;----------------------------------------------------------

; Problem1.s16

; French Fry Productions

;----------------------------------------------------------

CODESEGMENT ; Program logic

main EQU \*

; using R15 for outputs

; using R2 for inputs

;----------------------------------------------------------

; INPUT "line? " x

;----------------------------------------------------------

LDR R15, #format1 ; Print format1

LDAR R2, x ; Input line

SVC #SVC\_READ\_FROM\_TERMINAL

;----------------------------------------------------------

; INPUT "n? " n

;----------------------------------------------------------

LDR R15, #format2 ; Print format2

SVC #SVC\_READ\_FROM\_TERMINAL ; input should happen before

STR R7, n ; Store n

;----------------------------------------------------------

; FOR i := n to 1 by -1

;----------------------------------------------------------

LDR R3, n ; Initialize i (R3) = n

LDR R4, #1 ; R4 is n ;lb

FOR1 EQU \* ; Loop start

; Condition check

COPYR R5, R3 ; Copy current i (R3) to R5 for comparison

CMPR R5, R4 ; Compare i (R5) with 1 (R4)

JMPN R5, endFOR1 ; Jump to end if R5 < 1 or R5 holds value 1

; Print x, ENDL

LDR R15, #format3 ; Print format3

LDAR R2, x ; Print line

SVC #SVC\_WRITE\_TO\_TERMINAL

DECR R3 ; i = i - 1

JMP FOR1 ; Repeat

;----------------------------------------------------------

; End loop

;----------------------------------------------------------

endFOR1 EQU \* ; Loop end

;----------------------------------------------------------

; STOP

;----------------------------------------------------------

SVC #SVC\_TERMINATE\_PROCESS

;----------------------------------------------------------

DATASEGMENT

;----------------------------------------------------------

; \*\*\*Service request number definitions

SVC\_TERMINATE\_PROCESS EQU 100 ; Terminating the process

SVC\_READ\_FROM\_TERMINAL EQU 300 ; Reading input from the terminal

SVC\_WRITE\_TO\_TERMINAL EQU 301 ; Writing output to the terminal

; \*\*\*Static data

x RW 80 ; Reserve 80 characters for x

n RW

format1 DS "line? %2s" ; indirect

format2 DS "n? %7i" ; direct

format3 DS "%2s%n" ; %n newline

END

A screenshot of a computer

Description automatically generated

;----------------------------------------------------------

; Problem2.s16

; French Fry Productions

;----------------------------------------------------------

CODESEGMENT

main EQU \*

; using R15 for outputs

;----------------------------------------------------------

; time := current time

;----------------------------------------------------------

SVC #SVC\_GET\_CURRENT\_TIME

STR R1, x ; Store time

;----------------------------------------------------------

; PRINT "Current time = ", time, endl

;----------------------------------------------------------

LDR R15, #format1 ; Print format1

LDAR R1, x ; Print time

SVC #SVC\_WRITE\_TO\_TERMINAL

;----------------------------------------------------------

; time := current time

;----------------------------------------------------------

SVC #SVC\_GET\_CURRENT\_TIME

STR R1, x ; Store time

;----------------------------------------------------------

; PRINT "Current time = ", time, endl

;----------------------------------------------------------

LDR R15, #format1 ; Print format1

LDAR R1, x ; Print time

SVC #SVC\_WRITE\_TO\_TERMINAL

;----------------------------------------------------------

; time := current time

;----------------------------------------------------------

SVC #SVC\_GET\_CURRENT\_TIME

STR R1, x ; Store time

;----------------------------------------------------------

; PRINT "Current time = ", time, endl

;----------------------------------------------------------

LDR R15, #format1 ; Print format1

LDAR R1, x ; Print time

SVC #SVC\_WRITE\_TO\_TERMINAL

;----------------------------------------------------------

; STOP

;----------------------------------------------------------

SVC #SVC\_TERMINATE\_PROCESS

;----------------------------------------------------------

DATASEGMENT

;----------------------------------------------------------

; \*\*\*Service request number definitions

SVC\_TERMINATE\_PROCESS EQU 100 ; Terminating the process

SVC\_WRITE\_TO\_TERMINAL EQU 301 ; Writing output to the terminal

SVC\_GET\_CURRENT\_TIME EQU 1 ; Getting the current time

; \*\*\*Static data

x RW

format1 DS "Current time = %1i%n"]

END

A screenshot of a computer

Description automatically generated

;----------------------------------------------------------

; Problem3.s16

; French Fry Productions

;----------------------------------------------------------

CODESEGMENT

main EQU \*

;----------------------------------------------------------

; INPUT "x? " x

;----------------------------------------------------------

LDR R15, #format1 ; Print "x? " prompt

SVC #SVC\_READ\_FROM\_TERMINAL ; Read input from terminal

STR R2, x ; Store the input in 'x'

;----------------------------------------------------------

; INPUT "y? " y

;----------------------------------------------------------

LDR R15, #format2 ; Print "y? " prompt

SVC #SVC\_READ\_FROM\_TERMINAL ; Read input from terminal

STR R3, y ; Store the input in 'y'

;----------------------------------------------------------

; PRINT "(x+y) = ", x + y

;----------------------------------------------------------

ADDR R4, R3 ; R4 = x + y

LDR R15, #format3 ; Load format3 "(x+y) = "

SVC #SVC\_WRITE\_TO\_TERMINAL

LDR R4, x ; Initialize (R4) as x to be Result

;----------------------------------------------------------

; PRINT "(x-y) = ", x - y

;----------------------------------------------------------

SUBR R4, R3 ; R4 = x - y

LDR R15, #format4 ; Load format4 ", (x-y) = ", ENDL

SVC #SVC\_WRITE\_TO\_TERMINAL

LDR R4, x ; Initialize (R4) as x to be Result

;----------------------------------------------------------

; PRINT "(x\*y) = ", x \* y

;----------------------------------------------------------

MULR R4, R3 ; R4 = x \* y

LDR R15, #format5 ; Load format5 "(x\*y) = "

SVC #SVC\_WRITE\_TO\_TERMINAL

LDR R4, x ; Initialize (R4) as x to be Result

;----------------------------------------------------------

; PRINT "(x/y) = ", x / y

;----------------------------------------------------------

DIVR R4, R3 ; R4 = x / y

LDR R15, #format6 ; Load format6 ", (x/y) = "

SVC #SVC\_WRITE\_TO\_TERMINAL

LDR R4, x ; Initialize (R4) as x to be Result

;----------------------------------------------------------

; PRINT "(x%y) = ", x % y

;----------------------------------------------------------

MODR R4, R3 ; R4 = x % y

LDR R15, #format7 ; Load format7 ", (x%y) = ", ENDL

SVC #SVC\_WRITE\_TO\_TERMINAL

;----------------------------------------------------------

; STOP

;----------------------------------------------------------

SVC #SVC\_TERMINATE\_PROCESS

;----------------------------------------------------------

DATASEGMENT

;----------------------------------------------------------

; \*\*\*Service request number definitions

SVC\_TERMINATE\_PROCESS EQU 100 ; Terminating the process

SVC\_READ\_FROM\_TERMINAL EQU 300 ; Reading input from the terminal

SVC\_WRITE\_TO\_TERMINAL EQU 301 ; Writing output to the terminal

; \*\*\*Static data

x RW

y RW

format1 DS "x? %2i"

format2 DS "y? %3i"

format3 DS "(x+y) = %4i"

format4 DS ", (x-y) = %4i%n"

format5 DS "(x\*y) = %4i"

format6 DS ", (x/y) = %4i"

format7 DS ", (x MOD y) = %4i%n"

END

A screenshot of a computer program

Description automatically generated

;----------------------------------------------------------

; Problem4.s16

; French Fry Productions

;----------------------------------------------------------

CODESEGMENT

main EQU \*

;----------------------------------------------------------

; count := 0

;----------------------------------------------------------

LDR R3, #0 ; Initialize count (R3) = 0

;----------------------------------------------------------

; sum := 0

;----------------------------------------------------------

LDR R4, #0 ; Initialize sum (R4) = 0

;----------------------------------------------------------

; WHILE (INPUT "x? " x does not return EOF )

;----------------------------------------------------------

WHILE1 EQU \* ; Loop start

;----------------------------------------------------------

; INPUT "x? " x

;----------------------------------------------------------

LDR R15, #format1 ; Print "x? " prompt

SVC #SVC\_READ\_FROM\_TERMINAL ; Read input from terminal

JMPN R0, endWHILE1 ; if R0 is not -1 it wont jump EOF

;----------------------------------------------------------

; count := count + 1

;----------------------------------------------------------

INCR R3 ; count = count + 1

;----------------------------------------------------------

; sum := sum + x

;----------------------------------------------------------

ADDR R4, R2 ; sum (R4) = sum (R4) + x (R2)

JMP WHILE1 ; Repeat

;----------------------------------------------------------

; END loop

;----------------------------------------------------------

endWHILE1 EQU \* ; Loop end

;----------------------------------------------------------

; PRINT "count = ", count, ENDL

;----------------------------------------------------------

LDR R15, #format2 ; Print format2

SVC #SVC\_WRITE\_TO\_TERMINAL

;----------------------------------------------------------

; PRINT " sum = ", sum, ENDL

;----------------------------------------------------------

LDR R15, #format3 ; Print format3

SVC #SVC\_WRITE\_TO\_TERMINAL

;----------------------------------------------------------

; STOP

;----------------------------------------------------------

SVC #SVC\_TERMINATE\_PROCESS

;----------------------------------------------------------

DATASEGMENT

;----------------------------------------------------------

; \*\*\*Service request number definitions

SVC\_TERMINATE\_PROCESS EQU 100 ; Terminating the process

SVC\_READ\_FROM\_TERMINAL EQU 300 ; Reading input from the terminal

SVC\_WRITE\_TO\_TERMINAL EQU 301 ; Writing output to the terminal

; \*\*\*Service request return code definitions

SVC\_EOF EQU -1 ; Terminal input EOF

; \*\*\*Static data

count RW

sum RW

x RW

format1 DS "x? %2i"

format2 DS "count = %3i%n"

format3 DS " sum = %4i%n"

END

A screenshot of a computer program

Description automatically generated

;----------------------------------------------------------

; Problem5.s16

; French Fry Productions

;----------------------------------------------------------

CODESEGMENT

main EQU \*

;----------------------------------------------------------

; c := RandomInt(0,25)+'A'

;----------------------------------------------------------

LDR R10, #65 ; `A` unicode

LDR R1, #25 ; Load upper bound for rand()

SVC #SVC\_GET\_RANDOM\_INTEGER

ADDR R1, R10 ; Add 'A' unicode to random number

;----------------------------------------------------------

; n := RandomInt(0,6)+1

;----------------------------------------------------------

LDR R10, #1 ; Lower bound for random number

LDR R2, #6 ; Load upper bound for rand()

SVC #SVC\_GET\_RANDOM\_INTEGER

ADDR R2, R10 ; Add 1 to random number

;----------------------------------------------------------

; PRINT "c = ",c,ENDL

;----------------------------------------------------------

LDR R15, #format2 ; Print format2

SVC #SVC\_WRITE\_TO\_TERMINAL

;----------------------------------------------------------

; PRINT "n = ",n,ENDL

;----------------------------------------------------------

LDR R15, #format3 ; Print format3

SVC #SVC\_WRITE\_TO\_TERMINAL

;----------------------------------------------------------

; FOR i := 1 TO n

;----------------------------------------------------------

LDR R3, #1 ; Initialize count i (R3) = 1

FOR1 EQU \* ; Loop start

COPYR R10, R3 ; temp value for R3

CMPR R10, R2 ; Compare i (R3) with n (R2)

JMPP R10, endFOR1 ; If i > n, exit loop

;---------------------------------------------------------------------------------------------------

;----------------------------------------------------------

; FOR j := 1 TO i

;----------------------------------------------------------

LDR R4, #1 ; Initialize count j (R4) = 1

FOR2 EQU \* ; Loop start

COPYR R11, R4 ; temp value for R4

CMPR R11, R3 ; Compare j (R4) with i (R3)

JMPP R11, endFOR2 ; If j > i, exit loop

;----------------------------------------------------------

; PRINT c

;----------------------------------------------------------

LDR R15, #format1 ; Print format1

SVC #SVC\_WRITE\_TO\_TERMINAL

INCR R4 ; j = j + 1

JMP FOR2 ; Repeat if j <= i

;----------------------------------------------------------

; END

;----------------------------------------------------------

endFOR2 EQU \*

;----------------------------------------------------------

; IF ( i = n ) THEN

;----------------------------------------------------------

COPYR R10, R3 ; temp value for R3

CMPR R10, R2 ; Compare i (R3) to n (R2)

JMPNZ R10, ELSE ; If i != n, jump to ELSE

;----------------------------------------------------------

; PRINT ".",ENDL

;----------------------------------------------------------

LDR R15, #format4

SVC #SVC\_WRITE\_TO\_TERMINAL

JMP endIF

;----------------------------------------------------------

; ELSE

;----------------------------------------------------------

ELSE EQU \*

;----------------------------------------------------------

; PRINT ","

;----------------------------------------------------------

LDR R15, #format5

SVC #SVC\_WRITE\_TO\_TERMINAL

JMP endIF

;----------------------------------------------------------

; END

;----------------------------------------------------------

endIF EQU \*

;----------------------------------------------------------

; END

;----------------------------------------------------------

INCR R3 ; i = i + 1

JMP FOR1 ; Repeat the FOR1 loop

endFOR1 EQU \*

;----------------------------------------------------------

; STOP

;----------------------------------------------------------

SVC #SVC\_TERMINATE\_PROCESS

;----------------------------------------------------------

DATASEGMENT

;----------------------------------------------------------

; \*\*\*Service request number definitions

SVC\_TERMINATE\_PROCESS EQU 100 ; Terminating the process

SVC\_READ\_FROM\_TERMINAL EQU 300 ; Reading input from the terminal

SVC\_WRITE\_TO\_TERMINAL EQU 301 ; Writing output to the terminal

SVC\_GET\_RANDOM\_INTEGER EQU 2 ;

; \*\*\*Static data

;format10 DS "Out of for loop 1"

;format11 DS "Out of for loop 2"

;format12 DS "in if"

;format13 DS "in else"

;format14 DS "i = %3i (1) n = %2i%n"

;format15 DS "j = %4i (1) i = %3i%n (1)"

format1 DS "%1c"

format2 DS "c = %1c%n"

format3 DS "n = %2i%n"

format4 DS "."

format5 DS ","

END

A screenshot of a computer

Description automatically generated