@ Calculate "sum of even numbers"

# @ YOUR CODE

BL SUM\_SERIES

@ save the result into variable "sumEven"

@ YOUR CODE

@ Calculate "sum of numbers which are multiples of 3"

# @ YOUR CODE

BL SUM\_SERIES

@ save the result into variable " sumMultipleOfThree "

# @ YOUR CODE

B ENDPROG

@ SUM\_SERIES - function

@ X0 - start number

@ X1 - end number

@ X2 - pointer to a selector function @ return to X0 - sum of values

SUM\_SERIES:

## PUSH {X3, X4}

MOV X3, #0 @ this saves the sum value

sumloop:

CMP X0, X1

BGT endsumloop

@ call selector function passed in X2 register

PUSH {X0, LR}

BL X2

MOV X4, X0

## POP {X0, LR}

@ check return value from selector function

CMP X4, #1

BNE skipsum

ADD X3, X3, X0 @ sum = sum + num

skipsum:

ADD X0, X0, #1

B sumloop endsumloop:

MOV X0, X3 @ return sum value

## POP {X3, X4}

## BR LR

@ IS\_EVEN - function

@ X0 - number to check

@ return to X0 - 1 or 0 if number is even or not IS\_EVEN:

## PUSH {X1}

AND X1, X0, #1

CMP X1, #1

BNE evennum

MOV X0, #0 @ odd number

B skipeven evennum:

MOV X0, #1 @ even number

skipeven:

## POP {X1}

## BR LR

@ IS\_MULTIPLE\_OF\_THREE - function

@ X0 - number to check

@ return to X0 - 1 or 0 if number is multiple of 3 or not

IS\_MULTIPLE\_OF\_THREE:

## PUSH {X1}

@ detect if number is positive or negative

CMP X0, #0

BLT makepositive

B skipmakepositive

makepositive:

MOV X1, #-1

## MUL X0, X0, X1

## skipmakepositive:

MOV X1, X0 loop3:

CMP X1, #3

BLT endloop3

SUB X1, X1, #3

B loop3 endloop3:

CMP X1, #0

BNE notmultiple3

MOV X0, #1 @multiple of three

B skipmultiple

notmultiple3:

MOV X0, #0 @ not multiple of three

skipmultiple:

POP {X1}

## BR LR

ENDPROG: