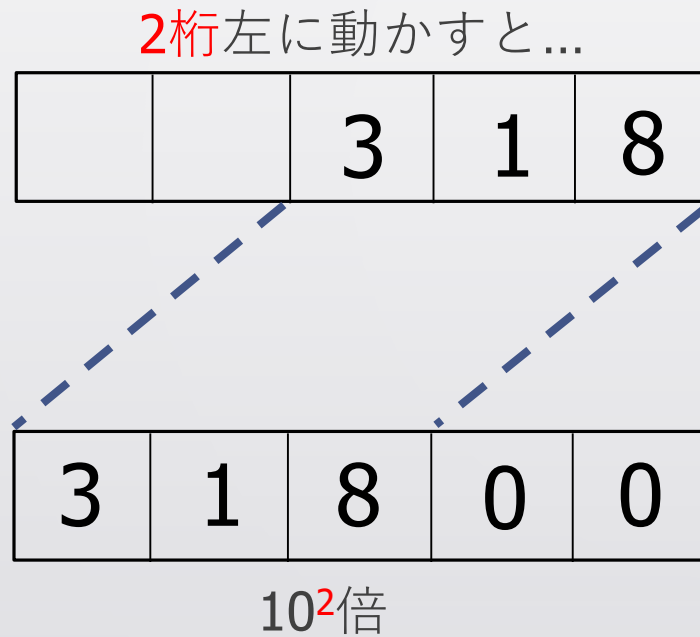
□例) SRL (SRA) r(=#FF9F), 4

➤ SRL: 1111 1111 1001, 1111 → 0000 1111 1111 1001 (0で埋める)

➤ SRA: 1111 1111 1001, 1111 → 1111 1111 1111 1001 (符号ビットで埋める)

なぜビットシフトで割り算できるのか？

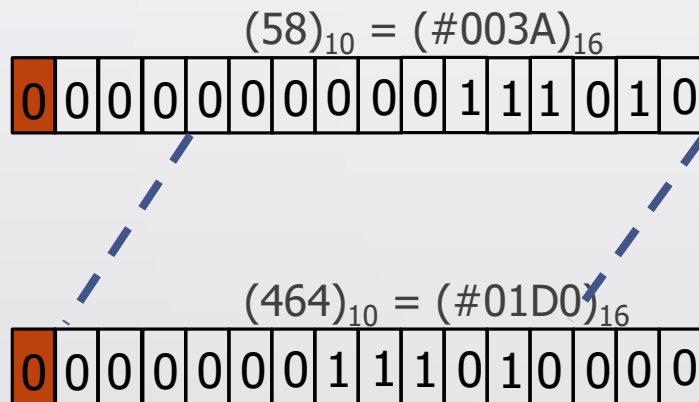
□10進数の場合



なぜビットシフトで割り算できるのか？

□2進数の場合

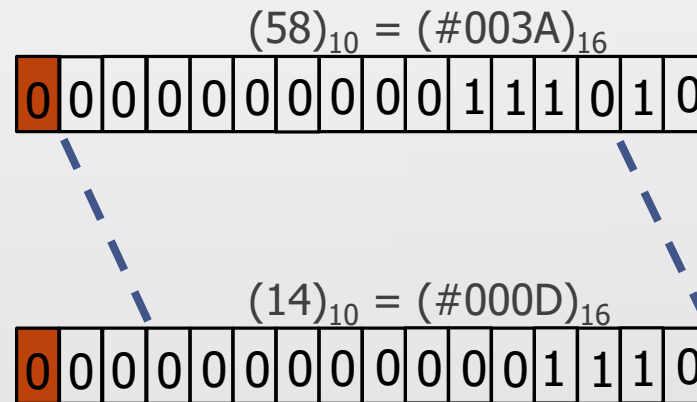
3桁左に動かすと...



58 $\Rightarrow$ 464 (8倍 = 2³倍)

n桁 左にシフト 2ⁿ倍

2桁右に動かすと...



58 $\Rightarrow$ 14 (1/4倍 = 1/2²倍)
÷ 4

m桁 右にシフト 1/2^m倍 $\Rightarrow$ ÷ 2^m

鶴亀算

□ カメの匹数: $(\text{足数} - \text{頭数} \times 2) / 2$

- GR0: 足数
- GR0から頭数を2回引く
(足数 - 頭数 × 2)
- GR0を算術右1ビットシフト
(GR0 ÷ 2) → なぜ?
- GR0をKAMEへストア
- KAMEを10進数に変換 (サブルーチン)

; 例7 鶴亀算 (サブルーチンの使い方)

EX7 START

```
LD  GR0,ASI
SUBA GR0,ATAMA
SUBA GR0,ATAMA ;足の総数-頭の総数×2
SRA  GR0,1 ;÷2
ST   GR0,KAME
LAD  GR1,6 ;亀桁位置設定
CALL DECCNV ;10進数に変換
```

```
LD  GR0,ATAMA
SUBA GR0,KAME
LAD  GR1,23 ;鶴桁位置設定
CALL DECCNV ;10進数に変換
OUT  OBUF,OLEN
RET
```

```
ATAMA DC 10
ASI DC 26
KAME DS 1
OBUF DC 'KAME' HIKI:TURU WA'
OLEN DC 31
```

;SUBROUTINE (10進数に変換)

```
DECCNV LAD GR3,0 ;文字列を数えるループカウンタ
LOOP1 LAD GR2,-1 ;商のカウンタ
LOOP2 LAD GR2,1,GR2 ;商のカウンタ+1
SUBA GR0,CONST,GR3
JPL LOOP2
JZE LOOP2
ADDA GR0,CONST,GR3 ;GR0←(GR0)+(CONST+GR3)
ADDA GR2,ZONE ;ゾーン部付加
ST GR2,OBUF,GR1
LAD GR1,1,GR1 ;桁位置のポインタ+1
LAD GR3,1,GR3 ;ループのカウンタ+1
CPA GR3,F5
JMI LOOP1 ;終り?
RET
```

```
ZONE DC '0'
CONST DC 10000,1000,100,10,1
F5 DC 5
END
```

サブルーチン

- 主となるプログラムから見て別の場所に定義されたプログラム
(C言語の関数のような機能)
 - 同じ処理を複数回する必要がある場合
 - 例題6の10進数の印刷をサブルーチンとして利用
- CALL命令で呼び出し
 - [ラベル] CALL adr[,x] (例題ではadr[,x]がラベルDECCNV)
- RET命令でリターン
 - 呼び出し元へ戻る

鶴亀算

□ ツルの羽数: 頭数 - KAME

- GR0に頭数をロード
- GR0 - KAME
- GR0(ツルの頭数)を10進数に変換
(サブルーチン)

;例7 鶴亀算 (サブルーチンの使い方)

```
EX7 START
LD GR0,ASI
SUBA GR0,ATAMA
SUBA GR0,ATAMA ;足の総数-頭の総数×2
SRA GR0,1 ;÷2
ST GR0,KAME
LAD GR1,6 ;亀桁位置設定
CALL DECCNV ;10進数に変換
LD GR0,ATAMA
SUBA GR0,KAME
LAD GR1,23 ;鶴桁位置設定
CALL DECCNV ;10進数に変換
OUT OBUF,OLEN
RET
ATAMA DC 10
ASI DC 26
KAME DS 1
OBUF DC 'KAME HIKI:TURU WA'
OLEN DC 31
;
;SUBROUTINE (10進数に変換)
DECCNV LAD GR3,0 ;文字列を数えるループカウンタ
LOOP1 LAD GR2,-1 ;商のカウンタ
LOOP2 LAD GR2,1,GR2 ;商のカウンタ+1
SUBA GR0,CONST,GR3
JPL LOOP2
JZE LOOP2
ADDA GR0,CONST,GR3 ;GR0←(GR0)+(CONST+GR3)
ADDA GR2,ZONE ;ゾーン部付加
ST GR2,OBUF,GR1
LAD GR1,1,GR1 ;桁位置のポインタ+1
LAD GR3,1,GR3 ;ループのカウンタ+1
CPA GR3,F5
JMI LOOP1 ;終り?
RET
ZONE DC '0'
CONST DC 10000,1000,100,10,1
F5 DC 5
END
```

例題8: 奇数の合計

□ 1から100までの奇数の合計を10進数で印刷

□ 手順

- 合計格納用GR0=0・奇数格納GR1=1を指定(初期値)
- GR0 += GR1 (GR0に奇数を足していく)
- GR1 += 2 (GR1は奇数なので)
- GR1が100を超えているかどうか
(超えていなかったらLOOPへ戻る)

```
-----  
; 例8 奇数の合計  
-----  
EX8  START  
      LAD  GR0,0      ;合計を格納するレジスタGR0 ← 0  
      LAD  GR1,1      ;奇数を格納するレジスタGR1 ← 1  
LOOP  ADDA  GR0,GR1    ;GR0 ← ( GR0 + GR1 )  
      LAD  GR1,2,GR1  ;GR1 ← 2 + GR1  
      CPA  GR1,F100    ;GR1とF100を比較 (GR1) - (F100)  
      JMI  LOOP        ;比較結果が <0 ならLOOP へ分岐  
      JZE  LOOP        ;比較結果が =0 ならLOOP へ分岐  
      LAD  GR1,16      ;GR1 ← 16  
      CALL  DECPRT     ;10進数に変換  
      RET  
F100  DC   100  
;  
;SUBROUTINE (10進数に変換)  
DECPRT LAD  GR3,0      ;GR3 ← 0  
LOOP1  LAD  GR2,-1     ;GR2 ← -1  
LOOP2  LAD  GR2,1,GR2  ;GR2 ← 1 + GR2  
      SUBA  GR0,CONST,GR3 ;GR0 ← ( GR0 ) - ( CONST + GR3 )  
      JPL  LOOP2        ;減算結果が >0 ならLOOP へ分岐  
      JZE  LOOP2        ;減算結果が =0 ならLOOP2 へ分岐  
      ADDA  GR0,CONST,GR3 ;GR0 ← ( GR0 ) + ( CONST + GR3 )  
      ADDA  GR2,ZONE     ;GR2 ← ( GR2 ) + ( ZONE )  
      ST   GR2,OUTBUF,GR1 ;GR2を(OUTBUF + GR1)番地に格納  
      LAD  GR1,1,GR1    ;GR1 ← 1 + GR1  
      LAD  GR3,1,GR3    ;GR3 ← 1 + GR3  
      CPA  GR3,F5       ;GR3とF5を比較 (GR3)-(F5)  
      JMI  LOOP1        ;比較結果が <0 ならLOOP1 へ分岐  
      OUT  OUTBUF,OUTLEN ;文字を出力する  
      RET  
ZONE  DC   '0'  
OUTBUF DC   'SUM OF ODD NUM= '  
OUTLEN DC   21  
CONST  DC   10000,1000,100,10,1  
F5     DC   5  
END
```

例題8: 奇数の合計

□ 1から100までの奇数の合計を10進数で印刷

□ 手順（のつづき）

➤ 計算結果=GR0を格納する桁位置設定
（GR1が奇数用から桁位置指定用になるので注意）

➤ GR0を10進数に変換・印刷（サブルーチン）

□ 以上の流れをフローチャートで表してみよう！
（レポート課題ではフローチャートを
示すことが必要）

```
;-----  
; 例8 奇数の合計  
;-----  
EX8  START  
      LAD  GR0,0      ;合計を格納するレジスタGR0 ← 0  
      LAD  GR1,1      ;奇数を格納するレジスタGR1 ← 1  
LOOP  ADDA  GR0,GR1    ;GR0 ← ( GR0 + GR1 )  
      LAD  GR1,2,GR1   ;GR1 ← 2 + GR1  
      CPA  GR1,F100    ;GR1とF100を比較 (GR1) - (F100)  
      JMI  LOOP        ;比較結果が <0 ならLOOP へ分岐  
      JZE  LOOP        ;比較結果が =0 ならLOOP へ分岐  
      LAD  GR1,16      ;GR1 ← 16  
      CALL DECPRT      ;10進数に変換  
      RET  
F100  DC   100  
;  
;SUBROUTINE (10進数に変換)  
DECPRT LAD  GR3,0      ;GR3 ← 0  
LOOP1  LAD  GR2,-1     ;GR2 ← -1  
LOOP2  LAD  GR2,1,GR2  ;GR2 ← 1 + GR2  
      SUBA  GR0,CONST,GR3 ;GR0 ← ( GR0 ) - ( CONST + GR3 )  
      JPL  LOOP2        ;減算結果が >0 ならLOOP へ分岐  
      JZE  LOOP2        ;減算結果が =0 ならLOOP2 へ分岐  
      ADDA  GR0,CONST,GR3 ;GR0 ← ( GR0 ) + ( CONST + GR3 )  
      ADDA  GR2,ZONE     ;GR2 ← ( GR2 ) + ( ZONE )  
      ST   GR2,OUTBUF,GR1 ;GR2を(OUTBUF + GR1)番地に格納  
      LAD  GR1,1,GR1    ;GR1 ← 1 + GR1  
      LAD  GR3,1,GR3    ;GR3 ← 1 + GR3  
      CPA  GR3,F5       ;GR3とF5を比較 (GR3)-(F5)  
      JMI  LOOP1        ;比較結果が <0 ならLOOP1 へ分岐  
      OUT  OUTBUF,OUTLEN ;文字を出力する  
      RET  
ZONE  DC   '0'  
OUTBUF DC   'SUM OF ODD NUM= '  
OUTLEN DC   21  
CONST  DC   10000,1000,100,10,1  
F5     DC   5  
END
```

練習課題

1. 素数判定

- ある数 n (ユーザ指定)が素数か否かを判定するプログラムを作成
- n が m で割り切れるか否かの判定はサブルーチンで

2. 10進数のソート(オプション)

- **DATA**番地に格納されている正の**10**進数データを昇順にソート
- データ印刷部・**10**進数変換部はサブルーチン

※講義ページにヒントがあるので参考にとすること