**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)

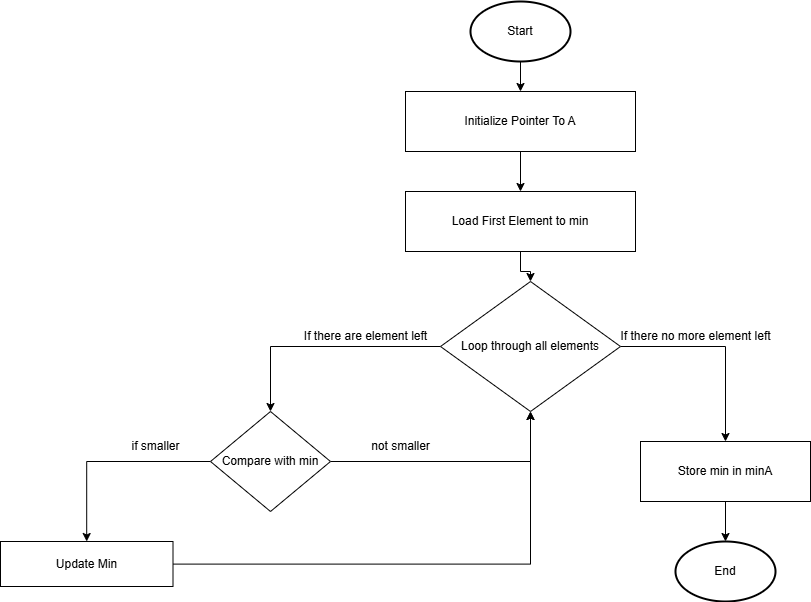
[**Operations and Expected Results 10**](#_b7l3zz7od43f)

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

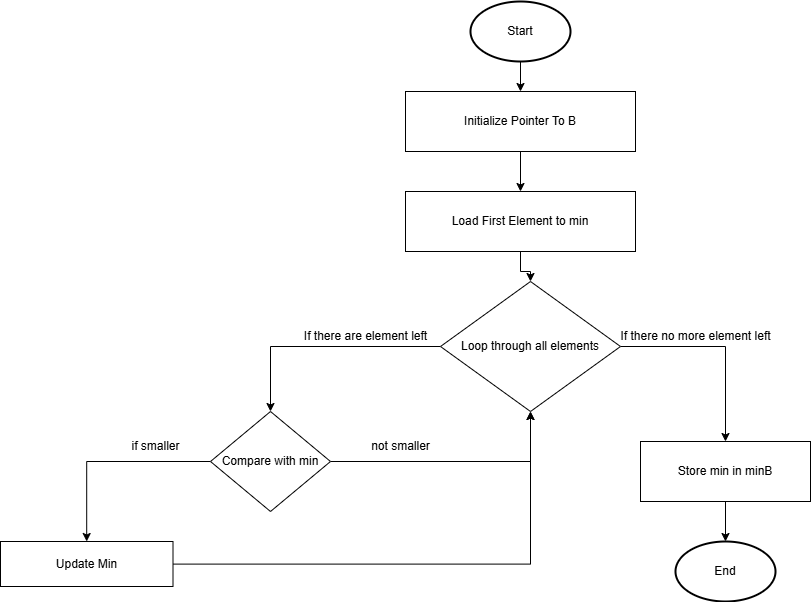
# 

# **Flowchart**

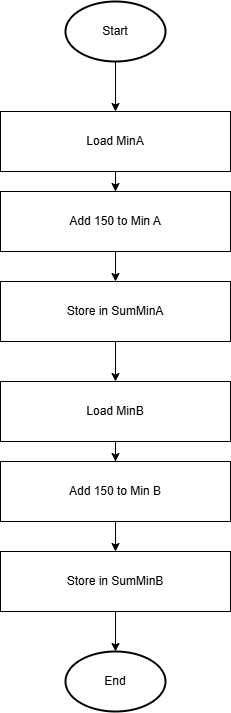
1. Find minimum value in A



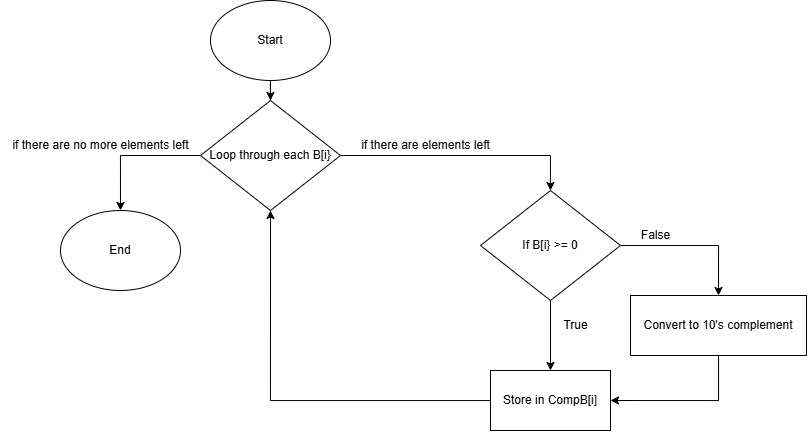
1. Find minimum value in B



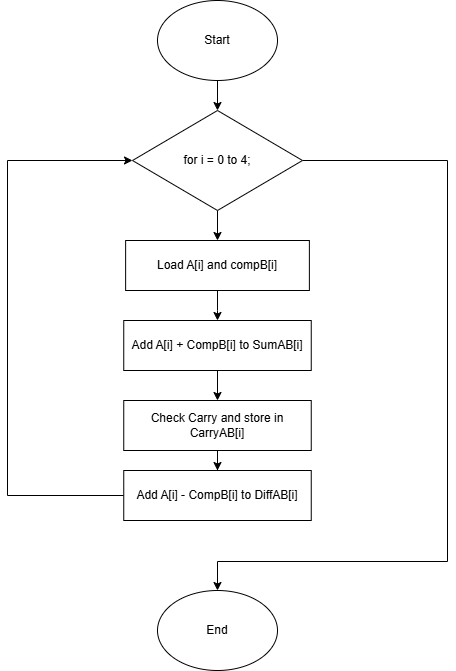
1. Add 150 to minimum A and minimum B



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 ; Call subroutine to find minimum value in array A

BL find\_min\_B ; Call subroutine to find minimum value in array B

BL convert\_B\_10s\_complement ; Convert negative values in B to 10's complement

BL add\_sub\_lists ; Add and subtract elements from A and CompB

stop B stop

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

find\_min\_A

LDR R0, =A ; Load address of array A into R0

MOV R1, #5 ; Set loop counter to 5

LDR R2, [R0], #4 ; Load first value from A into R2, increment R0

MOV R3, R2 ; Set initial minimum (R3) to first element

find\_min\_A\_loop

SUBS R1, R1, #1 ; counter

BEQ store\_min\_A ; If counter is 0, jump to store\_min\_A

LDR R2, [R0], #4 ; Load next element from A

CMP R2, R3 ; Compare with current minimum

MOVLT R3, R2 ; If R2 < R3, update minimum

B find\_min\_A\_loop ; Repeat loop

store\_min\_A

LDR R0, =MinA ; Load address of MinA variable

STR R3, [R0] ; Store minimum value

BX LR ; Return from subroutine

find\_min\_B

LDR R0, =B ; Load address of B

MOV R1, #5 ; Set loop counter to 5

LDR R2, [R0], #4 ; Load first value from B into R2

MOV R4, R2 ; Set initial minimum (R4)

find\_min\_B\_loop

SUBS R1, R1, #1 ; Decrement counter

BEQ store\_min\_B ; If done, store result

LDR R2, [R0], #4 ; Load next value

CMP R2, R4 ; Compare with current minimum

MOVLT R4, R2 ; Update if new value is smaller

B find\_min\_B\_loop ; Repeat loop

store\_min\_B

LDR R0, =MinB ; Load address to store MinB

STR R4, [R0] ; Store minimum value

; Sum MinA + 150

LDR R1, =MinA ; Load address of MinA

LDR R2, [R1] ; Load MinA value

ADD R2, R2, #150 ; Add 150

LDR R1, =SumMinA ; Load address of SumMinA

STR R2, [R1] ; Store result

; Sum MinB + 150

ADD R3, R4, #150 ; Add 150 to MinB

LDR R1, =SumMinB ; Load address of SumMinB

STR R3, [R1] ; Store result

BX LR ; Return from subroutine

convert\_B\_10s\_complement

LDR R0, =B ; R0 -> base address of B

LDR R1, =CompB ; R1 -> base address of CompB

MOV R2, #5 ; R2 = loop counter (5 elements)

LDR R10, =10000 ; R10 = constant 10000

convert\_loop

LDR R3, [R0], #4 ; Load next value from B into R3 and post-increment R0

CMP R3, #0 ; Check if R3 is non-negative

BGE store\_positive ; If R3 >= 0, store it directly

; Negative number: compute 10's complement

RSBS R4, R3, #0 ; R4 = -R3 = absolute value of R3

RSB R5, R4, R10 ; R5 = 10000 - abs(R3)

STR R5, [R1], #4 ; Store result and increment R1

B next\_convert

store\_positive

STR R3, [R1], #4 D ; Store positive number directly

next\_convert

SUBS R2, R2, #1 ; Decrement loop counter

BNE convert\_loop ; If R2 != 0, repeat loop

BX LR ; Return from subroutine

add\_sub\_lists

LDR R0, =A ; Load address of array A

LDR R1, =CompB ; Load address of converted B

LDR R2, =SumAB ; Load address to store sums

LDR R3, =DiffAB ; Load address to store differences

LDR R4, =CarryAB ; Load address to store carry info

MOV R5, #5 ; Set loop counter

add\_sub\_loop

LDR R6, [R0], #4 ; Load value from A

LDR R7, [R1], #4 ; Load value from CompB

ADDS R8, R6, R7 ; Add with carry update

STR R8, [R2], #4 ; Store sum in SumAB

MOV R9, #0 ; Clear R9 for carry

ADC R9, R9, #0 ; Add carry flag to R9

STR R9, [R4], #4 ; Store carry in CarryAB

SUB R8, R6, R7 ; Subtract CompB from A

STR R8, [R3], #4 ; Store result in DiffAB

SUBS R5, R5, #1 ; Decrement loop counter

BNE add\_sub\_loop ; Continue if not zero

BX LR ; Return from subroutine

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

AREA Data1, DATA, READWRITE

A DCD 3521, 379, 5611, 919, 1318

B DCD 8141, 2615, 53, 951, 217

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, 9947, 951, 9783}

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

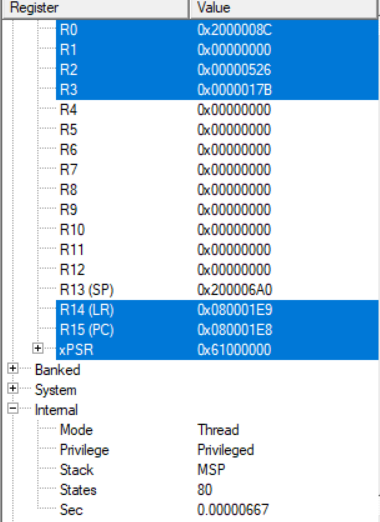
Expected Output

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| i | A[i] | CompB[i] | SumAB[i] | Carry | DiffAB[i] |
| 0 | 3521 | 8141 | 11662 | 0 | -4620 |
| 1 | 379 | 2615 | 2994 | 0 | -2236 |
| 2 | 5611 | 9947 | 15558 | 1 | -4336 |
| 3 | 919 | 951 | 1870 | 0 | -32 |
| 4 | 1318 | 9783 | 11101 | 1 | -8465 |

# **Results and final values**

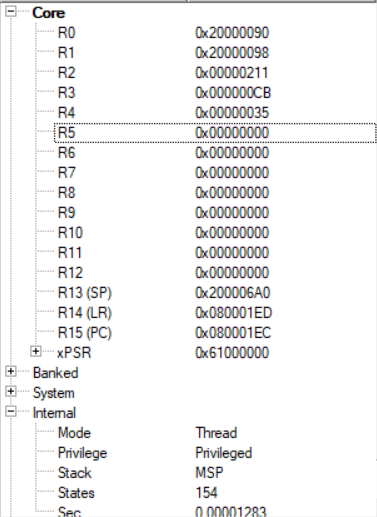
1. Find minimum value in list A

* 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}

* R1 is pointer to CompB

A screenshot of a computer

AI-generated content may be incorrect.

* CompB[0] = 1FCD = 8141

A computer screen with numbers and letters

AI-generated content may be incorrect.

* CompB[1] = A37 = 2615

A screenshot of a computer

AI-generated content may be incorrect.

* CompB[2] = 26DB = 9947

A screen shot of a computer

AI-generated content may be incorrect.

* CompB[3] = 3B7 = 951

A screenshot of a computer

AI-generated content may be incorrect.

* CompB[4] = 2637 = 9783

A screen shot of a computer

AI-generated content may be incorrect.

1. Addition Subtraction and Carry detection:

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

* R2 is address of SumAB
* R3 is address of DiffAB
* R4 is to store info of carry

A screenshot of a computer

AI-generated content may be incorrect.

SumAB = {00002D8E, 00000BB2, 00003CC6, 0000074E, 00002B5D}

= {11662, 2994, 15558, 1870, 11101}

A screenshot of a computer

AI-generated content may be incorrect.

DiffAB = {FFFFEDF4, FFFFF744, FFFFEF10, FFFFFFE0, FFFFDEEF}

DiffAB = { -4620, -2236, -4336, -32, -8465 }

A close-up of a computer screen

AI-generated content may be incorrect.