// GET 20 INPUT DATA FROM USER

/\* Get one character from the keyboard \*/

LDR 1,0,0 // Set R1 To 0

CHK 0,0 // Check If There Is An Input

JZ 0,3,2 // If Not Jump CHK Till User Provides Input

IN 2,0 // Get Input From Device And Save At R2

STR 2,0,158 // Store Input Temporarily In Addr-158

LDR 3,0,158 // Copy Input To Register-3

SMR 3,0,155 // Subtract Cr From Input (AT Addr-155, We Have 13-ASCII Value Of Cr)

JZ 3,3,13 // If It Is Cr, User Provided Numbers, Go To Calculate Integer Value

STR 2,2,0 // IF NOT CR SAVE FIRST INPUT IN THE FIRST INDEX ADDRESS

AIR 1,1 // Increase R1 By 1 To Check How Many Inputs Provided By User (Character-Counter)

AIX 2,1 // Increase Index Register-2 By 1 To Save Next Input

JMP 3,2 // Jump Back To Take Other Input From User

/\* Calculate the integer value from the numbers user provided \*/

LDR 3,0,153 // Load 1 To Register-3 To Calculate Units Digit

SIX 2,1 // Subtract 1 From Index Register-2 To Get The Latest Number User Provided

LDR 2,2,0 // Load This Number Register 2

MLT 2,3 // Multiply By Reg-3 To Calculate Units Digit

AMR 2,0,156 // Add Prev. Result (FOR Units Digit Add 0)

STR 2,0,156 // Store Result To Addr-156

LDR 0,0,154 // Load 10 To Reg-3 (ADDR-154 Holds To In The Beginning Of The Program

MLT 3,0 // Multiply 10 By Reg-3 To Calculate Factor On Next Unit (10X1 For Decimal Units)

STR 3,0,154 // Store Back The Factor At Addr-154

SIX 2,1 // Decrease Index Reg-2 By 1 To Get The Next Number

SIR 1,1 // Decrease Character-Counter By 1

JZ 1,3,26 // If It Is The Last Input, Calculation Of Integer Finished, Go Print Function

JMP 3,15 // If Not Get The Other Input Provided By User

OUT 2,1 // Print The Input As Integer Value

STR 2,0,151,1 // Store The First Input In The Predefined Address

LDR 3,0,151 // Load Predefined Address In Reg-3

AIR 3,1 // Increase By 1 Predefined Addr To Save Next Input

STR 3,0,151 // Save Back Updated Predefined Address

LDR 1,0,150 // Load Input-Counter To Reg-1 (ADDR-15O Holds Input-Counter=20)

SIR 1,1 // Decrease By 1 Input-Counter

STR 1,0,150 // Save Back The Updated Input-Counter

AIX 2,1 // Increase Index Reg-2 By 1

LDR 2,0,0 // Zeroise Reg-2

STR 2,0,160 // Zeroise Addr-160 For Next Inputs

STR 2,0,156 // Zeroise Addr-156 For Next Inputs

AIR 2,10 // Set 10 Reg-2

STR 2,0,154 // Set Back 1o Addr-154 For Factorization

AIX 1,1 // Show Input Provided So Far At The Index-Register 2

JNE 1,3,1 // Go Back To First Line To Take Next Input

LDR 0,0,11 // User Provided 20 Inputs, Load Separator From Addr-11. (IT Holds Separator)

OUT 0,1 // Print Separator (111111)

// COPY TO ANOTHER LOCATION

LDR 0,0,249 // R0 set to the counter 20

LDR 1,0,249 // R1 set to the counter 20

LDX 1,200 // IX1 set to 0

LDX 2,200 // IX2 set to 0

LDR 2,2,100 // Fetch the first data into R2---------------line number one

STR 2,2,210 // Store R2 to M[210]

AIX 2,1 // IX2+1, update the index

SIR 0,1 // R0-1, update the counter

JNE 0,3,45 // Jump to line number one above.this loop copy data into memory starting from 210.

LDR 0,0,249 // R0 set to the counter 20

LDX 2,200 // IX2 set to 0

// SORT

LDR 2,2,100 // Fetch the first data into R2----------------line number two

JZ 2,3,62 // If R2=0, jump to print section.line number three

SIR 2,1 // R2-1

STR 2,2,100 // Update the date in memory

AIX 2,1 // IX2+1 Update the index

SIR 1,1 // R1-1 R1 is the counter

JNE 1,3,52 // If R1 not 0,jump to the start of this cycle line number two

LDR 1,0,249 // R1 set to 5

LDX 2,200 // IX2 set to 0

JNE 1,3,52 // If R1 not 0,jump to the start of this cycle

// PRINT

LDR 3,2,210; // Fetch data form memory 210 into R3--------------line number three

STR 3,1,60 // Store R3 into memory starting form 60. The data in this address will be stored in ascending order

LDR 3,1,60 // Load R3 from memory 60

AIX 1,1 // IX1+1

OUT 3,1 // Print out R3 to printer

SIR 0,1 // R0-1, R0 is the counter

SIR 2,1 // R2-1

STR 2,2,100 // Update the date in memory

JNE 0,3,52 // If R0 not 0, jump to sorting section

LDR 1,0,11 // Load Separator

OUT 1,1 // Print Separator

// GET SEARCH-KEY INPUT FROM THE USER

/\* This section is exactly the same as first section, but it gets only one integer

and saves as integer in the Register-2 to search this key in the vector. \*/

LDR 1,0,0

LDX 1,143

CHK 0,0

JZ 0,3,78

IN 2,0

STR 2,0,158

LDR 3,0,158

SMR 3,0,155

JZ 3,3,89

STR 2,1,0

AIR 1,1

AIX 1,1

JMP 3,78

LDR 3,0,153

SIX 1,1

LDR 2,1,0

MLT 2,3

AMR 2,0,156

STR 2,0,156

LDR 0,0,144

MLT 3,0

STR 3,0,144

SIX 1,1

SIR 1,1

JZ 1,3,102

JMP 3,91

OUT 2,1

LDR 0,0,11

OUT 0,1

// COMPARISON

LDR 0,0,249 // R0 is set to the counter

LDR 1,0,249 // R1 is set to the counter

LDX 1,200 // IX1 is set to 0

LDX 2,200 // IX2 is set to 0

STR 2,0,205 // Store R2 to memory 205

LDR 1,0,205 // Store R1 from memory 205

SMR 1,2,60 // Subtract R1 by M[60+index]

AIX 2,1 // Add IX2 by 1

JGE 1,3,103 // If R1 is greater or equal to 0,jump to the beginning of this section

LDR 0,2,58 // Load R0 with memory M[58+Index]

OUT 0,1 // Print R0 out

LDR 0,2,59 // Load R0 with memory M[59+Index]

OUT 0,1 // Print R0 out

HLT // End of program