```
BSA         GetPARAMETERS
HLT
GetPARAMETERS,HEX 0
            LDA   POWERPTR1
            BSA   PRINT
InputNUM1,  SKI
            BUN   InputNUM1
            INP
Echo1,      SKO
            BUN   Echo1
            OUT         //START INPUT FROM THE USER
            STA   CC
            CMA
            INC
            ADD   MINUS       //IF(CC='-')
            SZA
            BUN   POSITIVE
THEN1,      ISZ   SIGN1//sign1=1; the first is negative
            BUN   CONTINUE
POSITIVE,   LDA   PLUS// if I got plus
            CMA
            INC
            ADD   CC
            SZA
            BUN   CHECK1
THEN2,      BUN   CONTINUE
CONTINUE,   SKI
            BUN   CONTINUE
            INP
Echo2,      SKO
            BUN   Echo2
            OUT
            STA   CC
            BUN   CHECK1
CHECK1,     LDA   MINHEX            //CHECK ITS NOT LETTERS
            CMA
            INC
            ADD   CC
```

```
IF(CC=='-')
        SIGN1++;
IF(29<CC<40)
        NUM*=10;
        NUM+=CC;
        CONTINUE;
ELSE
        STOP;
```

```
                SPA                             //-29+CC>0
                BUN     FINAL1
THEN3,          LDA     MAXHEX
                CMA
                INC
                ADD     CC
                SNA             //IF(CC-40<0)
                BUN     FINAL1
THEN4,          LDA     TNUM1           //PUT CC IN NUM AND MITIFCATION IN 10
                CIL
                CIL
                CIL
                ADD     TNUM1
                ADD     TNUM1
                ADD     CC
                ADD     MINUS30
                STA     TNUM1
                LDA     MAXNUM          // check overflow\underflow
                CMA
                INC
                ADD     TNUM1
                SNA
                BUN     TRYAGAIN
                BUN     CONTINUE
FINAL1,         LDA     SIGN1
                SPA
                BUN     CONTINUE3
THEN5,          LDA     TNUM1
                CMA
                INC
                STA     TNUM1

CONTINUE3,      LDA     MINUS
                CMA
                INC
                ADD     CC
                SZA
                BUN     ELSE1
THEN6,          LDA     MINUS
```

```
                STA     OP
                ISZ     SIGN2
                BUN     CONTINUE2
ELSE1,          LDA     PLUS
                CMA
                INC
                ADD     CC
                SZA
                BUN     ELSE2
THEN7,          LDA     PLUS
                STA     OP
                BUN     CONTINUE2
ELSE2,          LDA     DIV
                CMA
                INC
                ADD     CC
                SZA
                BUN     ELSE3
THEN8,          LDA     DIV
                STA     OP
                BUN     CONTINUE2
ELSE3,          LDA     MULT
                CMA
                INC
                ADD     CC
                SZA
                BUN     TRYAGAIN
THEN9,          LDA     MULT
                STA     OP
                BUN     CONTINUE2
CONTINUE2,      SKI
                BUN     CONTINUE2
                INP
Echo3,          SKO
                BUN     Echo3
                OUT
                STA     CC
                LDA     EQ
                CMA
```

```
                INC
                ADD     CC      //      if(cc=='=')
                SZA
                BUN     ELSE4
                BUN     FINAL2
ELSE4,  LDA     MINUS
                CMA
                INC
                ADD     CC
                SZA
                BUN     CHECKNUM
OK,             LDA     SIGN2
MINUSMINUS,SZA
                BUN     ISPLUS
K,              ISZ     SIGN2
                BUN     CONTINUE2
ISPLUS,         LDA     ZERO
                STA     SIGN2
                BUN     CONTINUE2
CHECKNUM ,  LDA     MINHEX      //CHECK ITS NOT LETTERS
                CMA
                INC
                ADD     CC
                SPA             //-29+CC>0
                BUN     TRYAGAIN
THEN10,         LDA     MAXHEX
                CMA
                INC
                ADD     CC
IF9,            SNA                         //IF(CC-40<0)
                BUN     TRYAGAIN
THEN11,         LDA     TNUM2       //PUT CC IN NUM AND MITIFCATION IN 10
                CIL
                CIL
                CIL
                ADD     TNUM2
                ADD     TNUM2
                ADD     CC
                ADD     MINUS30
```

```
                STA     TNUM2
                LDA     MAXNUM      // check overflow\underflow
                CMA
                INC
                ADD     TNUM2
                SNA
                BUN     TRYAGAIN
                BUN     CONTINUE2
FINAL2,         LDA     SIGN2
                SPA
                BUN     Operation
THEN12,         LDA     TNUM2
                CMA
                INC
                STA     TNUM2
Operation,      LDA     MINUS
                CMA
                INC
                ADD     OP
                SZA                 //IF(OP=='-')
                BUN     ELSE5
THEN13,         BUN     PLUS\MINUS
ELSE5,  LDA     PLUS        //IF(OP=='+')
                CMA
                INC
                ADD     OP
                SZA
                BUN     ELSE6
THEN14,         BUN     PLUS\MINUS
ELSE6,  LDA     MULT        //IF(OP=='*')
                CMA
                INC
                ADD     OP
                SZA
                BUN     CHECK2
THEN15,         LDA     TNUM1       //TURN THE NUMBERS TO POSITIVE
                SNA
                BUN     CHECK3
ONENEGATIVE1,CMA
```

```
                INC
                STA     TNUM1
CHECK3,         LDA     TNUM2
                SNA
                BUN     MULTIFICATION
TWONEGATIVE1,CMA
                INC
                STA     TNUM2
                BUN     MULTIFICATION

CHECK2,         LDA     DIV     //IF(OP=='/')
                CMA
                INC
                ADD     OP
                SZA
                BUN     TRYAGAIN
THEN16,         CLE
                LDA     TNUM1           //TURN THE NUMBERS TO POSITIVE
                SNA                     //IF(TNUM1<0)
                BUN     CHECK4
ONENEGATIVE2,CMA
                INC
                STA     TNUM1           //TNUM1*=-1;
CHECK4,         CLE
                LDA     TNUM2           //IF(TNUM2*=-1)
                SNA
                BUN     DIV_FUNC
TWONEGATIVE2,CMA
                INC
                STA     TNUM2           //TNUM2*=-1;
                BUN     DIV_FUNC
//-------------------------------------------------------------------------------

PLUS\MINUS, LDA     TNUM1
                ADD     TNUM2
                STA     RESULT          //result=tnum1+tnum2;
                BUN     OVER\UNDERFLOW
```

```
//-------------------------------------------------------------------------------

MULTIFICATION,CLE
              LDA     Bits
              CMA
              INC
              STA     MinusDigits     // MinusDigits = -Bits;
              CLA
              STA     RESULT          // MultResult = 0;
              STA     DigitCount      // DigitCount=0;
FOR_loop2,    LDA     DigitCount      // WHILE (DigitCount < Bits)
              ADD     MinusDigits
              SZA
              BUN     Body2
              BUN     End_loop3
Body2,        LDA     TNUM2           // DO
              CIR                     //      E = shift_right(multiplier)
              STA     TNUM2
              SZE                     //      IF (E==1)
              BUN     THEN17
              BUN     FI
THEN17,       LDA     RESULT          //THEN Result = Result + multiplicand;
              ADD     TNUM1
              STA     RESULT
FI,           LDA     TNUM1           //      FI;
              CIL                     //      Shift(multiplicand) 1 place to left;
              STA     TNUM1
              ISZ     DigitCount      //      DigitCount++;
              BUN     FOR_loop2       // END;
End_loop3,    LDA     RESULT
              SZE
              BUN     TRYAGAIN
THEN18,       LDA     SIGN1
              CMA
              INC
```

```
                ADD     SIGN2
                SZA
                BUN     ELSE7
THEN19,         LDA     RESULT
                BSA     PutSignedInt
                BUN     RESET
ELSE7,  LDA     RESULT
                CMA
                INC
                BSA     PutSignedInt
                BUN     RESET
//-------------------------------------------------------------------------------------------------
DIV_FUNC,       CLE
                LDA     TNUM2           //NOT WORKING
                SZA     //if (tnum2==0)
                BUN     WHILE11
                BUN     TRYAGAIN
WHILE11,        LDA     TNUM1//IF (TNUM1==0)//DEVIDE 8 BITS IN 4 BITS
                SZA
                BUN     CHECK5
                BUN     END2
CHECK5,         LDA     TNUM2           //IF(TNUM2==1)
                CMA
                INC
                ADD     ONE
                SZA
                BUN     CHECK6
END1,           LDA     TNUM1//         RESULT+=TNUM1;
                ADD     RESULT
```

```
                STA     RESULT
                BUN     END2
CHECK6,         LDA     TNUM1           //WHILE(TNUM1>0)
                SPA
                BUN     END2
DO2,            LDA     TNUM2
                CIR     //      IF(TNUM2%2==0)
                SZE
                BUN     ELSE8
THEN20,         LDA     TNUM1
                CIR
                SZE             //      IF(TNUM1%2==0)
                BUN     ELSE8
THEN21,         STA     TNUM1           //      TNUM1/=10;
                LDA     TNUM2
                CIR
                STA     TNUM2
                ISZ     RESULT
                BUN     WHILE11
ELSE8,          LDA     TNUM2//TNUM2 NOT DEVIDE IN 2
                CMA
                INC
                ADD             TNUM1//IF(TNUM1-TNUM2<0)
                SNA
                BUN     ELSE10
THEN22,         LDA     TNUM1
                ADD     Fraction//FRACTION+=TNUM1
                STA     Fraction
                BUN     END2
```

```
ELSE10,     STA     TNUM1//IF(TNUM1-TNUM2>0)TNUM1-=TNUM2;
            ISZ     RESULT//RESULT++;
            BUN     WHILE11
END2,       LDA     SIGN1
            CMA
            INC
            ADD     SIGN2
            SZA
            BUN     ELSE15
THEN27,     LDA     RESULT
            BSA     PutSignedInt
            LDA     Fraction
            SZA
            BUN     ELSE16
            BUN     RESET
ELSE16,     LDA     OPEN
            BSA     Putc
            LDA     Fraction
            BSA     PutSignedInt
            LDA     CLOSE
            BSA     Putc
            BUN     RESET
ELSE15,     LDA     RESULT
            CMA
            INC
            STA     RESULT
            BSA     PutSignedInt
            LDA     Fraction
            SZA
```

```
                BUN     ELSE17
                BUN     RESET
ELSE17,         LDA     OPEN
                BSA     Putc
                LDA     Fraction
                BSA     PutSignedInt
                LDA     CLOSE
                BSA     Putc
                BUN     RESET

//-------------------------------------------------------------------------

OVER\UNDERFLOW,SNA                  //if(result<0)
                BUN     IF51
THEN25,         CMA
                INC
IF51,           ADD     MINUSMAXNUM     //checkoverflow
                SPA
                BUN     ELSE13
                BUN     TRYAGAIN
ELSE13,         SZA
                BUN     ELSE14
THEN26,         BUN     TRYAGAIN
ELSE14,         LDA     RESULT
                BSA     PutSignedInt
                BUN     RESET
//-----------------------------------------------------------------------------------

TRYAGAIN,       LDA     ERROR //GO TO THE BEGINNING AND GIVE US ANOTHER
CHANCE
                BSA     PRINT
```

```
                BUN     RESET
//-------------------------------------------------------------------------
RESET,          LDA     ZERO
                STA     TNUM1
                STA     SIGN1
                STA     TNUM2
                STA     SIGN2
                STA     RESULT
                STA     Fraction
                LDA     NEXT
                BSA     PRINT
                BUN     InputNUM1



//DATA GetPARAMETERS
Fraction,       DEC     0
MINUSONE,       DEC     -1
ONE,            DEC     1
RESULT,         HEX     0
ZERO,           DEC     0
EQ,             DEC     61
MULT,           HEX     2A
DIV,            HEX     2F
PLUS,           HEX     2B
OP,             HEX     0
CC,             HEX     0
TNUM2,          HEX     0
TNUM1,          HEX     0
MINUS,          HEX     2D
```

```
SIGN1,          HEX   0
SIGN2,          HEX   0
MINUS30,        HEX   -30
MINHEX,         HEX   2F
MAXHEX,         HEX   3A
DigitCount,     DEC   0
Bits,           DEC   8       // we use only 8 bits from 16
MinusDigits,    DEC   0

ERROR,          HEX   500
                ORG   500
Array,          HEX   0A
                HEX   45
                HEX   52
                HEX   52
                HEX   4F
                HEX   52
                HEX   D
                HEX   54
                HEX   72
                HEX   79
                HEX   20
                HEX   61
                HEX   67
                HEX   61
                HEX   69
                HEX   6E
                HEX   0
OPEN,           HEX   28
CLOSE,          HEX   29

NEXT,           HEX   600
                ORG   600
ARR,            HEX   D
                HEX   3E
                HEX   0
```

```
MINUSMAXNUM,HEX -7FFF
MAXNUM,     HEX     7FFF

POWERPTR1,  HEX         550
            ORG         620
HELLOARRAY, HEX         57          // "Hello. Use these keys to
            //HEX       65          //   calculate: + - *
            //HEX       65
            //HEX       6C
            //HEX       63
            //HEX       6F
            //HEX       6D
            //HEX       65
            //HEX       20
            //HEX       74
            //HEX       6F
            //HEX       20
            //HEX       6F
            //HEX       75
            //HEX       72
            //HEX       20
            //HEX       70
            //HEX       6F
            //HEX       63
            //HEX       6B
            //HEX       65
            //HEX       74
            //HEX       20
            //HEX       63
            //HEX       61
```

```
                //HEX       6C
                //HEX       63
                //HEX       75
                //HEX       6C
                //HEX       61
                //HEX       74
                //HEX       6F
                //HEX       72
                //HEX       21
                //HEX       0D
////
                HEX       53
                HEX       65
                HEX       6C
                HEX       65
                HEX       63
                HEX       74
                HEX       20
                HEX       61
                HEX       6E
                HEX       20
                HEX       61
                HEX       63
                HEX       74
                HEX       69
                HEX       6F
                HEX       6E
                HEX       3A
                HEX       20
```

```
        HEX     2B
        HEX     20
        HEX     2D
        HEX     20
        HEX     2A
        HEX     20
        //HEX   2F
        //HEX   20
        HEX     0D          //LINE BREAK
        HEX     46          //"for result press ="
        //HEX   6F
        //HEX   72
        //HEX   20
        //HEX   72
        //HEX   65
        //HEX   73
        //HEX   75
        //HEX   6C
        //HEX   74
//      HEX     20
//      HEX     70
//      HEX     72
//      HEX     65
//      HEX     73
//      HEX     73
        //HEX   20
//      HEX     3D
//      HEX     0D
//      HEX     50          //"press numbers between
```

```
//      HEX         72          //0 to 9"
//      HEX         65
//      HEX         73
//      HEX         73
//      HEX         20
//      HEX         6E
//      HEX         75
//      HEX         6D
//      HEX         62
//      HEX         65
//      HEX         72
//      HEX         20
//      HEX         62
//      HEX         65
//      HEX         74
//      HEX         77
//      HEX         65
//      HEX         65
//      HEX         6E
//      HEX         20
//      HEX         30
//      HEX         20
//      HEX         74
//      HEX         6F
//      HEX         20
//      HEX         39
//      HEX         20
//      HEX         0D
//      HEX         54          //"to stop press x"
```

```
        //      HEX         6F
        //      HEX         20
        //      HEX         73
        //      HEX         74
        //      HEX         6F
        //      HEX         70
        //      HEX         20
        //      HEX         70
        //      HEX         72
        //      HEX         65
        //      HEX         73
        //      HEX         73
        //      HEX         20
        //      HEX         78
        //      HEX         0D
        //      DEC         62
                HEX         20
                DEC         0
                ORG     700
PRINT,          HEX     0
                STA     PTR
While,          LDA     PTR     I
                SZA
                BUN     Body
                BUN     MESSAGEEnd
Body,           BSA     Putc
                ISZ     PTR
                BUN     While
MESSAGEEnd,     CLA
                STA     PTR
                BUN     PRINT   I
//DATAFailERROR
```

```
PTR,            HEX     0
                ORG     750
 //void putc(char cc)
Putc,           HEX     0
Char_out,       SKO
                BUN     Char_out
                OUT
                BUN     Putc I
                ORG     800
PutSignedInt,   HEX     0           //print result
                STA     Number
                LDA     Power
                STA     PowerPtr
                LDA     LLZero
                STA     flagint
                LDA     Number
                AND     Mask
                SZA                 // IF ((Number && Mask) # 0)
                BUN     MinusL
                BUN     Continue5
MinusL,         LDA     Sign_ascii  // THEN
                BSA     Puts2       //      output ("-");
                LDA     Number
                CMA
                INC
                STA     Number      // Number = Number * -1;
Continue5,      CLA
                STA     Count       // Count = 0;
Out_Loop,       LDA     Count
                ADD     Minusfour
                SNA                 // While (Count < 4)
                BUN     End_LoopL2
                CLA                 // DO
                STA     Digit       // Digit = 0;
                LDA     PowerPtr I
                STA     Divisor     // Divisor = * PowerPtr;
Dividing,       LDA     Divisor
                CMA
                INC
                ADD     Number
                SNA                 //      While (Number - Divisor   >=  0)
                BUN     Continue6   //
                BUN     Zerocheck   //
Continue6,      STA     Number      //DO    Number = Number - Divisor;
```

```
Count = 0;
Loop1,  IF (Count < 4)
        TNEN
        digit =0;
        Divisor = *Power10Ptr;
Loop2,  IF  (num - Divisor   <  0)
        THEN
            Output(digit);
            Power10Ptr ++;
            Count++;
            goto Loop1;

        ELSE
            num = num - Divisor;
            digit++;
            goto Loop2,

         ELSE goto End;
End, Output(Number);  // print the 1's  left
```

```
                ISZ     Digit           //      Digit++;
                BUN     Dividing        //      OD
Zerocheck,      LDA     Flagint
                SZA
                BUN     Endloop2
                LDA     Digit
                SZA
                BUN     BeforeEloop2
                BUN     Zero1
BeforeEloop2,   ISZ     Flagint
Endloop2,       LDA     Digit
                ADD     ascii_Offset
                BSA     Puts2           //      Output(Digit);
Zero1,          ISZ     PowerPtr        //      PowerPtr++;
                ISZ     Count           //      Count++;
                BUN     Out_Loop        // OD
End_LoopL2,     LDA     Number
                ADD     ascii_Offset
                BSA     Puts2           // Output(Number);
                BUN     putSignedInt I

//outputs a visible number on the screen
// PutSignedInt,
Number,         DEC     0
Minusfour,      DEC     -4
Count,          DEC     0
Digit,          DEC     0
Divisor,        DEC     0
Power,          HEX     850
PowerPtr,       HEX     0
Sign_ascii,     HEX     2D              // the "-" character
ascii_Offset,   HEX     30
Mask,           HEX     8000
Flagint,        DEC     0
LLZero,         DEC     0
                                        // digit to ascii representation offset
                ORG     850
Power10,        DEC     10000
                DEC     1000
                DEC     100
                DEC     10
//prints only 1 variable, prints what it receives
Puts2,          HEX     0
Output_Loop2, SKO
BUN    Output_Loop2
                OUT
                BUN     Puts2 I
```