

## EE447 EXPERIMENT 1 PRELIMINARY WORK-revisited

1.

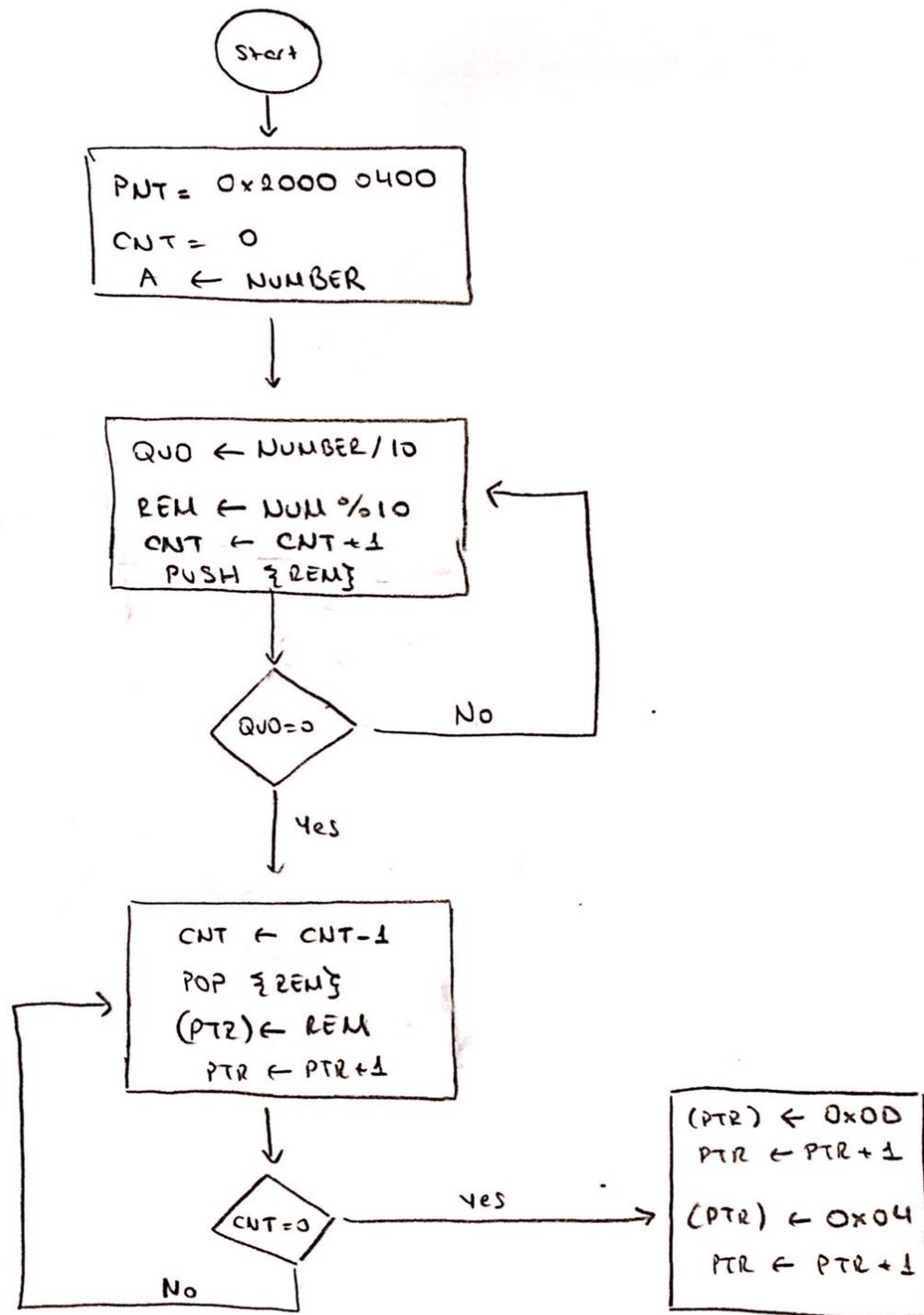


Fig 1. Flowchart of the CONVRT subroutine

```

;LABEL      DIRECTIVE  VALUECOMMENT
NUM    EQU    0X7FFFFFFF

        AREA routines, CODE, READONLY
        THUMB
        EXPORT CONVRT

CONVRT   PUSH {LR}

        LDR R4,=NUM
        LDR R5,=0X20000400
        MOV R0, #10                ; for division by ten
        MOV R7, #0                ;counter

Loop1    UDIV R2, R4, R0            ; R2 is quotient
        MUL R6, R0, R2
        SUB R3, R4, R6            ; R3 remainder

        MOV R4, R2                ; copy r2 to r4

        ADD R7, #1                ;counter increases

        CMP R4, #0

        PUSH {R3}

        BNE Loop1

Loop2    SUBS R7, #1                ;counter=counter-1

        POP {R3}

        MOV R0, R3

        ADD R0, #0X30

        STR R0, [R5], #1

        CMP R7, #0                ;IF R7!=0 LOOP2 continues

        BNE Loop2

        MOV R0, #0X0d

        STR R0, [R5], #1

        MOV R0, #0X04

        STR R0, [R5]

```

```
POP {LR}

BX LR

;LABEL DIRECTIVE VALUE COMMENT
ALIGN
END
```

**Fig 2.** CONVRT Subroutine code

```
;LABEL      DIRECTIVE  VALUE      COMMENT

                                AREA main , CODE, READONLY
                                THUMB
                                EXTERN OutStr
                                EXTERN CONVRT

                                EXPORT __main

                                ENTRY

__main PROC

    BL CONVRT

    LDR R5,=0X20000400

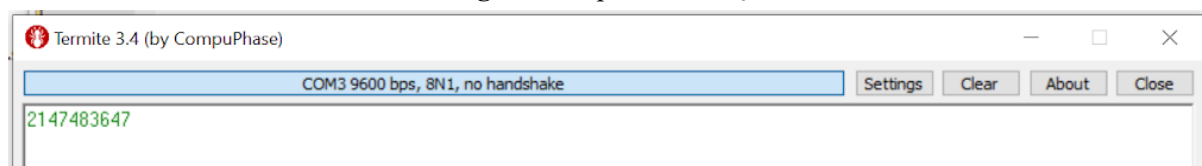
    BL OutStr

Done B Done

    ENDP

    ALIGN
    END
```

**Fig 3.** Main part of the Q1

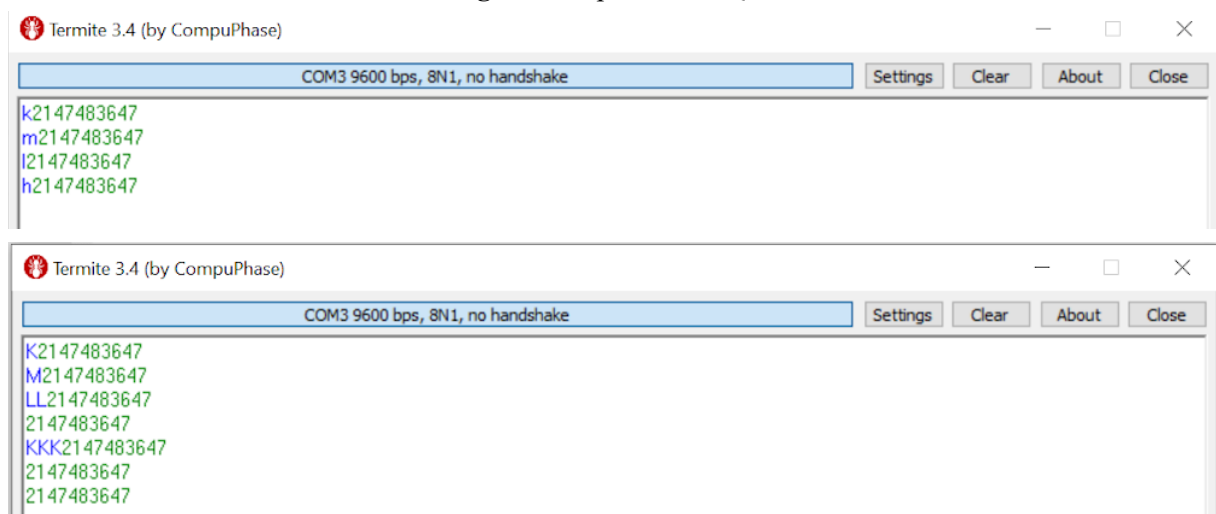


**Fig 4.** Termit Output of the Q1

2.

;LABEL	DIRECTIVE	VALUE	COMMENT
		AREA main , CODE, READONLY	
		THUMB	
		EXTERN OutStr	
		EXTERN CONVRT	
		EXTERN InChar	
		EXPORT __main	
		ENTRY	
__main	PROC		
get	BL InChar		
	BL CONVRT		
	LDR R5,=0X20000400		
	BL OutStr		
	B get		
	ENDP		
	ALIGN		
	END		

**Fig 5.** Main part of the Q2



**Fig 6.** Termit Output Result of the Q2

3.

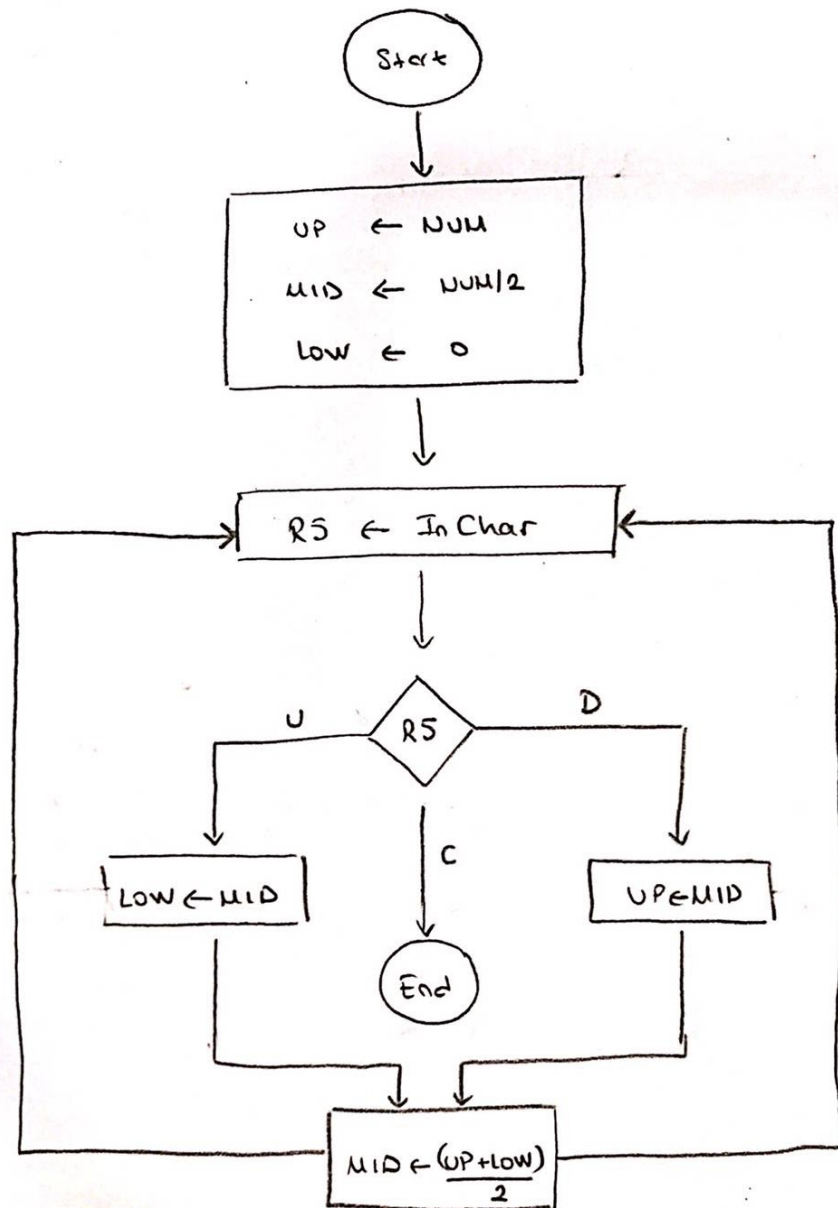


Fig 7. Flowchart of the Q3

;LABEL	DIRECTIVE	VALUE	COMMENT
			AREA routines, CODE, READONLY
			THUMB
			EXPORT UPBND
UPBND	PUSH	{LR}	
	CMP	R5, #0X55	;if the input is U

```

        BEQ Cont1

        CMP R5, #0x44    ;if the input is D

        BEQ Cont2

Cont1 MOV R0, R4        ;LOW<=MID

        B mid

Cont2 MOV R2, R4        ;UP<=MID

mid    ADD R4, R0, R2    ;MID=(LOW+UP)/2

        LSR R4, #1

        POP {LR}

        BX LR

        ALIGN
        END

```

**Fig 8.** UPBND Subroutine Code

```

;LABEL      DIRECTIVE  VALUECOMMENT
                                AREA sdata , DATA, READONLY
                                THUMB
MSG          DCB         "You have found the correct number"
              DCB         0X0D
              DCB         0X04

NUM    EQU    0X40

                                AREA main , CODE, READONLY
                                THUMB
                                EXTERN OutStr
                                EXTERN CONVRT
                                EXTERN UPBND
                                EXTERN InChar
                                EXPORT __main

                                ENTRY

__main PROC

        LDR R2,=NUM    ;initial upper boundary

        LSR R4, R2, #1 ;initial mid value

```

```
        MOV R0,#0X00        ;initial low boundary
get BL InChar

        CMP R5, #0X43        ;if the input is C

        BEQ Cont

        BL UPBND              ;else enter UPBND

        PUSH {R0-R4}

        BL CONVRT

        POP {R0-R4}

        LDR R5,=0X20000400

        B Final

Cont LDR R7, =MSG

        LDR R5,=0X20000400

        STR R7, [R5]

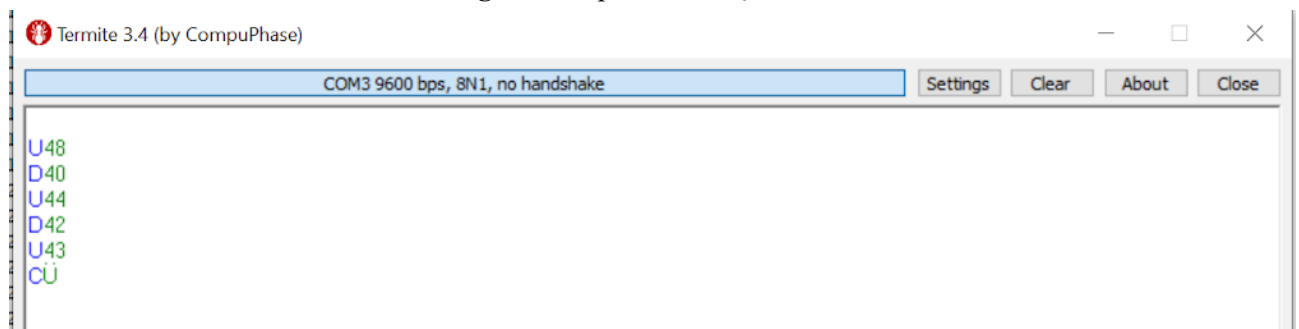
Final  BL OutStr

        B get

        ENDP

        ALIGN
        END
```

**Fig 9.** Main part of the Q3



**Fig 10.** Termit Output of the Q3

In this question, when pressed 'C', I would like to give a message "You've found the correct number". However, I couldn't manage to do it. When pressed C it prints out the letter Ü.

4.

;LABEL	DIRECTIVE	VALUECOMMENT
		AREA routines, CODE, READONLY
		THUMB
		EXPORT mFibonacci
mFibonacci		
		CMP R2, #0
		BEQ Exit
		PUSH{LR}
		MOV R3, R1
		LSL R1, #1
		ADD R1, R0
		MOV R0, R3
		PUSH{R1}
		SUB R2, #1
		BL mFibonacci
		POP{R1}
		POP{LR}
		STR R1, [R9], #4
Exit	BX LR	
		ALIGN
		END

**Fig 11.** mFibonacci Subroutine Code

	AREA main , CODE, READONLY
	THUMB
	EXTERN OutStr
	EXTERN CONVRT
	EXTERN UPBND
	EXTERN InChar
	EXTERN mFibonacci



```
EXPORT __main

ENTRY

__main PROC

    LDR R9,=0X20000500
;-----
    MOV R0, #0          ;Fn-2

    PUSH {R0}

    MOV R4, R0

    BL CONVRT

    LDR R5,=0X20000400

    BL OutStr

;-----
    MOV R1, #1          ;Fn-1

    MOV R4, R1

    BL CONVRT

    LDR R5,=0X20000400

    BL OutStr

;-----
    POP{R0}

    MOV R2, #7          ;INPUT

    PUSH{R2}

    BL mFibonacci

    POP{R2}             ;RESTORE R2

;-----
    LDR R9,=0X20000500

    LSL R2, #2          ;4*R2

    ADD R9, R2          ;R9->R9+4*R2

Loop1 CMP R2, #0

    BEQ Final
```

```

    LDR R4, [R9, #-4]!

    PUSH {R2,R4}

    BL CONVRT

    POP {R2,R4}

    LDR R5,=0X20000400

    BL OutStr

    SUB R2, #4

    B Loop1

Final B Final

    ENDP

        ALIGN
    END

```

**Fig 13.** Main Part of the Q4 entering input from the input

```

;LABEL      DIRECTIVE  VALUECOMMENT

                                AREA main , CODE, READONLY
                                THUMB
                                EXTERN OutStr
                                EXTERN CONVRT
                                EXTERN UPBND
                                EXTERN InChar
                                EXTERN mFibonacci
                                EXPORT __main

                                ENTRY

__main PROC
get BL InChar

    PUSH {R5}

    LDR R9,=0X20000500
;-----
    MOV R0, #0                ;Fn-2

    PUSH {R0}

```

```
MOV R4, R0

BL CONVRT

LDR R5,=0X20000400

BL OutStr

;-----
MOV R1, #1          ;Fn-1

MOV R4, R1

BL CONVRT

LDR R5,=0X20000400

BL OutStr

;-----

POP {R0}

POP {R2} ;r5, the input is moved to r2

SUB R2, #48 ;to interpret ASCII

SUB R2, #2 ;to exclude 0 and 1

PUSH {R2}

BL mFibonacci

POP {R2}          ;RESTORE R2

;-----

LDR R9,=0X20000500

LSL R2, #2

ADD R9, R2          ;R9->R9+(R2-1)

Loop1 CMP R2, #0

BEQ Final

LDR R4, [R9,#-4]!

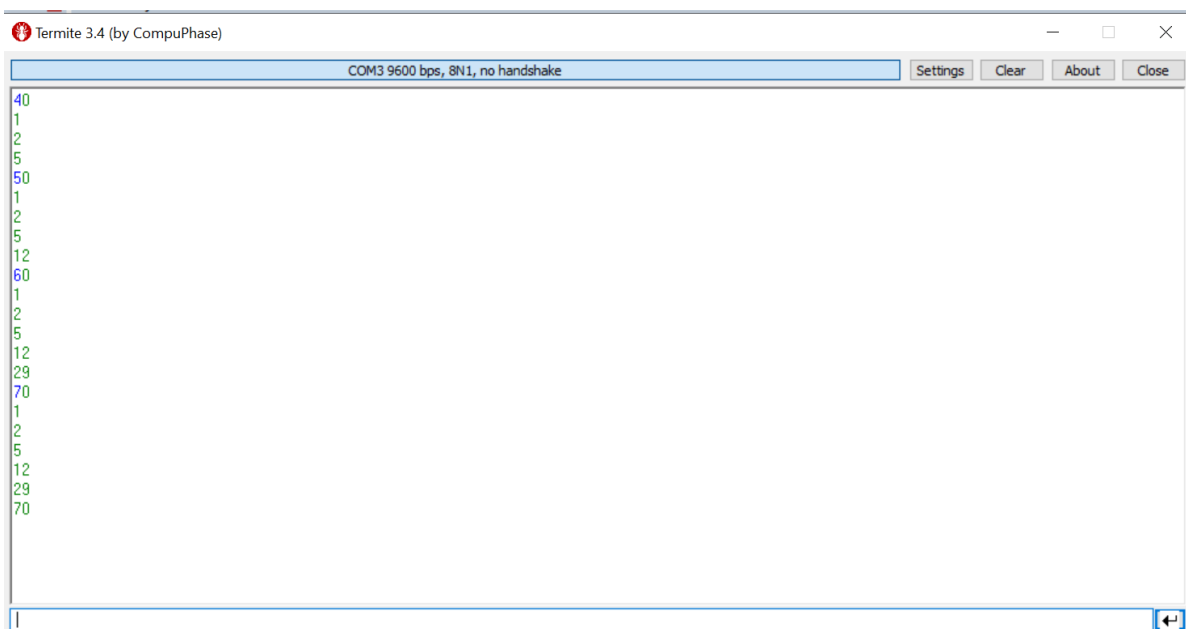
PUSH {R2,R4}

BL CONVRT

POP {R2,R4}
```

```
LDR R5,=0X20000400  
  
BL OutStr  
  
SUB R2, #4  
  
B Loop1  
  
Final B get  
  
ENDP  
  
ALIGN  
END
```

**Fig 14.** Main Part of the ASCII Code



**Fig 15.** Termit Output of the Q4