\*\*\*\*\*\*\*\*\* Ask User to Input 10 Words \*\*\*\*\*\*\*\*\*\*\*\*\*

LDR 0,0,0 \\ R0=0

AIR 0,10 \\ R0=R0+10; R0 is initialized as counter

LDX 2,100 \\ IX2=151

CHK 1,0 \\ check the status of keyboard to R1

JZ 1,3,4 \\ Jump to line 4 if R1=0

IN 2,0 \\ Take input for keyboard to R2

STR 2,2,0 \\ Store R2; save the key user pressed in the memory

OUT 2,1 \\ Print R2 out

AIX 2,1 \\ Add IX2 by 1

SMR 2,0,110 \\ Sub 13 R2 (ASCII of the CR is saved in 110 ADDR) to check if it is CR

JNE 2,3,4 \\ Jump to line 4 if R2!=0 (if is not CR, take another char for the word, otherwise skip to take next word

AIX 1,1 \\ Add IX1 by 1

STX 1,111 \\ Store IX1 to memory 111;

LDX 2,111,1 \\ Load IX2 form memory 111 indirectly

SIR 0,1 \\ Sub R0 by 1; decrease counter by 1

JNE 0,3,4 \\ Jump to line 4 if R0!=0; if not then input provided, go back to take more

LDR 0,0,0 \\ R0=0

AIR 0,42 \\ R0=R0+42; set R0 as ASCII value of ‘\*’ character

OUT 0,1 \\ Print out R0; as separator

OUT 0,1 \\ Print out R0

OUT 0,1 \\ Print out R0

OUT 0,1 \\ Print out R0

OUT 0,1 \\ Print out R0

OUT 0,1 \\ Print out R0

LDR 0,0,110 \\ set R0 to ASCII of CR

OUT 0,1 \\ print out CR (new line)

\*\*\*\*\*\*\*\*\*\* Sorting \*\*\*\*\*\*\*\*\*

LDR 0,0,115 \\ set R0=25

LDX 1,0

LDX 2,0 \\ set IX1 and IX2 = 0

LDR 2,0,0 \\ load r2

AIR 2,10 \\ and set it 10 as counter

LDR 1,1,150 \\ load the first letter of first word

SIR 1,65 \\ sub it by 65. It optimize the calculation process

MLT 1,0 \\ multiple R1 by 25

AIX 1,1 \\ index IX1 increase by 1

AMR 1,1,150 \\ add next letter into R1

STR 1,2,50 \\ store R1 in memory 50

AIX 2,1 \\ index IX2 increased by 1

SIR 2,1 \\ counter decreased by 1

AIX 1,9 \\ IX1 increased by 9

JNE 2,3,32 \\ if counter not equal to 0, go fetch the next word

LDX 2,0 \\ IX2 set to 0

LDR 1,0,0

AIR 1,10 \\ R1 set to 10

LDR 0,0,0

AIR 0,10 \\ R0 set to 10

LDR 2,2,50 \\ Load first word for memory 50

JZ 2,3,57 \\ If it's 0, go to print it out

SIR 2,1 \\ IR2 decreased by 1

STR 2,2,50 \\ Store it back to memory

AIX 2,1 \\ add index IX2 by 1

SIR 1,1 \\ Sub counter IR1 by 1

JNE 1,3,47 \\ if counter is not 0 , start the loop again

AIR 1,10 \\ add IR1 by 10 to restore the counter

LDX 2,0 \\ IX2 set to 0

JNE 1,3,47 \\ If R1 not 0, go to the start point of this loop

SIR 0,1 \\ Sub R0 by 1 to update the counter

\*\*\*\*\*\*\*\*\*\*\*\*\* Print Out the Word in Ascending Order \*\*\*\*\*\*\*

LDR 3,0,0

LDR 3,2,100 \\ load R3 the index address of the word

STR 3,0,20 \\ store it memory 20

LDX 1,20 \\ load it to IX1

LDR 3,1,0 \\ load the start address of the word

SIR 3,13

JZ 3,3,69 \\ check if this letter is CR. If yes, end of this section and go back to sorting.

AIR 3,13 \\ add R3 back to its original value

OUT 3,1 \\ print out this letter

AIX 1,1 \\ index IX1 added by 1

JMP 3,62 \\ jump to the start point of print section

LDR 3,0,110 \\ load R3 CR

OUT 3,1 \\ print a new line

JNE 0,3,49 \\ if R0 not 0, go back to sorting

LDR 1,0,0 \\ print out separator

AIR 1,42

OUT 1,1

OUT 1,1

OUT 1,1

OUT 1,1

OUT 1,1

OUT 1,1 \\ print out the separator at the end of this section

LDR 1,0,110

OUT 1,1 \\ print out a new line

AIR 1,50

OUT 1,1 \\ start of next section

\*\*\*\*\*\*\*\*\* Ask User to Input another word \*\*\*\*\*\*\*\*\*

LDR 1,0,0 \\ load R1 as counter

AIR 1,1 \\ counter is set to 1

LDX 2,99 \\ IX2 set to 140

AMR 2,0,110 \\ add IR2 by 13

CHK 0,0 \\ check the status of the keyboard

JZ 0,3,88 \\ If the status is 0, waiting for user input

IN 2,0 \\ Take the input to IR2

STR 2,2,0 \\ Store IR2 to memory 140

OUT 2,1 \\ print out IR2

AIX 2,1 \\ add IX2 by 1

SMR 2,0,110 \\ Check if R2 is CR

JNE 2,3,87 \\ If no, take the next letter

LDR 1,0,0

AIR 1,42 \\ set R1=42

OUT 1,1

OUT 1,1

OUT 1,1

OUT 1,1

OUT 1,1

OUT 1,1 \\ print R1 as separator

LDR 1,0,110 \\ load R1 the CR

OUT 1,1 \\ print out a new line

SIX 2,141 \\ calculate the length of the word

STX 2,98 \\ Store R2 in memory 97. The value is saved as counter for the next function

\*\*\*\*\*\*\*\* Find All Match from the Word List \*\*\*\*\*\*\*\*

LDR 0,0,98 \\ load the counter to R0 form memory 98. This counter equals to the length of the word

LDR 2,0,90

AIR 2,100 \\ load another counter to R2. R2=100. That the overall space to store the word list

LDX 1,0 \\ load index 1 IX1=0

LDX 2,0 \\ load index 2 IX2=0

STX 2,97 \\ store IX2 to memory 97 for backup

LDR 1,1,140 \\ load the first letter of word into R1

SMR 1,2,150 \\ sub R1 form memory 150 (the start address of the wrod list)

STR 1,1,90 \\ store R1 to memory 90

AIX 1,1 \\ increase IX1 by 1

AIX 2,1 \\ increase IX2 by 1

SIR 0,1 \\ sub counter R0 by 1.

JNE 0,3,114 \\ If one loop not finished, go back to the loop starting point

LDR 0,0,98 \\ Re-initilize the counter R0

LDX 2,97 \\ load IX2 for backup

AIX 2,1 \\ Add IX2 by 1

STX 2,97 \\ Store IX2 for backup

LDR 1,0,0 \\ load R1=0

AMR 1,0,90 \\ add R1 with memory 90

AMR 1,0,91 \\ add R2 with memory 91

AMR 1,0,92 \\ add R2 with memory 92

SIR 2,1 \\ R2-1. Decrease the overall counter

LDX 1,0 \\ Re-initialize R1=0

JZ 2,3,156 \\ If R2=0, indicate the end of the program. Jump to the last instruction HLT

JZ 1,3,134 \\ If R1=0, we find a match. Go to print out section

JNE 2,3,114 \\ If R2!=0, restart the looping

\*\*\*\*\*\*\*\*\* Print Out All Match Words \*\*\*\*\*\*\*\*\*\*\*\*

STR 0,0,96

STR 2,0,95 \\ store R2 and R0 for backup

LDR 2,0,97 \\ load R2 form memory 97 which is the index value of current loop

LDR 0,0,0

AIR 0,10 \\ set R0=0

DVD 2,0 \\ divide R2 by R0

MLT 2,0 \\ multiple R2 by R0. Those steps are intended to get the address of the first lettler of the word

AIR 2,150 \\ R2= R2 + 150.Those steps are intended to get the address of the first lettle of the word

STR 2,0,94 \\ store R2 to memory 94 for backup

LDR 2,0,94,1 \\ load R2 from memory 94 indirectly. Fetch the first lettle of the word.

OUT 2,1 \\ Print out R2

LDR 2,0,94 \\load R2 from memory 94

AIR 2,1 \\ and add it by 1

STR 2,0,94 \\ store it back

LDR 2,0,94,1 \\ load R2 from memory 94 indirectly. Fetch the next lettle of the word.

SMR 2,0,110 \\ check the word if it's CR

JNE 2,3,143 \\ if no, go to print it out

LDR 2,0,110 \\ load R2=CR

OUT 2,1 \\ print a new line for the next word

LDR 0,0,96 \\ restore R0

LDR 2,0,95 \\ restore R2

JNE 2,3,114 \\ return to the previous section to find the next match

HLT \\ End of the program