**Algorithm explanation :**

We can only input digit in single, so we should multiply 10 to the record and add the new input to the record, if we encounter the enter key we jump to the transfer mode.

If the unsigned number is larger than xefff, we should transfer it into a legal number to LC-3, so we mask the highest bit in it, and add 8 to the highest bit of the output hex number. For legal number, first we continue minis x1000 until it is smaller than 0, and record the largest number of the highest bit of hex, and then x0100, x0010, x0001, after every circle, we should add the over-minis number to it to restore its value.

Then we can output the result in turn, if the hex number is smaller than 10, we can transfer it to ASCII code and output it at once, if it is larger than 9, we should transfer the number to the ASCII code of ABCDF, and then we stop the program.

**Essential parts of your code with sufficient comments:**

......

ld r5,enter ;r5=xfff6,the negative ASCII code of enter

Get trap x20 ;get input

add r1,r5,r0 ;judge whether the input is enter or not

brz trans ;if the input is enter,than branch to the transfer part

trap x21 ;output the input we get

add r2,r2,#9 ;let r2=9,as a loop counter

loop1 add r3,r1,r3 ;if there exists other single digit input, we should

multiply 10 with r3 and then add to the new input

add r2,r2,#-1

brp loop1 ;then use trap x21 receive ASCII code,we need transfer it into decimal digit which equals to minis 48

......

Trans ...... ;let r0=10,output a new line

lea r4,high ;load the location of current bit digit

ldr r1,r4,#0 ;load the data

add r3,r3,#0 ;if r3>xefff,we need adjust it first

brn adjust

loop2 add r3,r3,r1 ;compare the r3 and r1,if r3<r1,then we need stop

the transfer of current four bits and output it

brn looprestart ;we use r2 to record the current hex number

add r2,r2,#1 ;if r3>r1,we need operate r3-r1,and start the next loop

brnzp loop2

looprestart jsr output ;call the output function

add r4,r4,#1 ;let r4 point to the next parameter to calculate the

next four bits

ldr,r0,r4,#0 ;if the data equals to 0,then stop the program

brz done ;after one circle,we initialize the r2and minis r1 r3 to make up the over-minis load the data of new location

......

brnzp loop2

done trap x25

adjust ld r5,data3 ;if the input is larger than x1000,we can use 0 to

mask the first bit of it

add r2,r2,#8 ;and add 8 to its first r2

output add r0,r2,#-9 ;compare r2 with 9,if r2<9,we can transfer it to

ASCII code and output it at once

brp select ;else we need transfer it to hex character number

ld r5,data1

return add r0,r2,r5 ;load the ASCII code to r0,then output it

select ld r5,data2 ;add x0037 to the decimal digit to transfer it to

ASCII code A B C D E F

brnzp return

......

deal .fill x0000 ;judge the transfer complete or not

data1 .fill x0030 ;store the transfer hex number of decimal number

data2 .fill x0037 ;store the special transfer hex number

......

**Questions TA asked you and your answer in Check:**

Question 1:

Please describe your algorithm.

Answer 1:

Similar to algorithm explanation mode, and add some necessary comments of the machine code. See above for details.

Questions 2:

whether line 23 and 24 instruction can be replaced by LDI instruction?

Answer 2:

Yes, but I use r4 to point the operate address, and with the offset of r4, it decides the continue and end of the program.