1. **Write a program to find the largest/smallest number in an array of 32 numbers**

**LARGEST NUMBER:**

AREA LARGEST, CODE, READONLY

ENTRY ; Mark first instruction to execute

START

MOV R5,#6 ; INTIALISE COUNTER TO 6(i.e. N=7)

LDR R1,=VALUE1 ; LOADS THE ADDRESS OF FIRST VALUE

LDR R2, [R1],#4 ; WORD ALIGN T0 ARRAY ELEMENT

LOOP

LDR R4, [R1], #4 ; WORD ALIGN T0 ARRAY ELEMENT

CMP R2, R4 ; COMPARE NUMBERS

BHI LOOP1 ; IF THE FIRST NUMBER IS > THEN GOTO LOOP1

MOV R2, R4 ; IF THE FIRST NUMBER IS < THEN MOV CONTENT R4 TO R2

LOOP1

SUBS R5, R5, #1 ; DECREMENT COUNTER

CMP R5, #0 ; COMPARE COUNTER TO 0

BNE LOOP ; LOOP BACK TILL ARRAY ENDS

LDR R4, =RESULT ; LOADS THE ADDRESS OF RESULT

STR R2, [R4] ; STORES THE RESULT IN R2

BACK B BACK

NOP

NOP

; ARRAY OF 32 BIT NUMBERS (N=7)

VALUE1

DCD 0X44444444 ;

DCD 0X22222222 ;

DCD 0X11111111 ;

DCD 0X33333333 ;

DCD 0XAAAAAAAA ;

DCD 0X88888888 ;

DCD 0X99999999 ;

AREA DATA2, DATA, READWRITE; TO STORE RESULT IN GIVEN ADDRESS

RESULT DCD 0X0

END ; Mark end of file

**SMALLEST NUMBER:**

AREA SMALLEST, CODE, READONLY

ENTRY ; Mark first instruction to execute

START

MOV R5, #6 ; INTIALISE COUNTER TO 6(i.e. N=7)

LDR R1, =VALUE1 ; LOADS THE ADDRESS OF FIRST VALUE

LDR R2, [R1], #4 ; WORD ALIGN T0 ARRAY ELEMENT

LOOP

LDR R4, [R1], #4 ; WORD ALIGN T0 ARRAY ELEMENT

CMP R2, R4 ; COMPARE NUMBERS

BLS LOOP1 ; IF THE FIRST NUMBER IS < THEN GOTO LOOP1

MOV R2, R4 ; IF THE FIRST NUMBER IS > THEN MOV CONTENT R4 TO R2

LOOP1

SUBS R5, R5, #1 ; DECREMENT COUNTER

CMP R5, #0 ; COMPARE COUNTER TO 0

BNE LOOP ; LOOP BACK TILL ARRAY ENDS

LDR R4,=RESULT ; LOADS THE ADDRESS OF RESULT

STR R2, [R4] ; STORES THE RESULT IN R2

BACK B BACK

NOP

NOP

; ARRAY OF 32 BIT NUMBERS (N=7)

VALUE1

DCD 0X44444444 ;

DCD 0X22222222 ;

DCD 0X11111111 ;

DCD 0X22222222 ;

DCD 0XAAAAAAAA ;

DCD 0X88888888 ;

DCD 0X99999999 ;

AREA DATA2, DATA, READWRITE; TO STORE RESULT IN GIVEN ADDRESS

RESULT DCD 0X0

END ; Mark end of file

1. **Develop an ALP to count the number of ones and zeros in two consecutive memory locations.**

AREA ONEZERO, CODE, READONLY

ENTRY ; Mark first instruction to execute

START

MOV R2, #0 ; COUNTER FOR ONES

MOV R3, #0 ; COUNTER FOR ZEROS

MOV R7, #2 ; COUNTER TO GET TWO WORDS

LDR R6, =VALUE ; LOADS THE ADDRESS OF VALUE

LOOP MOV R1, #32 ; 32 BITS COUNTER

LDR R0, [R6], #4 ; GET THE 32 BIT VALUE

LOOP0 MOVS R0, R0, ROR #1 ; RIGHT SHIFT TO CHECK CARRY BIT (1's/0's)

BHI ONES ; IF CARRY BIT IS 1 GOTO ONES BRANCH OTHERWISE NEXT

ZEROS ADD R3, R3, #1 ; IF CARRY BIT IS 0 THEN INCREMENT THE COUNTER BY 1(R3)

B LOOP1 ; BRANCH TO LOOP1

ONES ADD R2, R2, #1 ; IF CARRY BIT IS 1 THEN INCREMENT THE COUNTER BY 1(R2)

LOOP1 SUBS R1, R1, #1 ; COUNTER VALUE DECREMENTED BY 1

BNE LOOP0 ; IF NOT EQUAL GOTO TO LOOP0 CHECKS 32BIT

SUBS R7, R7, #1 ; COUNTER VALUE DECREMENTED BY 1

CMP R7, #0 ; COMPARE COUNTER R7 TO 0

BNE LOOP ; IF NOT EQUAL GOTO TO LOOP

BACK B BACK

NOP

NOP

VALUE DCD 0X11111111, 0XAA55AA55; TWO VALUES IN AN ARRAY

END ; Mark end of file