```
1                AREA        main, READONLY, CODE
2                THUMB
3                EXTERN      OutChar ; Reference external subroutine
4                EXTERN      OutStr
5                EXPORT      __main  ; Make available
6
7    __main      PROC
8    start       LDR         R0,=0x0 ;Temp value 1
9                LDR         R1,=0x0 ;Temp value 2
10               LDR         R2,=0x0 ;Temp value 3
11               LDR         R3,=0xA ;Since we are converting hex to decimal. It's based is 10 ( Hexa
     [A]= Deci [10])
12               LDR         R4,=0x7FFFFFFF ;Value that will be converted
13               LDR         R5,=0x20000480  ;Address value that will be written ASCII Value
14               PUSH        {R5} ; Pushing adress value
15               MOV         R6,R5
16               BL          CONVRT ;Starter for subroutine
17               LDR         R9,=0x0D
18               STR         R9,[R5]
19               ADD         R5,R5,#1
20               LDR         R9,=0x04
21               STR         R9,[R5]
22               LDR         R0,=0x20000480
23               BL          OutStr
24   forever     B           forever
25               ENDP
26
27   CONVRT      PROC
28   loop        CMP         R4,#0
29               BEQ         finish
30               UDIV        R0,R4,R3 ; R0=(R4//0xA)
31               MUL         R1,R0,R3 ; R1=(R0*10) That will be our current digit, starting from unit digit
32               SUB         R2,R4,R1 ; R2= R4-R1 (that will be data for the current digit,starting from
     unit digit)
33               STRB        R2,[R5],#1 ; Writing Datas
34               MOV         R4,R0 ; Updating number so that we can go to next digit
35               CMP         R4,#10 ; If it finishes, the number will be less than 10 otherwise it should
     go to label "loop"
36               BMI         finish
37               B           loop
38   finish      STRB        R4,[R5];    Writing converted data is finished here. It is time to rearrange
     numbers and converting ASCII values
39               MOV         R7,R5
40               ADD         R5,R5,#1
41               MOV         R8,R5
42   loop1       LDRB        R1,[R7] ; This loop is writing the same table at the end of it. However, it
     is in reversed order
43               STRB        R1,[R5]
44               ADD         R5,R5,#1
45               SUB         R7,R7,#1
46               CMP         R7,R6
47               BPL         loop1
48   loop2       LDRB        R1,[R8] ; This loop is writing ASCII values in the reversed table at the
     desired location
49               ADD         R1,R1,#48
50               MOV         R0,R1
51               STRB        R1,[R6]
52               ADD         R6,R6,#1
53               ADD         R8,R8,#1
54               CMP         R8,R5
55               BMI         loop2
56               MOV         R5,R6
57               BX          LR
58               ENDP
59               END
60
61
```

```
1                AREA        main, READONLY, CODE
2                THUMB
3                EXTERN      InChar  ; Reference external subroutine
4                EXTERN      CONVRT; Make available
5                EXTERN      OutStr; Make available
6                EXPORT      __main
7    NUM         EQU         0x20000480
8
9    __main      PROC
10   start       BL          InChar
11               MOV         R4,R0   ;Temp value 1
12               LDR         R1,=0x0 ;Temp value 2
13               LDR         R2,=0x0 ;Temp value 3
14               LDR         R3,=0xA ;Since we are converting hex to decimal. It's based is 10 ( Hexa
     [A]= Deci [10])
15               LDR         R5,=0x20000480  ;Address value that will be written ASCII Value
16               LDR         R9,=0x10
17               MOV         R6,R5
18               BL          CONVRT
19               LDR         R0,=0x20000480
20               BL          OutStr
21               BL          InChar
22               B           start
23   forever     B           forever
24
25               ALIGN
26               ENDP
27               END
```

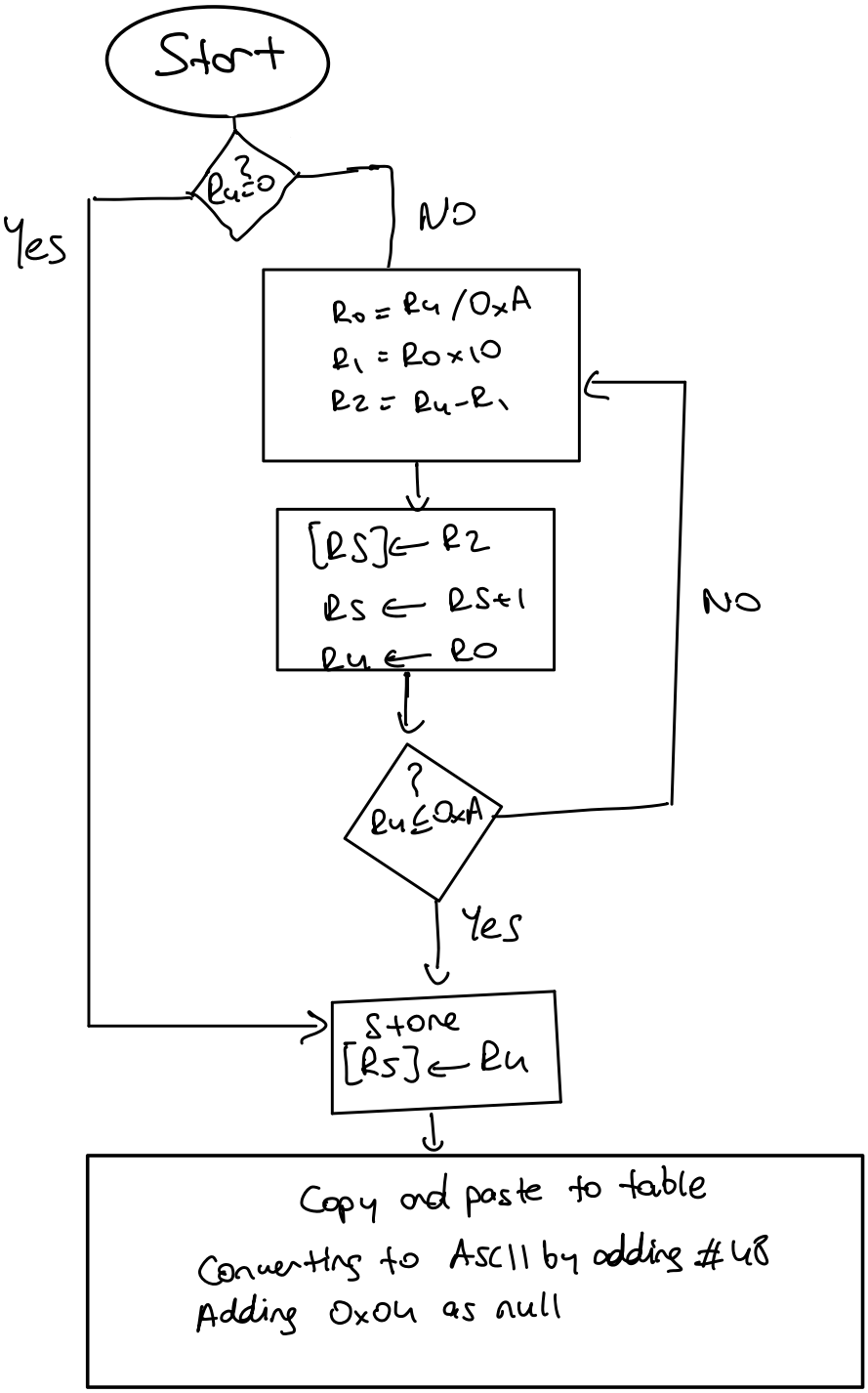
```
1                AREA        main, READONLY, CODE
2                THUMB
3
4                EXTERN      OutStr; Make available
5                EXPORT      CONVRT
6
7    CONVRT      PROC
8    loop        CMP         R4,#0
9                BEQ         finish
10               UDIV        R0,R4,R3 ; R0=(R4//0xA)
11               MUL         R1,R0,R3 ; R1=(R0*10) That will be our current digit, starting from unit digit
12               SUB         R2,R4,R1 ; R2= R4-R1 (that will be data for the current digit,starting from
     unit digit)
13               STRB        R2,[R5],#1 ; Writing Datas
14               MOV         R4,R0 ; Updating number so that we can go to next digit
15               CMP         R4,#10 ; If it finishes, the number will be less than 10 otherwise it should
     go to label "loop"
16               BMI         finish
17               B           loop
18   finish      STRB        R4,[R5];    Writing converted data is finished here. It is time to rearrange
     numbers and converting ASCII values
19               MOV         R7,R5
20               ADD         R5,R5,#1
21               MOV         R8,R5
22   loop1       LDRB        R1,[R7] ; This loop is writing the same table at the end of it. However, it
     is in reversed order
23               STRB        R1,[R5]
24               ADD         R5,R5,#1
25               SUB         R7,R7,#1
26               CMP         R7,R6
27               BPL         loop1
28   loop2       LDRB        R1,[R8] ; This loop is writing ASCII values in the reversed table at the
     desired location
29               ADD         R1,R1,#48
30               MOV         R0,R1
31               STRB        R1,[R6]
32               ADD         R6,R6,#1
33               ADD         R8,R8,#1
34               CMP         R8,R5
35               BMI         loop2
36               LDR         R9,=0x0D
37               STR         R9,[R6]
38               ADD         R6,R6,#1
39               LDR         R9,=0x04
40               STR         R9,[R6]
41               BX          LR
42               ENDP
43
```

```
1                AREA        main, READONLY, CODE
2                THUMB
3                EXTERN      CONVRT
4                EXTERN      OutStr
5                EXTERN      InChar
6                EXTERN      UPBND
7                EXPORT      __main
8
9    NUM         EQU         0x20000400
10
11
12
13   __main      PROC;TAKING DATA AS WITH TWO DIGITS
14               LDR         R3,=0xA
15               LDR         R5,=NUM
16               MOV         R6,R5
17               BL          InChar
18               PUSH        {R0}
19               BL          InChar
20               POP         {R7}
21               SUB         R7,#0x30
22               SUB         R0,#0x30
23               MUL         R7,R3
24               ADD         R0,R7 ; R0 represents input value with decimal
25               ;Prepreation for the alghoritm
26               MOV         R10,#0;This will be held for minimum limit
27               MOV         R11,#1; R11=1
28               LSL         R11,R11,R0;This will be the maximum == 2^n
29
30
31
32   calcu       ADD         R12,R11,R10 ; R1=R11+R0 (MAX+MIN)
33               LSR         R12,R12,#1 ;    R1=R1/2     (MIN+MAX)/2
34               MOV         R4,R12
35               PUSH        {R5,R6}
36               BL          CONVRT
37               LDR         R0,=NUM
38               BL          OutStr
39               POP         {R5,R6}
40               BL          UPBND
41               B           calcu
42   loop        B           loop
43               ALIGN
44               ENDP
```

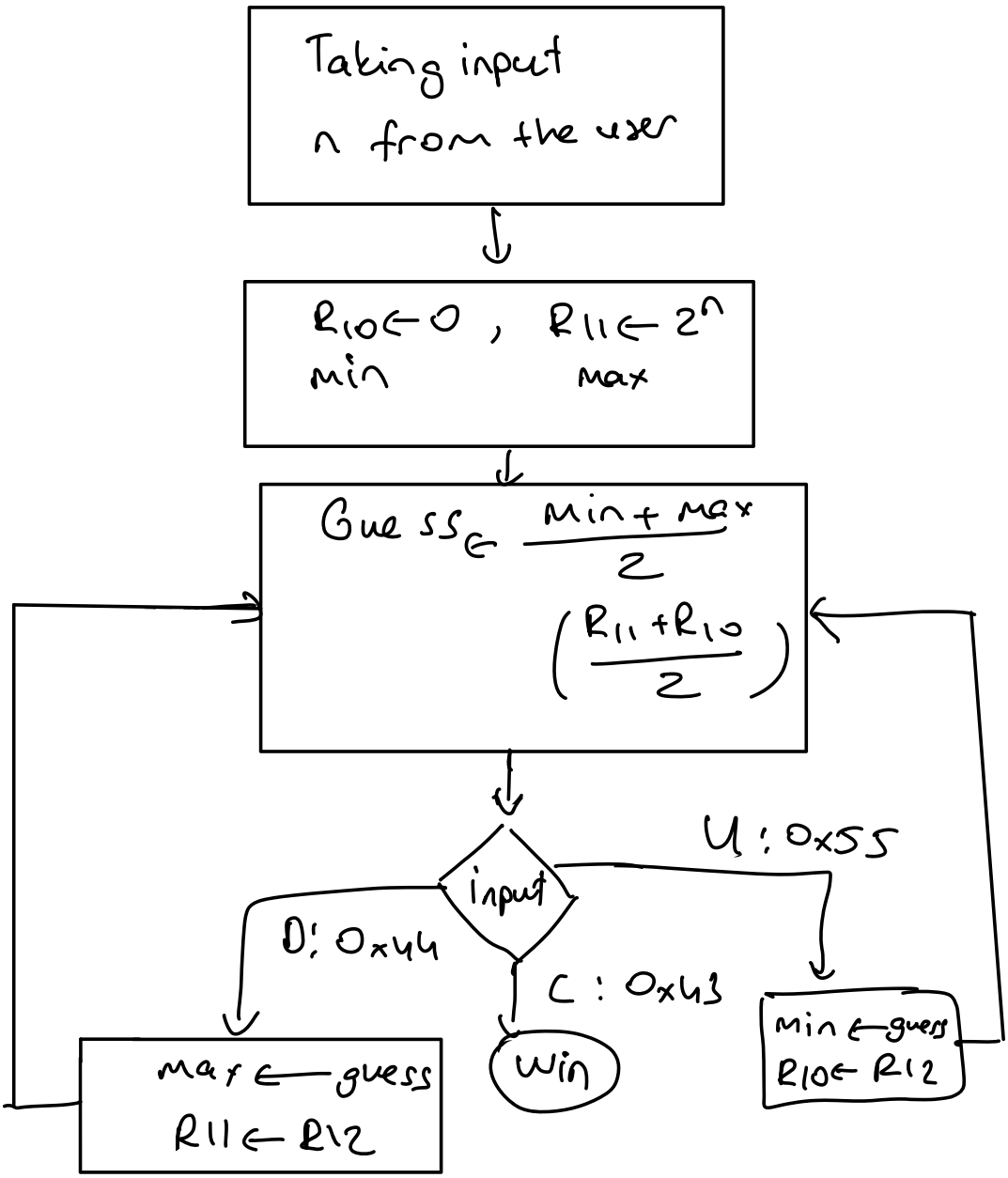
```
1               AREA        main, READONLY, CODE
2               THUMB
3               EXTERN      InChar  ; Reference external subroutine
4               EXPORT      UPBND   ; Make available
5
6   UPBND       PROC
7   start       PUSH        {R2,R3,R4,R5,R6,LR}
8   geta        BL          InChar
9               CMP         R0,#0x43 ; if input == 'C'
10              BEQ         forever  ; Finalize and go to infinite loop
11              CMP         R0,#0x55 ; if input == 'U'
12              BEQ         chLW    ; Change lower bound
13              CMP         R0,#0x44 ; if input == 'D'
14              BEQ         chHG    ; Change upper bound
15              B           geta
16  forever     B           forever
17              ENDP
18
19  chLW        PROC
20              MOV         R10,R12
21              POP         {R2,R3,R4,R5,R6,LR}
22              BX          LR
23
24  chHG        PROC
25              MOV         R11,R12
26              POP         {R2,R3,R4,R5,R6,LR}
27              BX          LR
28              ENDP
29
30
31              END
```

```
1                AREA        main, READONLY, CODE
2                THUMB
3                EXTERN      OutStr
4                EXTERN      InChar
5                EXTERN      CONVRT
6                EXPORT      __main
7   NUM2         EQU         0x20000600
8   NUM          EQU         0x20000400
9   __main       PROC;TAKING DATA AS WITH TWO DIGITS
10               LDR         R3,=0xA
11               LDR         R9,=NUM2
12               BL          InChar
13               PUSH        {R0}
14               BL          InChar
15               POP         {R7}
16               SUB         R7,#0x30
17               SUB         R0,#0x30
18               MUL         R7,R3
19               ADD         R0,R7 ; R0 represents input value with decimal
20               ;Prepreation for the alghoritm
21               MOV         R2,R0
22               BL          InChar
23               MOV         R0,R2
24               MOV         R2,#0x31 ; Represents Fn-1
25               MOV         R3,#0x31 ; Represents Fn-2
26               SUB         R0,#2
27               STR         R2,[R9],#1
28               STR         R3,[R9],#1
29               MOV         R2,#1 ; Represents Fn-1
30               MOV         R3,#1 ; Represents Fn-2
31               BL          fibo  ;It writes mFibo Values to the NUM adress
32  final        LDR         R10,=0x0D
33               STR         R10,[R9],#1
34               LDR         R10,=0x04
35               STR         R10,[R9],#1
36               LDR         R0,=NUM2
37               BL          OutStr
38               B           __main
39               ALIGN
40               ENDP
41
42  fibo         PROC
43               LDR         R5,=NUM
44               MOV         R6,R5
45               LSL         R8,R3,#1
46               ADD         R7,R2,R8
47               MOV         R3,R2
48               MOV         R2,R7
49               MOV         R4,R2
50               CMP         R4,#10
51               BMI         fibo2
52               PUSH        {R0,R2,R3,R7}
53               BL          CONVRT
54               POP         {R0,R2,R3,R7}
55               SUBS        R0,#1
56               BPL         fibo
57               B           final
58
59
60  fibo2        ADD         R4,R4,#48
61               STR         R4,[R9],#1
62               SUBS        R0,#1
63               B           fibo
```

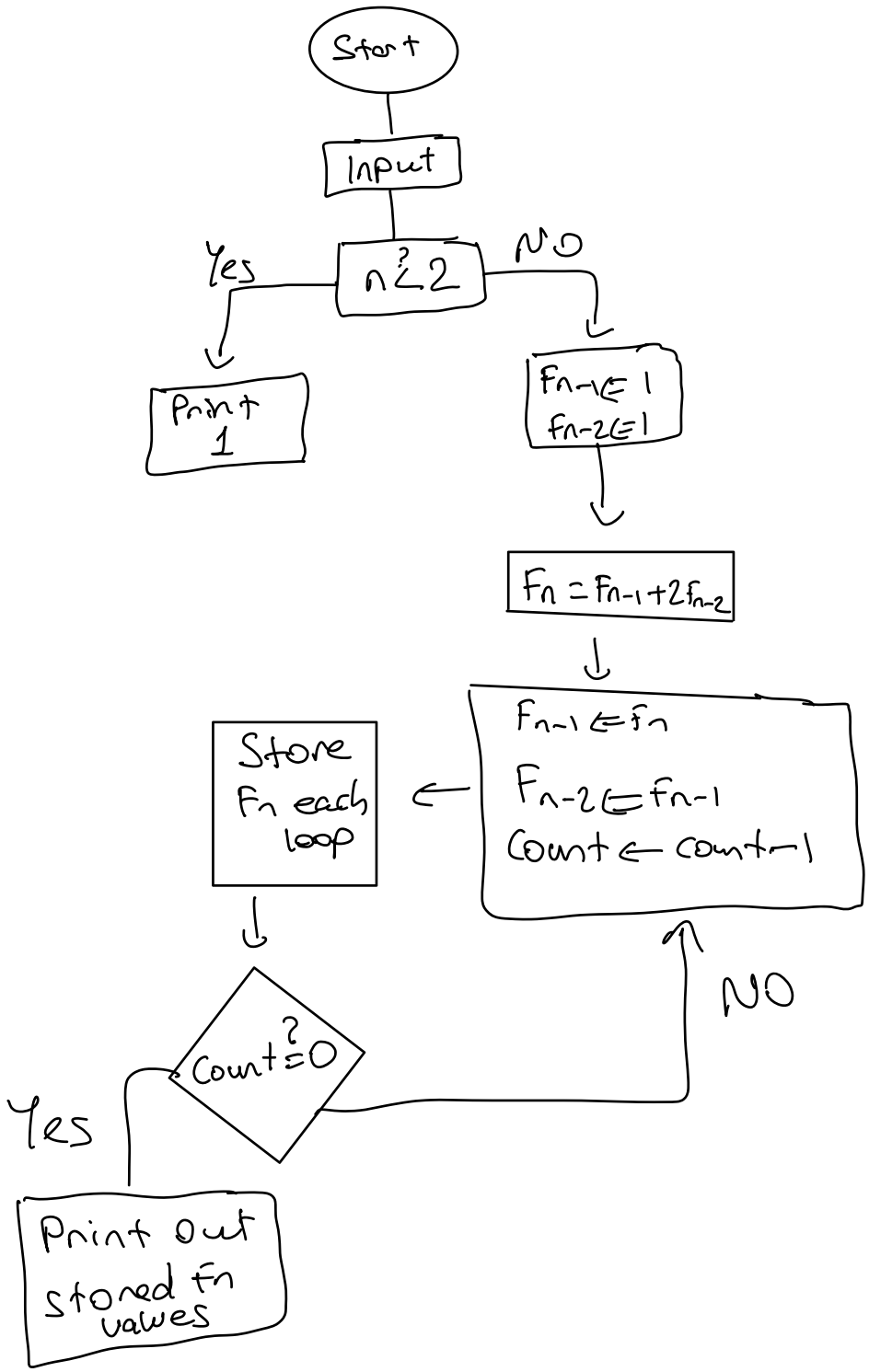
Q1) Flowchart

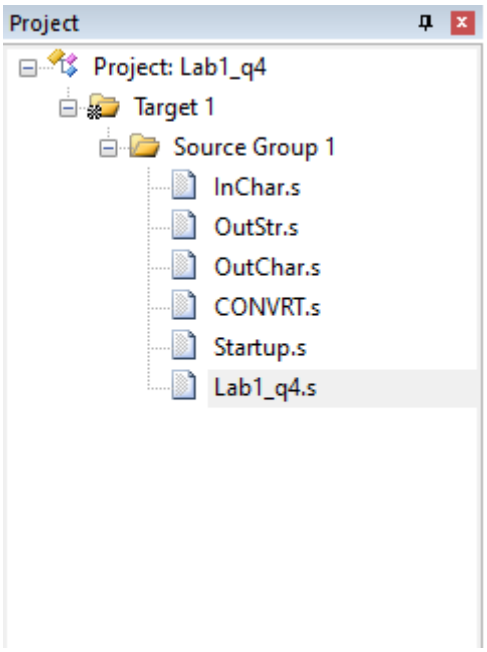


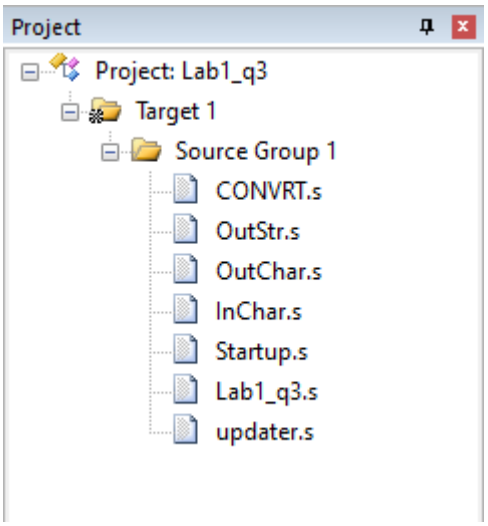
Q3) Flowchart

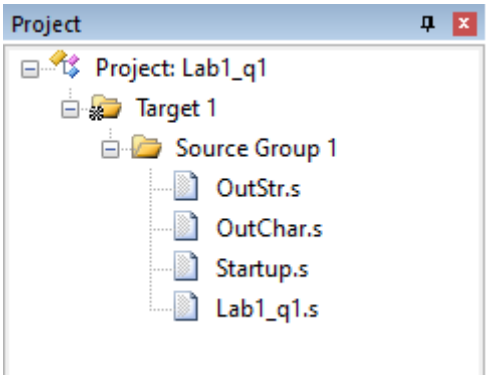


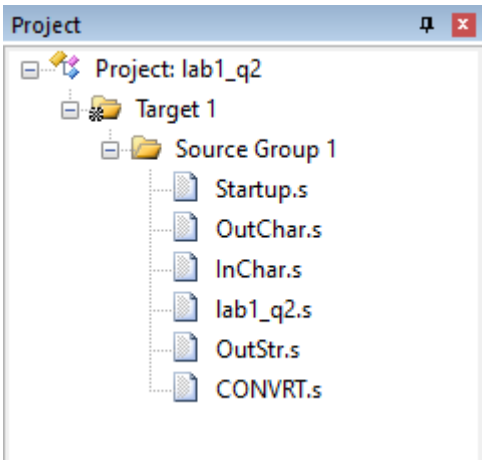
Q4) Flowchart



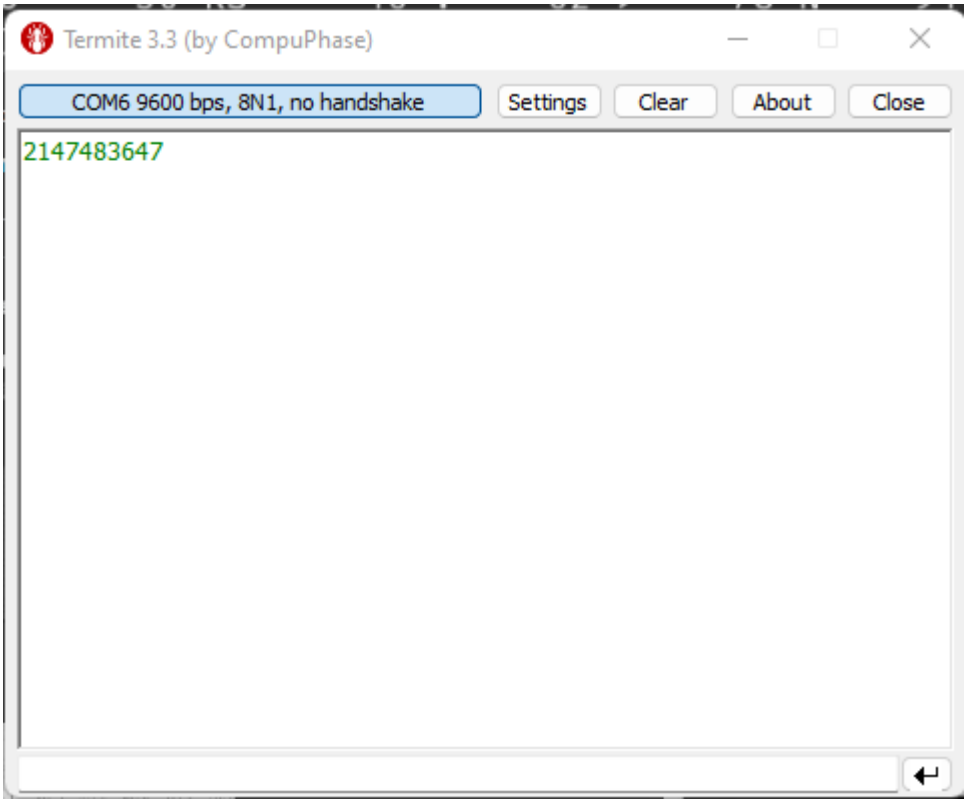

Project
Project: Lab1_q4
Target 1
Source Group 1
InChar.s
OutStr.s
OutChar.s
CONVRT.s
Startup.s
Lab1_q4.s


Project
Project: Lab1_q3
Target 1
Source Group 1
CONVRT.s
OutStr.s
OutChar.s
InChar.s
Startup.s
Lab1_q3.s
updater.s

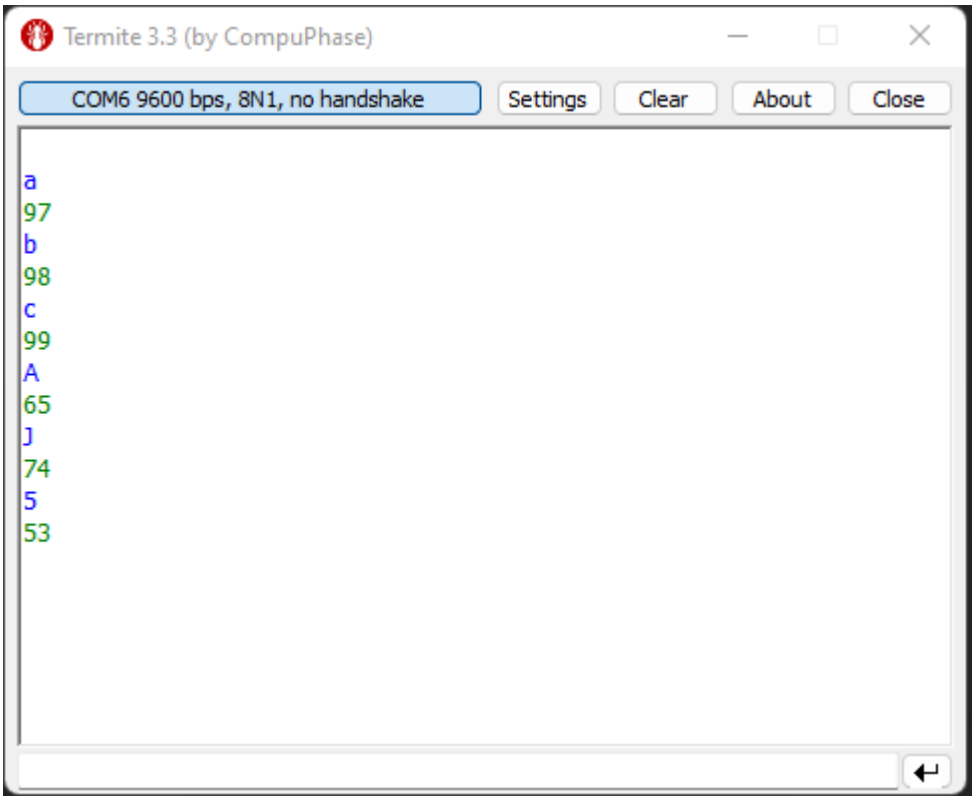

Project
Project: Lab1_q1
Target 1
Source Group 1
OutStr.s
OutChar.s
Startup.s
Lab1_q1.s


Project
Project: lab1_q2
Target 1
Source Group 1
Startup.s
OutChar.s
InChar.s
lab1_q2.s
OutStr.s
CONVRT.s

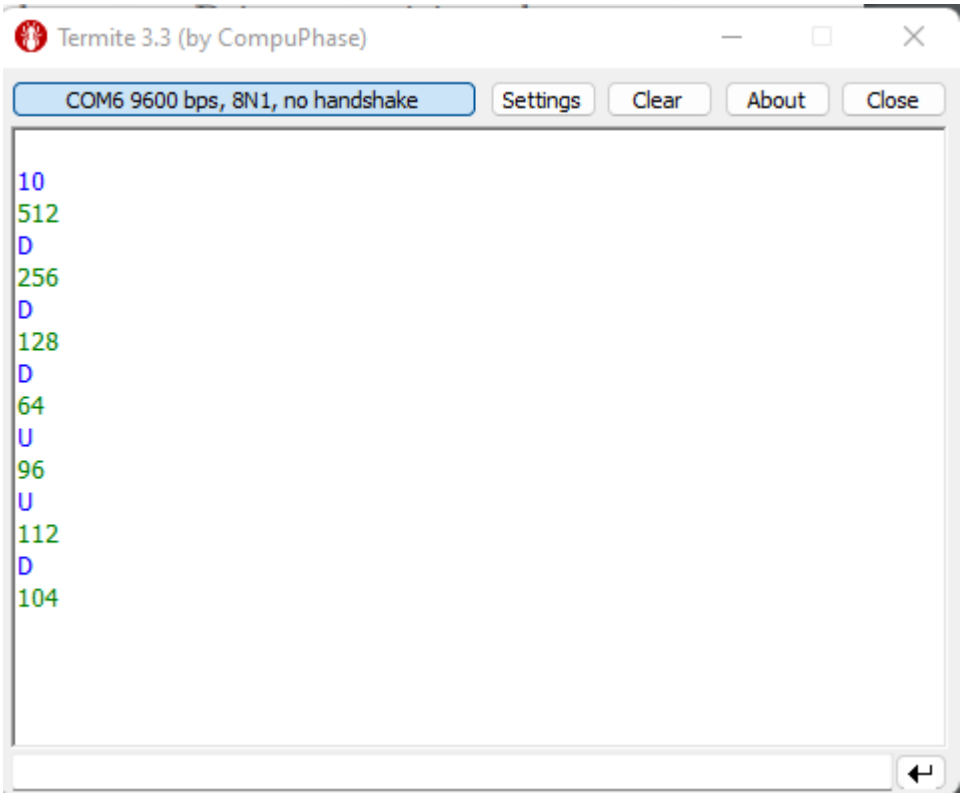
1)


Termite 3.3 (by CompuPhase)
COM6 9600 bps, 8N1, no handshake
Settings
Clear
About
Close
2147483647

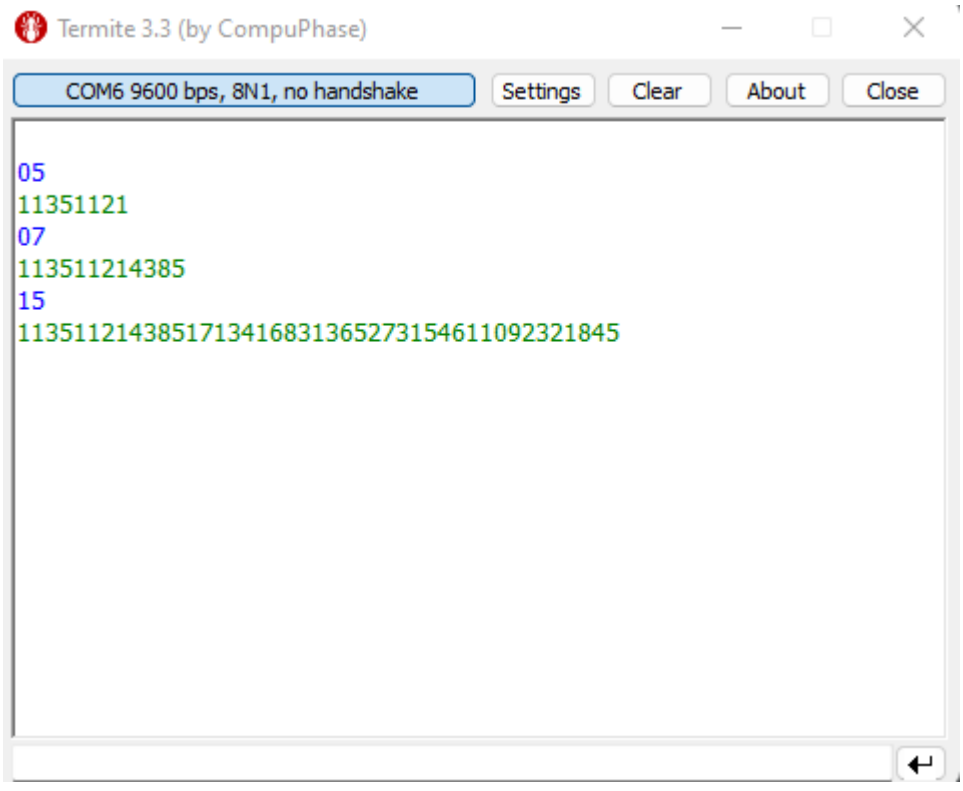
2)


Termite 3.3 (by CompuPhase)
COM6 9600 bps, 8N1, no handshake
Settings
Clear
About
Close
a
97
b
98
c
99
A
65
J
74
5
53

3)


Termite 3.3 (by CompuPhase)
COM6 9600 bps, 8N1, no handshake
Settings
Clear
About
Close
10
512
D
256
D
128
D
64
U
96
U
112
D
104

4)


Termite 3.3 (by CompuPhase)
COM6 9600 bps, 8N1, no handshake
Settings
Clear
About
Close
05
11351121
07
113511214385
15
1135112143851713416831365273154611092321845