.globl binary\_search

binary\_search:

SUB sp,sp,#20

SUB R10,R3,R2 //endindex-startindex

ADD R10,R2,R10,LSR#1 //startindex+0.5(endindex-startindex)

ADD R8,R8,#1 //numcalls++

STR R8,[sp,#16] //store r8

STR R12,[sp,#12] //numbers[middleindex]

STR R10,[sp,#8] //middleindex

STR R0,[sp,#4] //numbers

STR lr,[sp,#0] //link register

LDR R12,[R0,R10,LSL#2] //numbers[middleindex]

CMP R2,R3 //startindex>endindex

BHI LOOP1 //unsigned >

CMP R12,R1 //numbers[middleindex] and key

BEQ LOOP2 //numbers[middleindex]==key

BHI LOOP3 //numbers[middleindex]>key

BLO LOOP4 //numbers[middleindex]<key

MAJOR:

MOV R7,#0

LDR R8,[sp,#16]

SUB R8,R7,R8// -numcalls

MOV R12,R8

LDR lr,[sp,#0]

LDR R10,[sp,#8]//middleindex

LDR R7,[sp,#4]//base address changed to another register

STR R12,[R7,R10,LSL#2]

ADD sp,sp,#20

MOV pc,lr

LOOP1:

MOV R0,#-1 //return -1

LDR lr,[sp,#0] //restore link register

ADD sp,sp,#20 //clean the flashmem

MOV pc,lr //go back to lr address

LOOP2:

MOV R0,R10 //keyindex=middleindex

B MAJOR

LOOP3:

SUB R3,R10,#1 //endindex=middleindex-1

BL binary\_search

B MAJOR

LOOP4:

ADD R2,R10,#1 //startindex=middleindex+1

BL binary\_search

B MAJOR