Lab2 Report

郑浩博 3210105321

Algorithm Explanation

We use trap x20 to get input and end until `"ENTER"` ,during geting input, we transform it to decimal number :

```
;;Decimal Input -> R1
ENTER       TRAP x20
            ADD R2,R0,#-10  ;;(R2 <- R0ASC - 10)
            BRz Next
            JSR Multi10     ;;(R1 = R1 * 10)
            ADD R0,R0,R7    ;;(R0 = R0 - 48)
            ADD R1,R1,R0    ;;(R1 = R1 + R0)
            BRnzp   ENTER
```

Then we transform the number R1 form binaray to decimal, 4 bits each step:

```
;;4 bits Binary to Decimal -> R2
Next        AND R2,R2,#0    ;;(R2 = 0)
            AND R3,R3,#0    ;;(R3 = 0)
            ADD R3,R3,#4    ;;(R3 = R3 + 4)
Transform   BRz Store       ;;(if R3 == 4 - 4 == 0, break)
            ADD R1,R1,#0    ;;(R1 = R1 + 0)
            BRn Neg         ;;(if R1 < 0, goto Neg)
            ADD R2,R2,R2    ;;(R2 = R2 * 2)
            ADD R1,R1,R1    ;;(R1 << 1)
            ADD R3,R3,#-1   ;;(R3 = R3 - 1)
            BRnzp Transform
Neg         ADD R2,R2,R2    ;;(R2 = R2 * 2)
            ADD R2,R2,#1    ;;(R2 = R2 + 1)
            ADD R1,R1,R1    ;;(R1 << 1)
            ADD R3,R3,#-1   ;;(R3 = R3 - 1)
            BRnzp  Transform
```

Then we store the 4 decimal number to location x4000, x4001, x4002, x4003:

```
Store0      STI R2,loc0     ;;(Mem[x4000] <- R2)
            ADD R4,R4,#1    ;;(R4 = R4 + 1)
            BRnzp Next

Store1      ADD R4,R4,#1    ;;(R4 = R4 + 1)
            STI R2,loc1     ;;(Mem[x4000] <- R2)
            ADD R4,R4,#1    ;;(R4 = R4 + 1)
            BRnzp Next

Store2      ADD R4,R4,#2    ;;(R4 = R4 + 2)
            STI R2,loc2     ;;(Mem[x4000] <- R2)
            ADD R4,R4,#1    ;;(R4 = R4 + 1)
```

```
            BRnzp Next

Store3      ADD R4,R4,#3    ;;(R4 = R4 + 3)
            STI R2,loc3     ;;(Mem[x4000] <- R2)
            ADD R4,R4,#1    ;;(R4 = R4 + 1)
            BRnzp Output
```

We should pay attention that decimal number should be stored differently as whether it is less than 10:

```
Store       ADD R2,R2,#-10  ;;(R2 = R2 - 10)
            BRzp Char      ;;(R2 >= 0)
Number      ADD R2,R2,#10   ;;(R2 = R2 + 10)
            ADD R2,R2,#12   ;;(R2 = R2 + 12)
            ADD R2,R2,#12   ;;(R2 = R2 + 12)
            ADD R2,R2,#12   ;;(R2 = R2 + 12)
            ADD R2,R2,#12   ;;(R2 = R2 + 12)
            BRnzp   Sto
```

Finally, we give 0 to location x4004 and `PUTS` :

```
Output      AND R2,R2,#0    ;;(R2 = 0)
            STI R2,loc4     ;;(Mem[x4004] <- 0)
            LD R0,loc0     ;;(R0 <- x4000)
            TRAP x22        ;;(PUTS)
```

Question & Answer

1. Question : What the meaning of your `Neg` below:

```
Transform   BRz Store       ;;(if R3 == 4 - 4 == 0, break)
            ADD R1,R1,#0    ;;(R1 = R1 + 0)
            BRn Neg         ;;(if R1 < 0, goto Neg)
```

Answer: to judge if the highest number is 1.

2. Question : What is it used for :

```
loc0        .FILL   x4000
loc1        .FILL   x4001
loc2        .FILL   x4002
loc3        .FILL   x4003
loc4        .FILL   x4004
```

Answer : to store the 4 decimal number (ASCII) at these locations.