**Algorithm explanation:**

At first, we should make clear the interrupt routine, when we tick a key, the instruction in x0180 will response, so we should put the interrupt program in this location which point to x2000. Then we load the PC and PSR into SSP as usual, start to deal with the program.

For normal part, we output point and character in certain order, when the counter add to 20, we start a new line, and we first output the point and follow three continuous letters, and then output the point to meet the number need. When we tick the key, we first judge the category we ticked, if the key represents a number, we add the number to adjust the output location of the letters; if the key is a character, we adjust the character output, then we complete the whole program.

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

.orig x0200

......

ld r1,KBINT ;when receive a type, point to the execution place

ld r2,branch

str r2,r1, #0 ;store the location in the x2000

ld r2,mask ;let the KBSR[15] to 1, stay the write enable state

ld r1,KBSR

str r2,r1, #0

......

rti

KBINT .fill x0180 ;the type responsed location

KBSR .fill xfe00 ;the location of KBSR

branch .fill x2000 ;the start location of response mode

mask .fill x4000

.end

.orig x2000

getc

......

add r5,r0, #0 ;judge the input

ld r2,number9 ;if the input is smaller than ASCII code of 9, branch to number mode

add r3,r2,r5

brnz number

ld r2,characa ;if the input is biger than ASCII code of a, branch to character mode

add r3,r2,r5

brzp charac

brnzp over ;if the input is illegal, branch to over and ignore it

number ......

not r5,r5 ;add the offset to r4

add r4,r5,r4

ld r5,numbertest

add r5,r5,r4

brzp over

ld r4,minisset ;if the result is smaller than initializing condition, initialize the r4

brnzp over

charac ......

add r1,r0, #0 ;load r0 to r1, to output it in the next line

brnzp over

......

rti

number9 .fill #-57 ;store the ASCII code of -9

number1 .fill #-48 ;store the ASCII code of -1

characa .fill #-97 ;store the ASCII code of -a

characz .fill #-122 ;store the ASCII code of -z

numbertest .fill #17 ;store the condition number 17

minisset .fill #-17 ;store the condition number -17

......

.end

.orig x3000

...... ;initialize r1 ,let r2=0,r3=3,r4=-17

ld r4,minisst

loop ld r0,point ;output point in the line first

add r5,r2,r4

brz output

trap x21

jsr delay ;delay the output appropriately

add r2,r2, #1 ;loop condition

brnzp loop

output ...... ;output the current character three times

reloop add r2,r2, #3 ;after output the character, continue output the point

ld r5,ministt

loop2 add r3,r5,r2 ;judge the total output number in one line

brz reset ;if the total number reach the max number of each line, branch to reset

ld r0,point

trap x21

jsr delay ;delay the output appropriately

add r2,r2, #1

brnzp loop2

reset ......

and r2,r2, #0 ;reinitialize the count number

ld r3,loopno ;reinitialize the character count number

add r4,r4, #1 ;minis one to output the character one location ahead

brnz loop

brp change

change add r4,r4, #-1 ;if the first output location is front end, we maintain the location number in r4

brnzp loop

delay ...... ;delay mode, store the original data in

ld r2,delaypart

loop3 add r2,r2, #-1 ;use large number circle to delay each output state

brnp loop3

......

ret

first .fill x0061 ;store the ASCII code of a

point .fill x002e ;store the ASCII code of .

enter .fill x000a ;store the ASCII code of enter

......

delaypart .fill x0300 ;set the delay degree

savep2 .blkw 1

.end

**Questions and Answers:**

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.

Question 2:

What is the meaning of x8002 in PSR after execute the x0200 mode or is it contradicts with the instruction x3000 ?

Answer 2:

No, the x8002 in PSR doesn’t contradict with the instruction x3000, for x8002, the first bit of it is 1, which stands for user mode, and x3000 is in the use mode, so the current data ] can match with its requirement.

And the priority in x8002 is zero, which means the last, so the interrupt can easily bother it and take other action.

The last three bits of it is 010, which match the ordinary case.

Question 3:

Are there some inevitable bugs in input the character ?

Answer 3:

Yes. If we tick the key in a very high rates, the new input will replace the original one, and be output. So there will have some cases, the program output other illegal input and put other characters follow the original character in the same line.

Because we output the character and point relay on the trap x23, so it is hard to avoid it.