**UNIVERSITI TUNKU ABDUL RAHMAN**

**LEE KONG CHIAN FACULTY OF ENGINEERING AND SCIENCE**

**UECS1013 INTRODUCTION TO COMPUTER ORGANIZATION AND ARCHITECTURE**

**ASSIGNMENTS**

**February 2025 Trimester**

| **NO.** | **STUDENT NAME** | **STUDENT ID** | **Tutorial/ Practical Group** |
| --- | --- | --- | --- |
| **1.** | Koh Khai Jeck | 2304740 | **T4/P9** |
| **2.** | Edmund Chan Chee An | 2202241 |  |
| **3.** | Leon Siow Yi Hong | 2204403 |  |
| **4.** | Quak Jing | 2205378 |  |

# **Table of Content**

[**Table of Content 2**](#_kp9tkxbixrth)

[**Flowchart 3**](#_sir39sapthj9)

[**Complete Code 6**](#_dlzk3w9w8f5a)

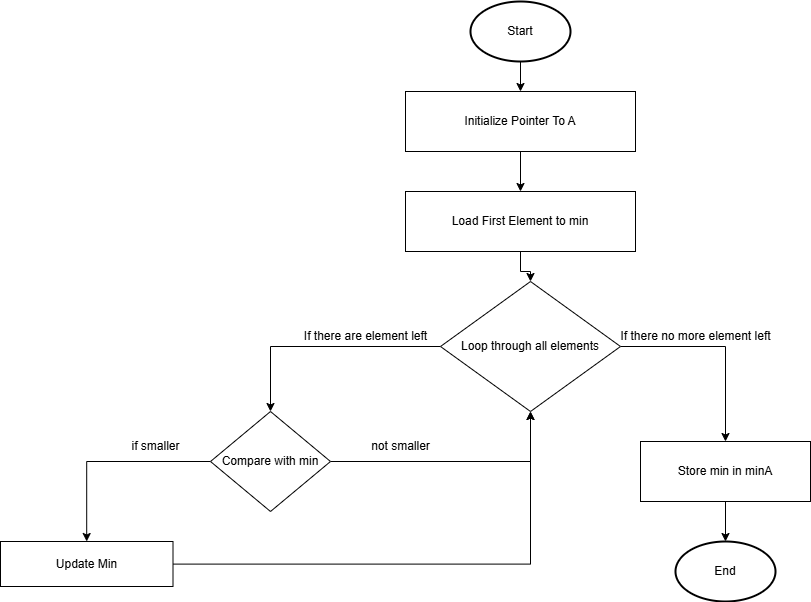
[**Operations and Expected Results 10**](#_b7l3zz7od43f)

[**Results and final values 12**](#_okuw3kamza5r)

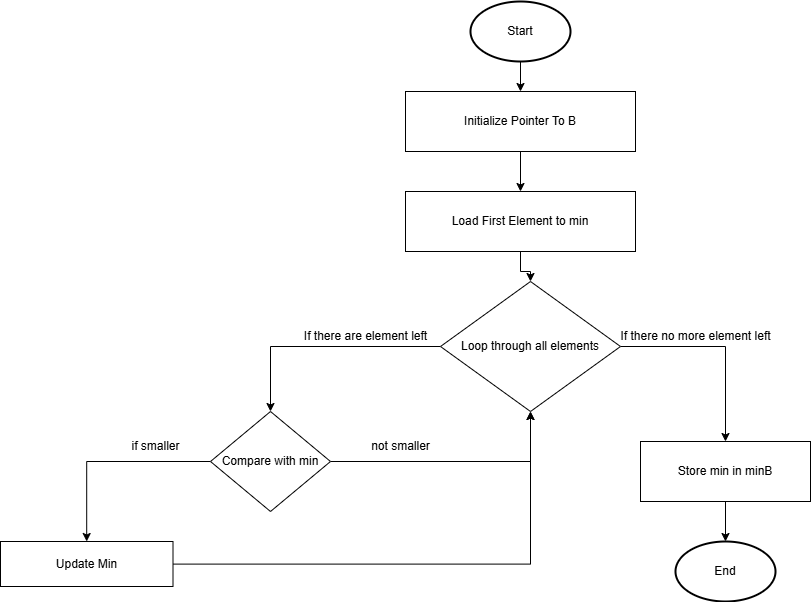
# 

# **Flowchart**

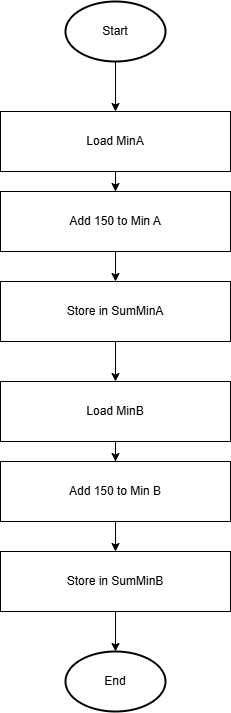
1. Find minimum value in A



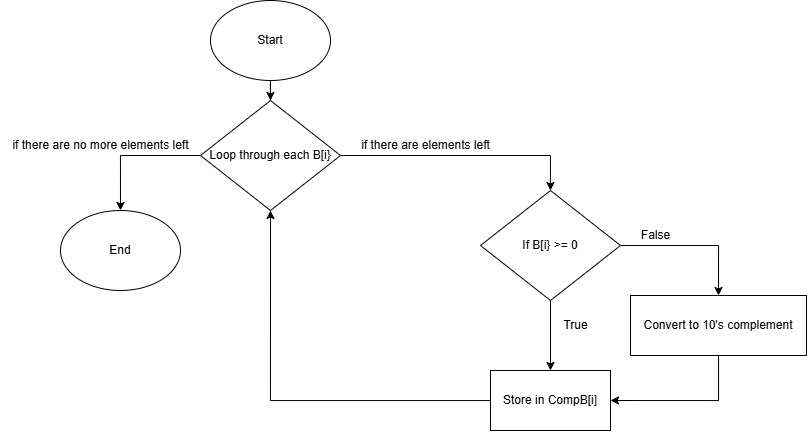
1. Find minimum value in B



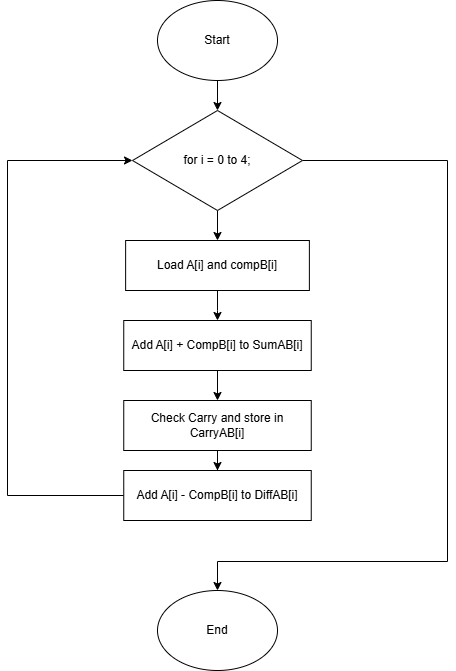
1. Add 150 to minimum A and minimum B



1. Convert List B to 10’s Complement



1. Add A[i] + B[i], A[i] - B[i] (10’s complement) and Detect Carry



# **Complete Code**

EXPORT \_\_main

AREA PROG\_1, CODE, READONLY

\_\_main

; Main program flow — call each subroutine in sequence

BL find\_min\_A

BL find\_min\_B

BL convert\_B\_neg\_assumed\_10s\_complement

BL add\_sub\_lists

stop

B stop

;================= Subroutines ===================

find\_min\_A

LDR R0, =A

MOV R1, #5

LDR R2, [R0], #4

MOV R3, R2

find\_min\_A\_loop

SUBS R1, R1, #1

BEQ store\_min\_A

LDR R2, [R0], #4

CMP R2, R3

MOVLT R3, R2

B find\_min\_A\_loop

store\_min\_A

LDR R0, =MinA

STR R3, [R0]

BX LR

find\_min\_B

LDR R0, =B\_neg\_assumed ; Now using B\_neg\_assumed instead of B

MOV R1, #5

LDR R2, [R0], #4

MOV R4, R2

find\_min\_B\_loop

SUBS R1, R1, #1

BEQ store\_min\_B

LDR R2, [R0], #4

CMP R2, R4

MOVLT R4, R2

B find\_min\_B\_loop

store\_min\_B

LDR R0, =MinB

STR R4, [R0]

; Sum MinA + 150

LDR R1, =MinA

LDR R2, [R1]

ADD R2, R2, #150

LDR R1, =SumMinA

STR R2, [R1]

; Sum MinB + 150

ADD R3, R4, #150

LDR R1, =SumMinB

STR R3, [R1]

BX LR

convert\_B\_neg\_assumed\_10s\_complement

LDR R0, =B\_neg\_assumed ; Use assumed negative list

LDR R1, =CompB

MOV R2, #5

convert\_loop

LDR R3, [R0], #4

CMP R3, #0

BGE store\_positive

; 10's complement for negative number

RSBS R4, R3, #0

MVN R5, R4

ADD R5, R5, #1

STR R5, [R1], #4

B next\_convert

store\_positive

STR R3, [R1], #4

next\_convert

SUBS R2, R2, #1

BNE convert\_loop

BX LR

add\_sub\_lists

LDR R0, =A

LDR R1, =CompB

LDR R2, =SumAB

LDR R3, =DiffAB

LDR R4, =CarryAB

MOV R5, #5

add\_sub\_loop

LDR R6, [R0], #4

LDR R7, [R1], #4

ADDS R8, R6, R7

STR R8, [R2], #4

MOV R9, #0

ADC R9, R9, #0

STR R9, [R4], #4

SUB R8, R6, R7

STR R8, [R3], #4

SUBS R5, R5, #1

BNE add\_sub\_loop

BX LR

;================= Data Section ===================

AREA Data1, DATA, READWRITE

A DCD 3521, 379, 5611, 919, 1318

B DCD 8141, 2615, 53, 951, 217

B\_neg\_assumed DCD 8141, 2615, -53, 951, -217 ; <- Negative versions used in logic

CompB DCD 0, 0, 0, 0, 0

SumAB DCD 0, 0, 0, 0, 0

DiffAB DCD 0, 0, 0, 0, 0

CarryAB DCD 0, 0, 0, 0, 0

MinA DCD 0

MinB DCD 0

SumMinA DCD 0

SumMinB DCD 0

END

# **Operations and Expected Results**

1) Find Minimum Value in List A:

* Load the first value of A into R3 as the initial minimum value
* Loop through the remaining elements of A
* Compare each element with the current minimum value
* If smaller value is found, update the minimum to new minimum value
* Store minimum value in MinA

Expected Output:

List A: {3521, 379, 5611, 919, 1318}

MinA = 379

2) Find Minimum Value in List B:

* Load the first value of B into R4 as the initial minimum value
* Loop through the remaining elements of B
* Compare each element with the current minimum value
* If smaller value is found, update the minimum to new minimum value
* Store minimum value in MinB

Expected Output:

List B: {8141, 2615, 53, 951, 217}

MinB = 53

3) Adding 150 to MinA and MinB

* Load MinA and add 150
* Store result in SumMinA
* Load MinB and add150
* Store result in SumMinB

ExpectedOutput:

SumMinA = 379 + 150 = 529

SumMinB = 53 + 150 = 203

4) Converting List B to 10’s Complement

Assume List B = {8141, 2615, -53, 951, -217}

* Loop through each element in B
* If value is positive store it in CompB
* If Negative, convert to 10’s complement using this formula: 10’s Complement of B[i]=∼(−B[i])+1
* Store result in Comp B

Expected Output:

CompB = {8141, 2615, 4294967243, 951, 4294967079}

5) Addition Subtraction and Carry detection:

Assume List B = {8141, 2615, -53, 951, -217}

* For each index i from 0 to 4:
* Load A[i] and CompB[i]
* Perform SumAB[i] = A[i] + compB[i]
* Perform DiffAB[i] = A[i] + compB[i]
* Find and store carry for each result

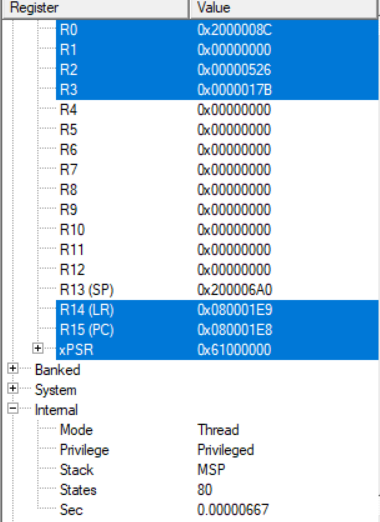
Expected Output

| i | A[i] | CompB[i] | SumAB[i] | Carry | DiffAB[i] |
| --- | --- | --- | --- | --- | --- |
| 0 | 3521 | 9141 | 11662 | 0 | -4620 |
| 1 | 379 | 2615 | 2994 | 0 | -2236 |
| 2 | 5611 | 4294967243 | 4294972854 | 1 | -4294961632 |
| 3 | 919 | 951 | 1870 | 0 | -32 |
| 4 | 1318 | 4294967079 | 4294968397 | 1 | -4294965761 |

# **Results and final values**

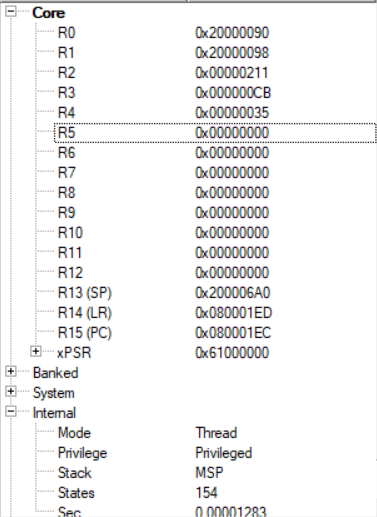
1. Find minimum value in list A

* R0 is the address pointing to MinA
* R1 is the loop counter
* R2 is the last loaded value from the list
* R3 is the final minimum value = 379



1. Find minimum value in list B and add 150 to both minimum value of A and B

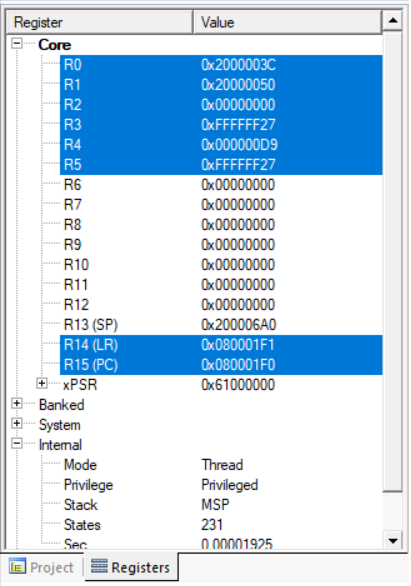
* R0 pointing to a variable
* R1 is the address of SumMinA
* R2 is the value of minimum A + 150 = 529
* R3 is the value of minimum B + 150 = 203
* R4 is the minimum value of list B = 53



1. Convert list B to 10’s complement assuming there are negative values

Assume list B = {8141, 2615, -53, 951, -217}

* R0 and R1 is the pointer
* R2 is the loop counter
* R3 is the last read value from B\_neg\_assumed (-217)
* R4 is the result of line RSBS R4, R3, #0
* R5 is the 10’s complement result stored in CompB



5) Addition Subtraction and Carry detection:

* R0,R1,R2,R3,R4 is pointers
* R5 is the loop counter
* R6 is the loaded value from A[1]
* R7 is the loaded value in CompB[1]
* R8 is the sum

