

## CASL II プログラミング 練習問題 解答

【1】 次のプログラムを実行した後の各レジスタの内容を4桁の16進数で答えよ。

```
P001  START
      LD   GR3, DAT1
      LD   GR0, DAT2
      LD   GR1, DAT3
      RET
DAT1  DC   10
DAT2  DC   -1
DAT3  DC   100
      END
```

実行前

```
GR0 : #0023  GR1 : #4567  GR2 : #89AB  GR3 : #CDEF  GR4 : #0000
      #FFFF          #0064          #000A
```

【2】 次のプログラムの空欄を埋めよ。

```
P002  START
      LD   GR0, =0 ;GR0 を 0 にする
      LD   GR1, LMAX ;GR1 に LMAX の内容を設定する
      RET
LMAX  DC   25
      END
```

【3】 次のコーディングは誤っている。その理由を説明し、正しくコーディングせよ。

```
LD   GR0, 1 ;GR0 に 1 を設定する
      =1
```

【4】 次のプログラムはA番地の内容をB番地に格納するものである。空欄を埋めよ。

```
P004  START
      LD   GR0, A
      ST   GR0, B ;(GR0) → B
      RET
A     DC   #544D
B     DS   1
      END
```

【5】 次のプログラムは各エリアをゼロクリアするものである。空欄を埋めよ。

```
P005  START
      LD   GR2, C0           ;GR2 を 0 にする
      ST   GR2, GOKEI1      ;GOKEI1 を 0 にする
      ST   GR2, GOKEI2      ;GOKEI2 を 0 にする
      ST   GR2, GOKEI3      ;GOKEI3 を 0 にする
      RET
C0    DC    0
GOKEI1 DS  1
GOKEI2 DS  1
GOKEI3 DS  1
      END
```

【6】 次のコーディングは誤っている。その理由を説明し、正しくコーディングせよ。

```
      ST   0, SW             ;SW 番地に 0 を格納する
      ST   AST, HYOKA        ;AST 番地の内容を HYOKA 番地に格納する
      }
      LD   GR0, =0
SW    DS   1                 ST   GR0, SW
AST   DC   '*'              LD   GR0, AST
HYOKA DS   1                 ST   GR0, HYOKA
```

【7】 次のプログラムを実行した後の GR1 の内容を符号付の 10 進数で答えよ。

```
      LD   GR1, PMAX
      ADDA GR1, =1           -32768
      }
PMAX  DC   32767
```

【8】 各 ADDA 命令を実行した後の GR1 の内容を符号付の 10 進数で、FR の内容を 3 桁の 2 進数で答えよ。

```
      LD   GR1, DAT1
      ADDA GR1, C1           32766  000
      ADDA GR1, C1           32767  000
      ADDA GR1, C1           -32768 110
      }
DAT1  DC   32765
C1    DC   1
```

【9】 次のプログラムは SU 番地の内容を 4 倍して BAI4 に設定するものである。空欄を埋めよ。

```
LD      GR3, SU
ADDA   GR3, SU      ;2倍する
ADDA   GR3, GR3     ;2倍した値をさらに2倍
ST     GR3, BAI4    ;4倍した値をBAI4へ
      <
SU     DC      1
BAI4   DS      1
```

【10】 次のプログラムは合計計算を行うものである。空欄を埋めよ。

```
P010   START
      LD      GR0, KOKUGO ;KOKUGOをGR0に設定する
      ADDA   GR0, EIGO    ;EIGOを加える
      ADDA   GR0, SUGAKU ;SUGAKUを加える
      ST     GR0, GOKEI
      RET
KOKUGO DC      80
EIGO   DC      85
SUGAKU DC      79
GOKEI  DS      1
      END
```

【11】 次のプログラムは合計計算を行うものである。空欄を埋めよ。

```
P011   START
      LD      GR0, =0     ;GR0を0クリアする
      ADDA   GR0, KOKUGO ;KOKUGOを加える
      ADDA   GR0, EIGO    ;EIGOを加える
      ADDA   GR0, SUGAKU ;SUGAKUを加える
      ST     GR0, GOKEI
      RET
KOKUGO DC      80
EIGO   DC      85
SUGAKU DC      79
GOKEI  DS      1
      END
```

【12】 次のコーディングは誤っている。その理由を説明し、正しくコーディングせよ。

```
ADDA   GR0, 1      ;GR0に1を加える
      =1
```

【13】 次のプログラムを実行した後の GR1 の内容を符号付の 10 進数で答えよ。

```
LD      GR1, PMAX
ADDL   GR1, =1      0
      }
PMAX   DC      65535
```

【14】 各 ADDL 命令を実行した後の GR1 の内容を符号なし 10 進数で、FR の値を 3 桁の 2 進数で答えよ。

```
LD      GR1, DAT1
ADDL   GR1, C1      65534    010
ADDL   GR1, C1      65535    010
ADDL   GR1, C1      0        101
      }
DAT1   DC      65533
C1     DC      1
```

【15】 各 SUBA 命令を実行した後の GR0 の内容を 4 桁の 16 進数で、FR の値を 3 桁の 2 進数で答えよ。

```
LD      GR0, =2
SUBA   GR0, C1      #0001    000
SUBA   GR0, C1      #0000    001
SUBA   GR0, C1      #FFFF    010
      }
C1     DC      1
```

【16】 次のプログラムを実行した後の GR2 の値を答えよ。

```
P016   START
LD      GR2, DT1
SUBA   GR2, DT1      0
RET
DT1    DC      8
END
```

【17】 次のプログラムは DA 番地の内容を符号を変えて DB 番地に格納するものである。空欄を埋めよ。

```
P017   START
LD      GR1,  ;GR1 を 0 にする
SUBA   GR1, 
ST      GR1, 
RET
DA     DC      8
DB     DS      1
END
```

【18】 次のプログラムは前問でリテラルを使用しない方法によるものである。空欄を埋めよ。

```

P018  START
      LD   GR1,  ; } GR1 を 0 にする
      SUBA GR1,  ; }
      SUBA GR1, 
      ST   GR1, 
      RET
DA    DC    8
DB    DS    1
      END

```

【19】 次のコーディングは誤っている。その理由を説明し、正しくコーディングせよ。

```

SUBA  GR0, 1          ;GR0 から 1 を引く
      =1

```

【20】 各 SUBL 命令を実行した後の GR0 の内容を 4 桁の 16 進数で、FR の値を 3 桁の 2 進数で答えよ。

```

LD    GR0, C2
SUBL  GR0, C1        #0001    000
SUBL  GR0, C1        #0000    001
SUBL  GR0, C1        #FFFF    110
      <
C1    DC    1
C2    DC    2

```

【21】 次のプログラムは AB 番地の内容と CD 番地の内容を交換するものである。空欄を埋めよ。

```

P021  START          ;GR0 しか使えない場合
       ; } AB 番地の内容を退避
       ; }
       ; } CD 番地の内容を AB 番地へ
       ; }
       ; } 退避した AB 番地の内容を CD 番地へ
       ; }
      RET
AB    DC    100
CD    DC    200
WK    DS    1
      END

```

【22】 次のプログラムは AB 番地の内容と CD 番地の内容を交換するものである。空欄を埋めよ。

```

P022  START                                ;GR2 と GR3 の 2 つのレジスタを使える場合
      LD  GR2, AB                          ;AB 番地の内容を GR2 に退避
      LD  GR3, CD                          ;CD 番地の内容を GR3 に退避
      ST  GR2, CD                          ; } 退避した AB 番地と CD 番地の内容を入替え
      ST  GR3, AB                          ; }
      RET
AB     DC     100
CD     DC     200
WK     DS     1
      END

```

【23】 次のプログラムはシフト命令を使用しないで数を 10 倍するものである。空欄を埋めよ。

```

      (4X + X) * 2 = 10X
      LD  GR0, SU
      ADDA GR0, SU                          ;2 倍
      ADDA GR0, GR0                         ;4 倍
      ADDA GR0, SU                          ;4 倍 + 元の数
      ADDA GR0, GR0                         ;5 倍した値を 2 倍する
      <
SU     DC     4

```

【24】 次のプログラムはシフト命令を使用しないで数を 10 倍するものである。空欄を埋めよ。

```

      8X + 2X = 10X
      LD  GR0, SU
      ADDA GR0, SU                          ;2 倍
      ST  GR0, WK1
      ADDA GR0, GR0                         ;4 倍 (2 倍した値を 2 倍する)
      ADDA GR0, GR0                         ;8 倍 (4 倍した値を 2 倍する)
      ADDA GR0, WK1                         ;8 倍+2 倍した値
      <
SU     DC     4
WK1    DS     1

```

【25】 次の ADDA 命令を実行した後の FR の値を 3 桁の 2 進数で答えよ。

```
LD      GR0, =#0008
ADDA    GR0, C30      000
  }
C30    DC      #0030
```

【26】 次の ADDA 命令を実行した後の FR の値を 3 桁の 2 進数で答えよ。

```
LD      GR1, =#7FFF
ADDA    GR1, C1      110
  }
C1     DC      #0001
```

【27】 次の SUBA 命令を実行した後の FR の値を 3 桁の 2 進数で答えよ。

```
LD      GR2, =#0003
SUBA    GR2, DT1     001
  }
DT1    DC      #0003
```

【28】 次の ADDA 命令を実行した後の FR の値を 3 桁の 2 進数で答えよ。

```
LD      GR3, =#FFFF
ADDA    GR3, M1     010
  }
M1     DC      #FFFF
```

【29】 次の SUBA 命令を実行した後の FR の値を 3 桁の 2 進数で答えよ。

```
LD      GR0, CHAR
SUBA    GR0, H3A     010
  }
CHAR   DC      '9'
H3A    DC      #003A
```

【30】 次の SUBA 命令を実行した後の FR の値を 3 桁の 2 進数で答えよ。

```
LD      GR0, CHAR
SUBA    GR0, H3A     001
  }
CHAR   DC      ':'
H3A    DC      #003A
```

【31】 次の SUBA 命令を実行した後の FR の値を 3 桁の 2 進数で答えよ。

```
LD      GR0, CHAR
SUBA    GR0, H3A      000
}
CHAR    DC      `;'
H3A     DC      #003A
```

【32】 各命令を実行した後の GR0 の内容を符号付き 10 進数で、FR の値を 3 桁の 2 進数で答えよ。

```
LD      GR0, C32      32766      000
ADDA    GR0, C1       32767      000
ADDA    GR0, C1      -32768      110
LD      GR0, C2         2         000
SUBA    GR0, C1         1         000
SUBA    GR0, C1         0         001
SUBA    GR0, C1        -1         010
}
C1      DC      1
C2      DC      2
C32     DC      32766
```

【33】 次の CPA 命令を実行した後の FR の値を 3 桁の 2 進数で答えよ。

```
LD      GR1, X
CPA     GR1, Y      000
}
X       DC      #000D
Y       DC      #000B
```

【34】 次の CPA 命令を実行した後の FR の値を 3 桁の 2 進数で答えよ。

```
LD      GR1, X
CPA     GR1, Y      010
}
X       DC      #FFFF
Y       DC      #000A
```

【35】 次の CPA 命令を実行した後の FR の値を 3 桁の 2 進数で答えよ。

```
LD      GR1, X
CPA     GR1, Y      001
}
X       DC      #8000
Y       DC      #8000
```

【36】 次の CPA 命令を実行した後の FR の値を 3 桁の 2 進数で答えよ。

```
LD      GR1, X
CPA     GR1, Y      000
}
X       DC      #7FFF
Y       DC      #8001
```

【37】 次のコーディングは誤っている。その理由を説明し、正しくコーディングせよ。

```
CPA     GR0, 10      ;GR0 の内容と 10 を比較する
          =10
```

【38】 次の CPL 命令を実行した後の FR の値を 3 桁の 2 進数で答えよ。

```
LD      GR3, X
CPL     GR3, Y      000
}
X       DC      #000D
Y       DC      #000B
```

【39】 次の CPL 命令を実行した後の FR の値を 3 桁の 2 進数で答えよ。

```
LD      GR3, X
CPL     GR3, Y      000
}
X       DC      #FFFF
Y       DC      #000A
```

【40】 次の CPL 命令を実行した後の FR の値を 3 桁の 2 進数で答えよ。

```
LD      GR3, X
CPL     GR3, Y      001
}
X       DC      #8000
Y       DC      #8000
```

【41】 次の CPL 命令を実行した後の FR の値を 3 桁の 2 進数で答えよ。

```
LD      GR3, X
CPL     GR3, Y      010
}
X       DC      #7FFF
Y       DC      #8001
```

【42】 CPA 命令と CPL 命令を実行した後の FR の値を 3 桁の 2 進数で答えよ。

```
LD      GR0, SU1
CPA     GR0, SU2      000
CPL     GR0, SU2      000
}
SU1     DC      #0010
SU2     DC      #000F
```

【43】 CPA 命令と CPL 命令を実行した後の FR の値を 3 桁の 2 進数で答えよ。

```
LD      GR0, SU1
CPA     GR0, SU2      000
CPL     GR0, SU2      010
}
SU1     DC      #0000
SU2     DC      #8000
```

【44】 CPA 命令と CPL 命令を実行した後の FR の値を 3 桁の 2 進数で答えよ。

```
LD      GR0, SU1
CPA     GR0, SU2      010
CPL     GR0, SU2      000
}
SU1     DC      #8000
SU2     DC      #7FFF
```

【45】 CPA 命令と CPL 命令を実行した後の FR の値を 3 桁の 2 進数で答えよ。

```
LD      GR0, SU1
CPA     GR0, SU2      000
CPL     GR0, SU2      010
}
SU1     DC      32767
SU2     DC      -1
```

【46】 CPA 命令と CPL 命令を実行した後の FR の値を 3 桁の 2 進数で答えよ。

```
LD      GR0, SU1
CPA     GR0, SU2      001
CPL     GR0, SU2      001
}
SU1     DC      -32768
SU2     DC      #8000
```

【47】 次のコーディングは誤っている。その理由を説明し、正しくコーディングせよ。

```
CPL    GR0, 'A'          ;GR0 の内容と 'A' を比較する
      = 'A'
```

【48】 データの入力に処理の流れを戻すコーディングをせよ。

```
INPUT  IN    BUF, LEN
      <
      JUMP   INPUT
```

【49】 GR1 から 1 を引いて GR1 > 0 であれば LOOP 番地へジャンプするコーディングをせよ。

```
SUBA   GR1, =1
      JPL   LOOP
      <
```

【50】 次のプログラムは KOKU+SUGK+EIGO の計算をして合計が 180 点以上であれば GR2 に '\*' を設定しそうでなければ ' ' を設定するものである。空欄を埋めよ。

```
LD     GR0, KOKU
ADDA   GR0, SUGK
ADDA   GR0, EIGO
CPA    GR0, C179
JPL    GOOD
LD     GR2, SPACE
      <
GOOD   LD     GR2, AST
      <
KOKU   DS     1
SUGK   DS     1
EIGO   DS     1
C179   DC     179
AST    DC     '*'
SPACE  DC     ' '
```

【51】 次のプログラムは HEX 番地に与えられた値が 15 以下かを判定し、範囲外であれば ERR 番地へジャンプする。空欄を埋めよ。

```
LD      GR3, 
CPL     GR3, 
JPL     ERR
{
ERR
{
HEX     DS      1
MAX     DC      
```

【52】 次のプログラムは HEX 番地に 0～15 で与えられる 2 進数を 16 進数の文字 '0'～'F' に変換して CHAR 番地に格納するものである。空欄を埋めよ。

```
LD      GR1, HEX
ADDL    GR1, 
CPL     GR1, MAX
 OK
ADDL    GR1, OFFSET
OK      ST      GR1, CHAR
{
HEX     DS      1
CHAR    DS      1
ZERO    DC      '0'
MAX     DC      #003A
OFFSET  DC      
```

【53】 次のプログラムは CHAR 番地に格納されている文字 '0' ~ 'F' を数値の 0 から 15 に変換するものである。範囲外の場合には ERR 番地へジャンプする。空欄を埋めよ。

```

LD      GR2, CHAR
SUBL   GR2, ='0'
JMI    ERR
CPL    GR2, =10
JMI    OK
SUBL   GR2, =7
CPL    GR2, =10
JMI    ERR
CPL    GR2, =15
JPL    ERR
OK     ST      GR2, HEX
      >
ERR
      >
CHAR  DS      1

```

【54】 次のプログラムは ERRSW=1 のとき 'ERROR' の文字を出力し、ERRSW=0 のときは文字を出力しない。空欄を埋めよ。

```

LD      GR0, ERRSW
ADDA   GR0, C0
JZE    NOERR
OUT    ERRMES, LEN5
NOERR  RET
C0     DC      0
C1     DC      1
ERRSW  DS      1
LEN5   DC      5
ERRMES DC      'ERROR'

```

【55】 次のプログラムは ERRSW=1 のとき 'ERROR' の文字を出力し、ERRSW=0 のときは文字を出力しない。空欄を埋めよ。

```
LD      GR0, ERRSW
CPA     GR0, C0
JZE     NOERR
OUT     ERRMES, LEN5
NOERR   RET
C0      DC      0
C1      DC      1
ERRSW   DS      1
LEN5    DC      5
ERRMES  DC      'ERROR'
```

【56】 次のプログラムは ERRSW=1 のとき 'ERROR' の文字を出力し、ERRSW=0 のときは文字を出力しない。空欄を埋めよ。

```
LD      GR0, ERRSW
CPA     GR0, C1
JNZ     NOERR
OUT     ERRMES, LEN5
NOERR   RET
C0      DC      0
C1      DC      1
ERRSW   DS      1
LEN5    DC      5
ERRMES  DC      'ERROR'
```

【57】 次のプログラムは CHAR 番地に格納されている文字が英小文字であれば英大文字に変換するものである。空欄を埋めよ。

```
LD      GR0, CHAR
CPL     GR0, LMIN
JMI     SFTQ
CPL     GR0, LMAX
JPL     SFTQ
SUBL    GR0, OFFSET
ST      GR0, CHAR
SFTQ    RET
CHAR    DS      1
LMIN    DC      'a'
LMAX    DC      'z'
OFFSET  DC      32 又は #0020
```

【58】 次のプログラムは GR0=0 のときラベル SYORI1 へ、GR0=1 のときラベル SYORI2 へ、GR0=2 のときラベル SYORI3 へそれぞれジャンプするものである。空欄を埋めよ。

```
CPA    GR0, CONST
 SYORI1
 SYORI2
 SYORI3
    <
CONST  DC    
```

【59】 次の命令を実行した後のレジスタの内容を 4 桁の 16 進数で答えよ。

```
LAD    GR0, 0          #0000
LAD    GR1, 10         #000A
LAD    GR2, -1        #FFFF
LAD    GR3, 65535     #FFFF
```

【60】 次の LAD 命令を実行した後のレジスタの内容を 10 進数で答えよ。

```
LD     GR1, =8
LAD    GR1, 2, GR1     10
LD     GR3, =65526
LAD    GR3, 10, GR3   0
LD     GR2, =65530
LAD    GR2, 10, GR2   4
LD     GR1, =8
LD     GR2, =100
LAD    GR2, 10, GR1   18
```

【61】 次の LAD 命令を実行した後のレジスタの内容を符号付きの 10 進数で答えよ。

```

LD      GR1, =8
LAD     GR1, -1, GR1      7
LD      GR2, =22
LAD     GR2, -2, GR2     20
LD      GR3, =2
LAD     GR3, -10, GR3    -8
LD      GR2, =9
LD      GR3, =100
LAD     GR3, -10, GR2    -1
LD      GR2, =10
LD      GR3, =100
LAD     GR3, -10, GR2     0
LD      GR3, =9
LAD     GR3, 65535, GR3   8
LD      GR2, =7
LAD     GR2, 65533, GR2   4

```

【62】 次の処理を LAD 命令を使ってコーディングせよ。

```

GR0 の内容を 0 にする          LAD  GR0, 0
GR1 の内容を-1 にする         LAD  GR1, -1
GR2 に 1 を加える             LAD  GR2, 1, GR2
GR3 から 1 を引く            LAD  GR3, -1, GR3
GR2 から 10 を引いて GR3 に設定する LAD  GR3, -10, GR2
GR3 の内容を GR0 に転送する   LAD  GR0, 0, GR3

```

【63】 次のプログラムは SU 番地の内容を 10 倍して SU10 番地に格納するものである。空欄を埋めよ。

```

P063  START
      LAD  GR0, 0
      LAD  GR2,  ;10 回繰り返す
LOOP  ADDA  GR0, SU
      SUBA   ;回数から 1 引く
       LOOP
      ST   GR0, SU10
      RET
SU    DS   1
SU10  DS   1
      END

```

【64】 次のプログラムは 10 から 1 までの和を求めるものである。空欄を埋めよ。

```
P064  START
      LAD  GR0, 0
      LAD  GR3,  ;10 から始める
LOOP  ADDA  
      SUBA   ;回数から 1 引く
       LOOP
      ST   
      RET
WA    DS   1
      END
```

【65】 ある処理を 40 回繰り返したい。処理の中で 0 の値も使いたい。空欄を埋めよ。

```
      LAD  GR2, 
LOOP  処理
      〈
      SUBA  GR2, =1
       LOOP
      JZE  LOOP
NEXT  〈
```

【66】 次のプログラムは 1 から 10 までの和を求めるものである。空欄を埋めよ。

```
P066  START
      LAD  GR0, 0
      LAD  GR1,  ;初期設定
AGN   LAD  GR1, 1, GR1
      ADDA  GR0, GR1
      CPA  GR1, 
       AGN
      ST   GR0, WA
      RET
C10  
WA   DS   1
      END
```

【67】 次のプログラムを実行したときに出力される様子を示せ。

```
P067  START
      LAD  GR2, 0                      1
PRN1  LAD  GR2, 1, GR2                 12
      ST   GR2, LEN                    123
      OUT  STR, LEN                     1234
      CPA  GR2, MAX                     12345
      JMI  PRN1                         123456
      RET                               1234567
MAX   DC   10                          12345678
LEN   DS   1                            123456789
STR   DC   '123456789ABCDEF'          123456789A
      END
```

【68】 次のプログラムは秒数(SECOND)を分(MINUTE)と秒(SECOND)に変換するものである。

空欄を埋めよ。

```
      LD   GR2, SECOND
      LAD  GR1, 0
AGN   CPA  
       ANS
      SUBA 
      LAD  GR1, 1, GR1
      JUMP AGN
ANS   ST   
      ST   GR2, SECOND
      }
C60   DC   60
SECOND DS  1
MINUTE DS  1
```

【69】次は A 番地の内容に B 番地の内容を掛けて積を C 番地に格納するプログラムであるが、加算の回数を減らすように工夫したものである。空欄を埋めよ。

```

LD      GR0, A
CPA    GR0, B
JPL    NXT
LD      GR1, B
ST      GR0, B
ST      GR1, A
NXT    LAD   GR1, 0
LD      GR2, B
JZE    OWR
MUL    ADDA  GR1, A
SUBA   GR2, =1
JNZ    MUL
OWR    ST    GR1, C
      }
A      DS   1
B      DS   1
C      DS   1

```

【70】次のプログラムは A 番地の内容が B 番地の内容で割り切れるときは FLG 番地に 0 を、割り切れないときは 1 を格納するものである。空欄を埋めよ。

```

LD      GR1, B
JZE    MODE          ;B=0 ならエラー
LD      GR1, A
MOD1   SUBA  GR1, B
JZE    MODQ
JPL    MOD1
MODE   LAD   GR1, 1
MODQ   ST    GR1, FLG
      }
A      DS   1
B      DS   1
FLG    DS   1

```

【71】 次の命令を実行した後の各レジスタの内容を 10 進数で答えよ。

	LD	GR1, =4	4
	LD	GR0, TBL, GR1	11
	LD	GR1, =5	5
	LD	GR3, =3	3
	ADDA	GR1, TBL, GR3	12
	LD	GR2, =0	0
	LD	GR3, =10	10
	SUBA	GR3, TBL, GR2	8
		{	
TBL	DC	2	
	DC	3	
	DC	5	
	DC	7	
	DC	11	
	DC	13	

【72】 次のプログラムで GR1 に値が設定された後に GR0 に設定される文字をそれぞれ答えよ。

	LD	GR1, =3	
	LD	GR0, HTBL, GR1	'3'
	LD	GR1, =0	
	LD	GR0, HTBL, GR1	'0'
	LD	GR1, =15	
	LD	GR0, HTBL, GR1	'F'
	LD	GR1, =1	
	LD	GR0, HTBL, GR1	'1'
	LD	GR1, =18	
	LD	GR0, HTBL, GR1	'M'
		{	
HTBL	DC	'0123456789ABCDEF'	
MID	DC	'COMET & CASL'	

【73】 次のプログラムは OBUF 番地から連続して 20 語確保された領域をスペースクリアするものである。空欄を埋めよ。

```

                LD      GR0, SPACE
                LAD     GR1, 
LOOP          ST      GR0, 
                SUBA   GR1, =1
                  OWARI
                 LOOP
OWARI        RET
OBUF         DS      20
SPACE        DC      ' '

```

【74】 次のプログラムは TEN 番地に格納されている 0 点～10 点までの得点に対応する個数の '\*' を GRAP 番地から連続して格納するものである。空欄を埋めよ。

```

                LAD     GR0, 32          ; ' '
                LAD     GR1, 
CLR          ST      GR0, GRAP, GR1
                SUBA   
                  NEXT
                 CLR
NEXT        LAD     GR0, 42          ; '*'
                LAD     GR1, 0
AST         CPA     GR1, TEN
                  OWR
                ST      
                LAD     GR1, 1, GR1
                JUMP   AST
OWR
                {
TEN         DS      1
GRAP        DS      10

```

【75】 次のプログラムは表として確保されている TBLA、TBLB のそれぞれの和を TBLC に格納するものである。空欄を埋めよ。

```

          LAD      GR2, 0
WA      LD        GR0, TBLA, GR2
          ADDA     GR0, TBLB, GR2
          ST       GR0, TBLC, GR2
          LAD      GR2, 1, GR2
          CPA      GR2, =10
          JNZ     WA
          ~
TBLA    DS        10
TBLB    DS        10
TBLC    DS        10

```

【76】 次のプログラムは月別の売上金額合計の当該月に今月の売上金額(KINGK)を加算するものである。空欄を埋めよ。

```

          LD        GR1, TUKI
          LAD      GR1, -1, GR1 ;月数からオフセットを計算
          LD        GR0, GKTBL, GR1
          ADDA     GR0, KINGK
          ST       GR0, GKTBL, GR1
          }
KINGK   DS        1
TUKI    DS        1
GKTBL   DS        12

```

【77】 次のプログラムは DAT 番地から連続した 100 語に格納されている数値の最大値を求め、その値を MAX 番地に、その位置を MXP 番地に格納するものである。空欄を埋めよ。

```

LD      GR0, MAX
LD      GR1, MXP
LOP    CPA  GR0, 
  NXT
LD      GR0, 
ST      GR0, 
ST      GR1, 
NXT    LAD  GR1, 
CPA    GR1, 
  LOP
      <
MAX    DC  
MXP    DC  0
DAT    DS  100

```

【78】 次のプログラムは COD 番地に格納されたコードの単価をテーブルサーチによって求め、TNK 番地に格納するものである。該当するコードがなければ 0 を格納する。空欄を埋めよ。

```

LD      GR0, COD
LAD     GR1, 0
CMP     CPA      GR0, CTBL, GR1
JMI     ERR
JZE     FND
LAD     GR1, 1, GR1
CPA     GR1, =5
JMI     CMP
ERR     LAD      GR3, 0
JUMP    STR
FND     LD       GR3, TTBL, GR1
STR     ST       GR3, TNK
      <
COD     DS      1
TNK     DS      1
CTBL    DC      101
        DC      118
        DC      133
        DC      204
        DC      226
TTBL    DC      100
        DC      180
        DC      250
        DC      300
        DC      400

```

【79】 次のプログラムは HTBL 中の 'F' の文字を GR0 に設定するものである。空欄を埋めよ。

```

LAD     GR1, 15
      LAD      GR1, HTBL, GR1
LD      GR0, 0, GR1
      <
HTBL    DC      '0123456789ABCDEF'

```

【80】 次のプログラムは INBUF の内容を先頭から検索し、スペース以外の文字、または 0 が最初に現れた位置の絶対アドレスを ADR に求めるものである。空欄を埋めよ。

```

        LAD      GR1, INBUF
TOPG    LD       GR2, 
        JZE      TOPQ
        CPA      GR2, SPACE
        JNZ      TOPQ
        LAD      
        JUMP     TOPG
TOPQ    ST       
        }
LEN     DS      1
ADR     DS      1
INBUF   DS      81
SPACE   DC      ' '

```

【81】 次のプログラムを実行した後の A 番地、B 番地および GR1 の内容を答えよ。

```

        LD       GR1, A
        AND      GR1, B          #1234
        }
A       DC      #B2F6          #B2F6
B       DC      #5E3C          #5E3C

```

【82】 次のプログラムを実行した後の GR3 の内容を答えよ。

```

        LD       GR3, CHAR
        OR       GR3, LOWR      #0061
        }
CHAR    DC      'A'
LOWR    DC      #0020

```

【83】 次のプログラムを実行した後の GR1 の内容を答えよ。

```

        LD       GR1, A
        XOR      GR1, B          #AAAA
        }
A       DC      #F4C7
B       DC      #5E6D

```

【84】 次のプログラムの各命令を実行した後の GR3 の内容をそれぞれ答えよ。

```
LD      GR3, DAT      #000A
XOR     GR3, DAT      #0000
XOR     GR3, DAT      #000A
  }
DAT     DC      #000A
```

【85】 次のプログラムを実行した後の E1 番地、E2 番地の内容を 16 進数で答えよ。

```
LD      GR0, E1
XOR     GR0, E2
ST      GR0, E1
XOR     GR0, E2
ST      GR0, E2
XOR     GR0, E1
ST      GR0, E1
  }
E1     DC      #ABCD      #1234
E2     DC      #1234      #ABCD
```

【86】 次のプログラムを実行した後の GR2 の内容をそれぞれ答えよ。

```
LD      GR2, DAT1
XOR     GR2, FFFF      #3CF0
  }
DAT1    DC      #C30F
FFFF    DC      #FFFF
```

【87】 次のプログラムの各 AND 命令を実行した後の GR0 の内容をそれぞれ答えよ。

```
LD      GR0, A
AND     GR0, C1      #A000
LD      GR0, A
AND     GR0, C2      #0B00
LD      GR0, A
AND     GR0, C3      #00C0
LD      GR0, A
AND     GR0, C4      #000C
{
A       DC      #ABCD
C1      DC      #F000
C2      DC      #0F00
C3      DC      #00F0
C4      DC      #000E
```

【88】 次のビット列を取り出すためのマスクデータを 4 桁の 16 進数で答えよ。

```
最上位ビットを取り出す      #8000
文字'0'～'9'を数値の 0～9 に変換する  #000F
英小文字を英大文字に変換する  #005F
```

【89】 次のプログラムは SUU 番地に格納されている数値が 4 の倍数であれば BAI4 番地へジャンプするものである。空欄を埋めよ。

```
LD      GR3, SUU
AND     GR3, MASK
JZE     BAI4
{
BAI4
{
SUU     DS      1
MASK    DC      #0003
```

【90】 次のプログラムは N 番地に格納されている値の 2 の補数を求めて NEG 番地に格納するものである。空欄を埋めよ。

```
LD      GR1, N
XOR     GR1, MSKF
LAD     
ST      GR1, NEG
        <
N       DS      1
NEG     DS      1
MSKF    DC      
```

【91】 次のプログラムは SU 番地に格納されている数の絶対値を ABS 番地に格納するものである。空欄を埋めよ。

```
LD      GR2, SU
 STR
 GR2, MSKF
LAD     
STR     ST      GR2, ABS
        <
SU      DS      1
ABS     DS      1
MSKF    DC      #FFFF
```

【92】 次のプログラムは GR0 の内容を 10 倍するものである。空欄を埋めよ。

```

ST      GR0, WK

ADDA   GR0, WK
```

【93】 次のプログラムは重さの単位を貫から kg に変換するものである。空欄を埋めよ。

1 貫 = 3.75kg である。

```
LD      GR0, KAN
SRA     GR0, 2
ST      GR0, KG
LD      GR0, KAN
SLA     GR0, 2
SUBA    GR0, KG
ST      GR0, KG
{
KAN     DS      1
KG      DS      1
```

【94】 次のプログラムは重さの単位を貫から kg に変換するものである。空欄を埋めよ。

上の問題とは別の方法で変換している。

```
LD      GR1, KAN
SLA     GR1, 1
LD      GR2, KAN
LAD     GR3, 3
LOP     ADDA    GR1, GR2
SRA     GR2, 1
SUBA    GR3, =1
JNZ     LOP
ST      GR1, KG
{
KAN     DS      1
KG      DS      1
```

【95】 次のプログラムを実行した後の AB 番地の値を 10 進数で答えよ。

```
LD      GR0, N
SRA     GR0, 15
ST      GR0, AB
XOR     GR0, N
SUBA    GR0, AB
ST      GR0, AB
{
N       DC      15
AB      DS      1      15
```

【96】 次のプログラムを実行した後の AB 番地の値を 10 進数で答えよ。

```
LD      GR0, N
SRA     GR0, 15
ST      GR0, AB
XOR     GR0, N
SUBA    GR0, AB
ST      GR0, AB
}
N      DC      -15
AB     DS      1
```

15

【97】 次のプログラムは GR1 に設定されているデータの 1 のビットの個数を GR2 に設定するものである。空欄を埋めよ。

```
LAD     GR2, 0
CNT1    ST     GR1, WRK
        AND    GR1, MSK1
        ADDA   GR2, GR1
        LD     GR1, WRK
        SRL   GR1, 1
        JNZ   CNT1
}
WRK     DS     1
BIT     DS     1
MSK1    DC     #0001
```

【98】 次のプログラムは GR1 に設定されているデータの 1 のビットの個数を GR2 に設定するものである。空欄を埋めよ。

```
LAD     GR2, 0
SLL     GR1, 0
LOP1    JZE   OWARI
        JPL   LOP2
        LAD   GR2, 1, GR2
LOP2    SLL   GR1, 1
        JUMP  LOP1
}
OWARI
```

【99】 次のプログラムは DAT 番地の内容の上位 8 ビットと下位 8 ビットを交換した値を CHG 番地に格納するものである。空欄を埋めよ。

```

LD      GR1, DAT
SRL    GR1, 
ST      GR1, WRK
LD      GR1, DAT
SLL    GR1, 8
OR      GR1, 
ST      GR1, CHG
      }
WRK    DS      1
DAT    DS      1
CHG    DS      1

```

【100】 次のプログラムは DAT 番地の内容を上位から 4 ビットずつ区切り、順位を逆転して INV 番地に格納するものである。空欄を埋めよ。

```

LAD     GR1, 0
LAD     GR2, 0
LAD     GR3, 4
LOOP   LD      GR0, DAT
SRL    GR0, 
AND    GR0, MASK
ST      GR0, INV
OR      GR2, INV
SUBA   
JZE    FNSH
SLL    GR2, 
LAD     
JUMP   LOOP
FNSH   ST      GR2, INV
      }
DAT    DS      1
INV    DS      1
MASK   DC      

```