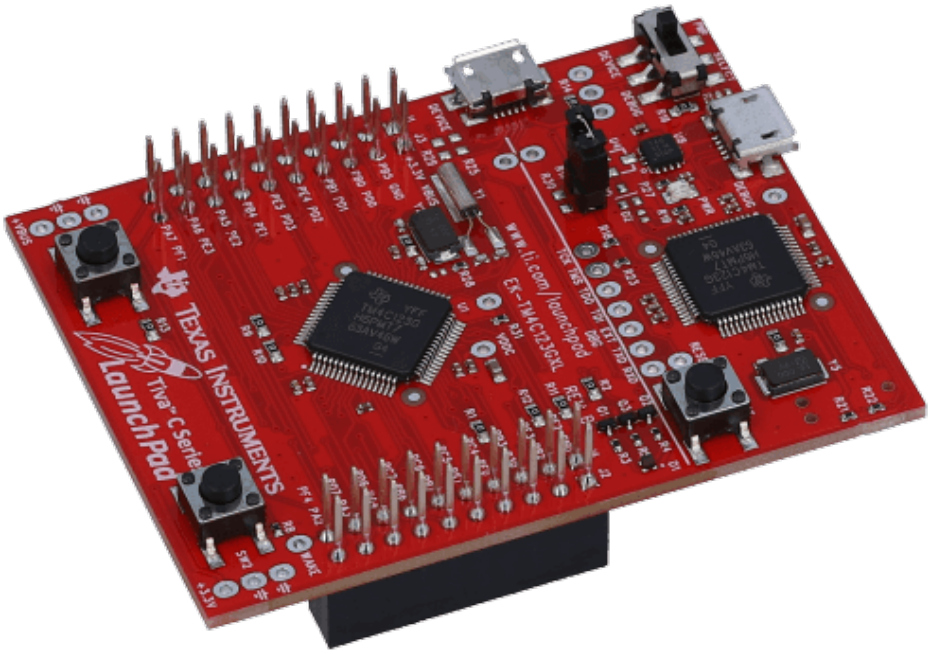
ORTA DOĞU TEKNİK ÜNİVERSİTESİ
MIDDLE EAST TECHNICAL UNIVERSITY

EE447- Introduction to Microprocessors Laboratory With Assembly Programming

(Preliminary Work 1)

EXPERIMENTAL WORK NO: 1

1st Group Member: Uğur SAMANCI - 2398915
2nd Group Member: Barış GÜZEL - 2304764



Question 1) Conversion of hexadecimal to decimal

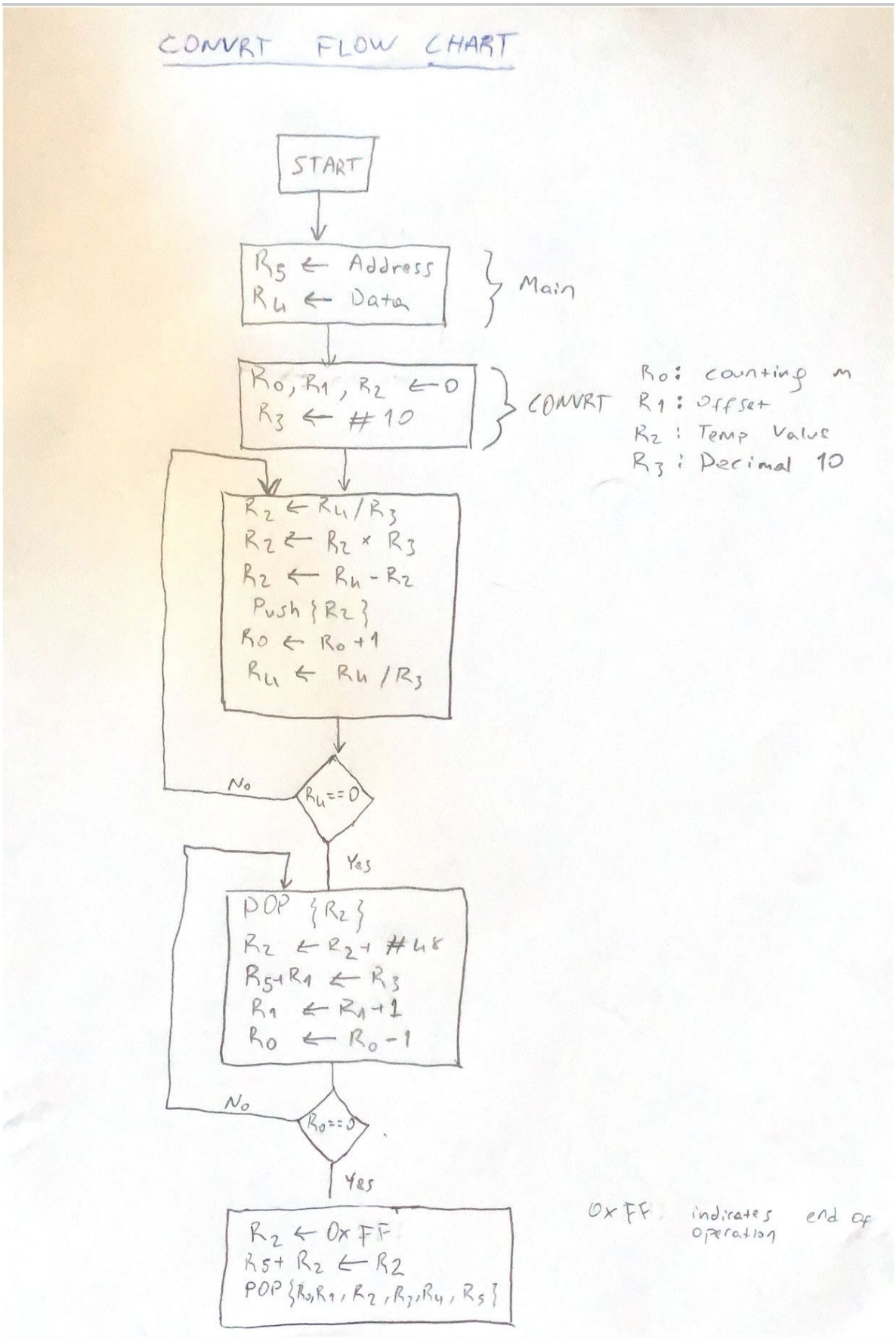


Figure 1:Handwriten Flowchart for question 1

Related Screen for Code – question 1

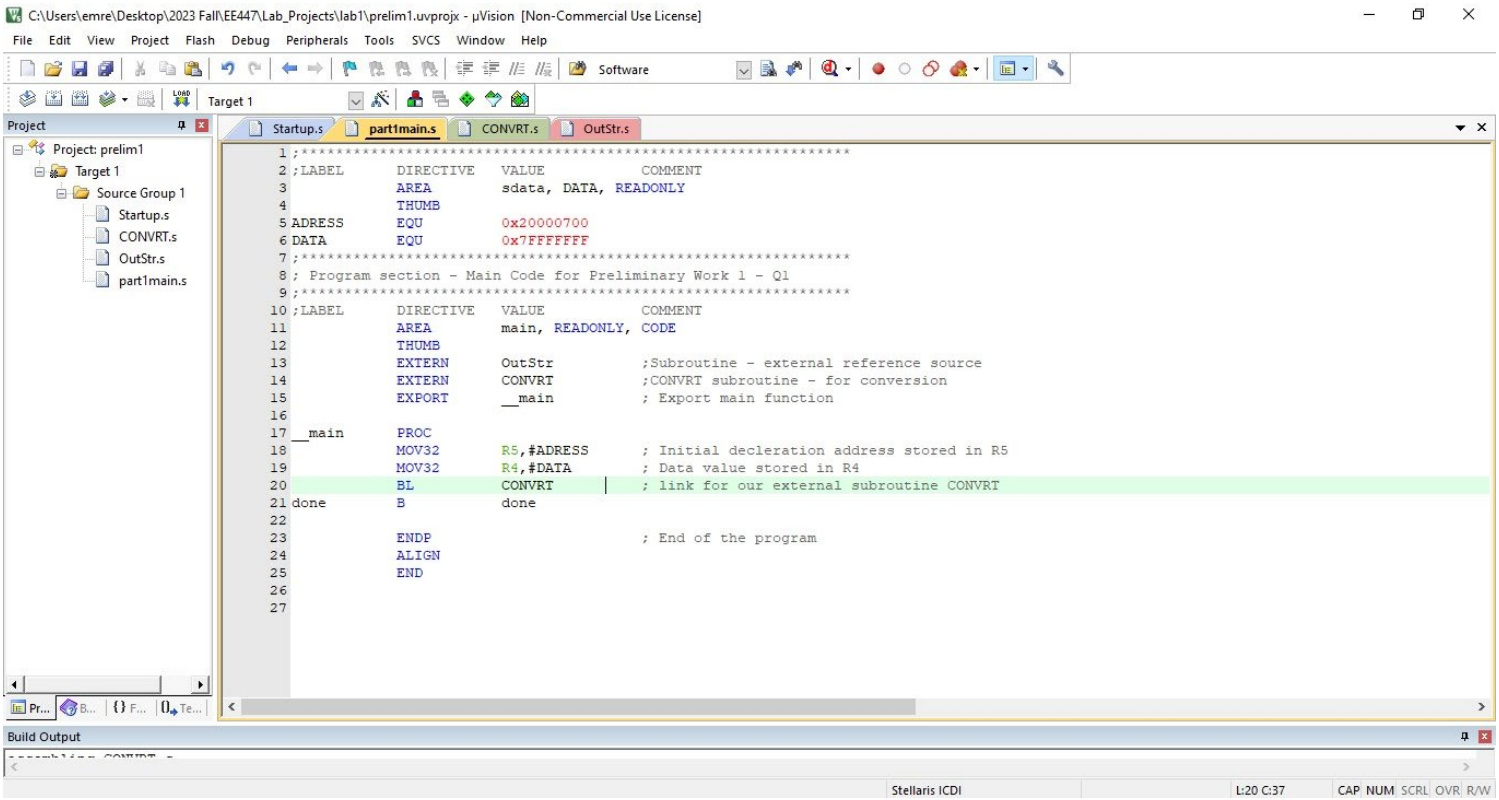


Figure 2: Main Code screen of Question 1

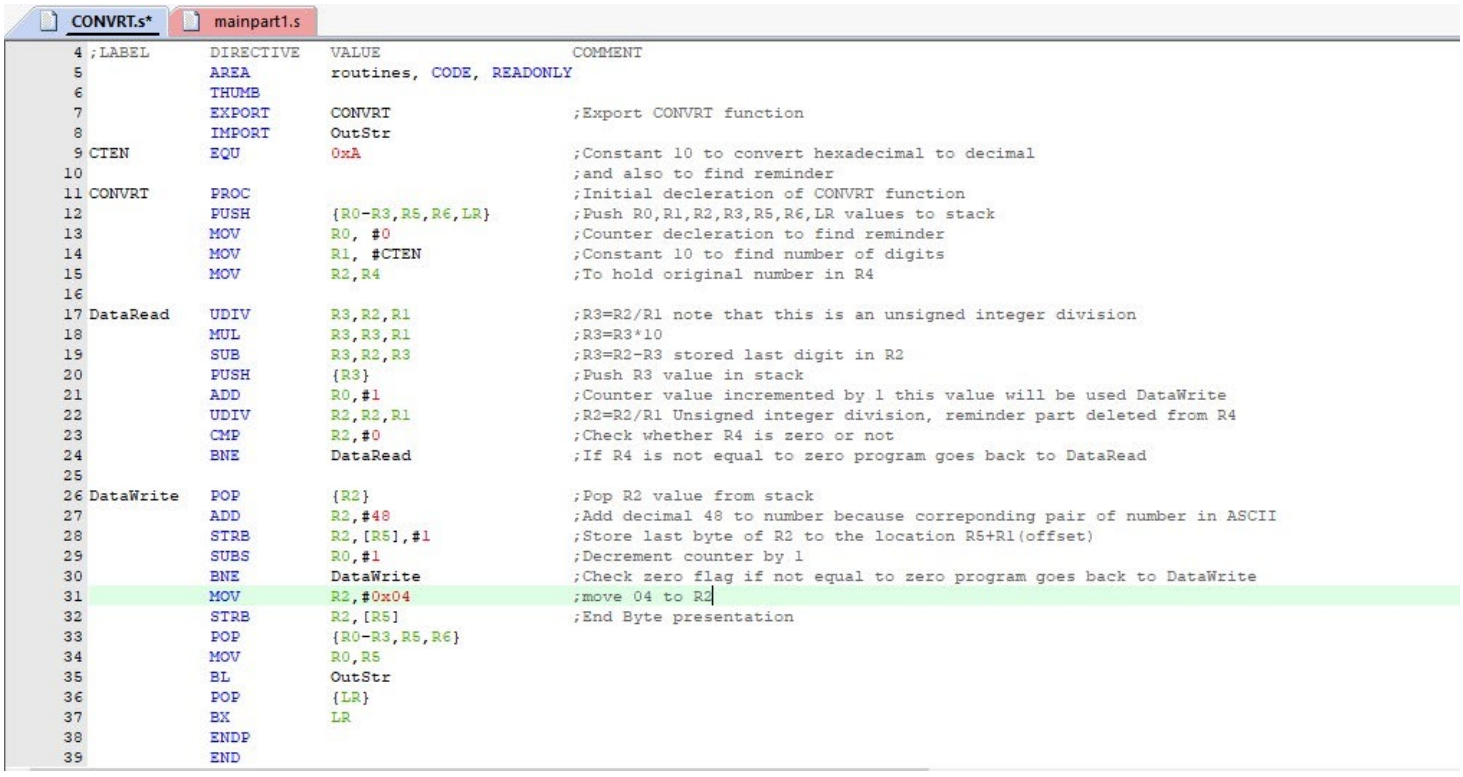


Figure 3: Convert function code screen

For – DATA = 0x7FFFFFFF

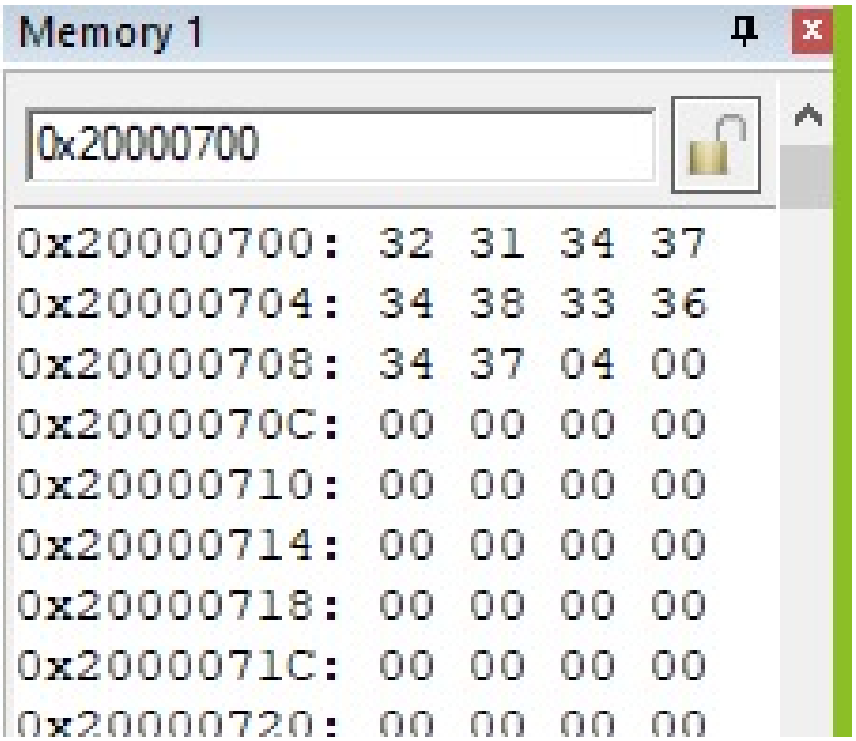


Figure 4:Memory and corresponding results for 0x7fffffff

For – DATA = 0xA

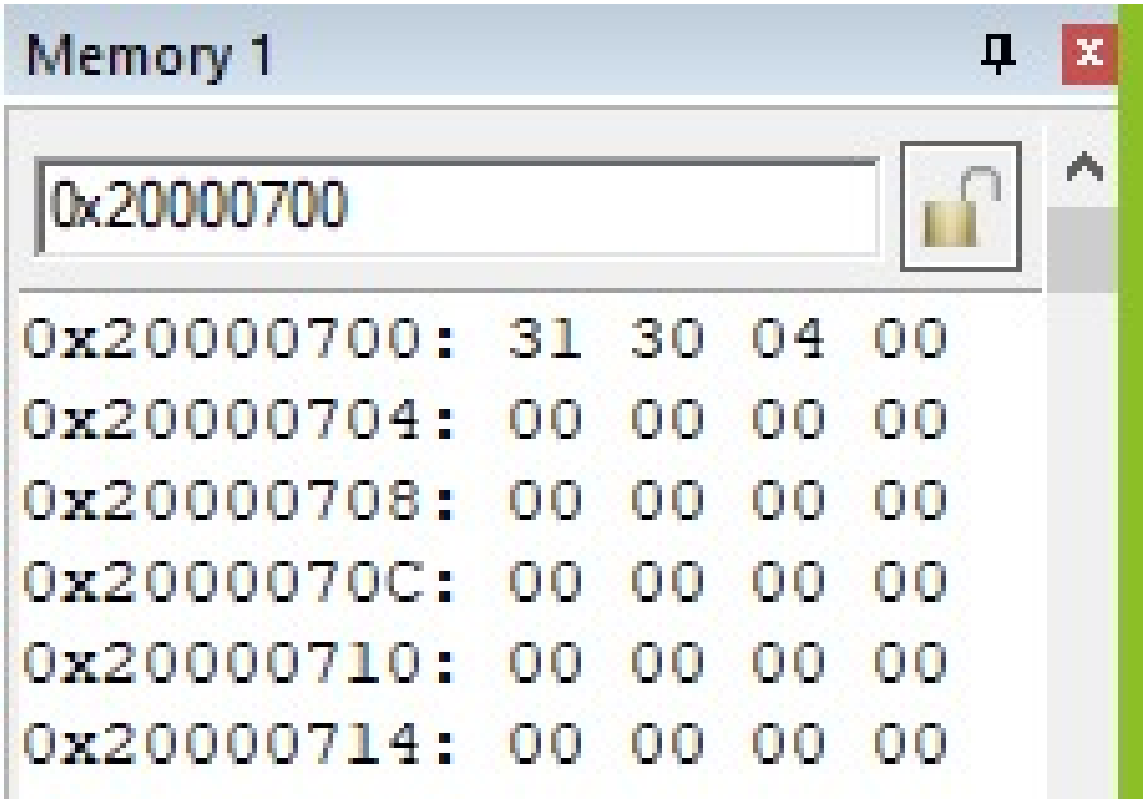


Figure 5:Memory and corresponding results for 0x10

For – DATA =0x0

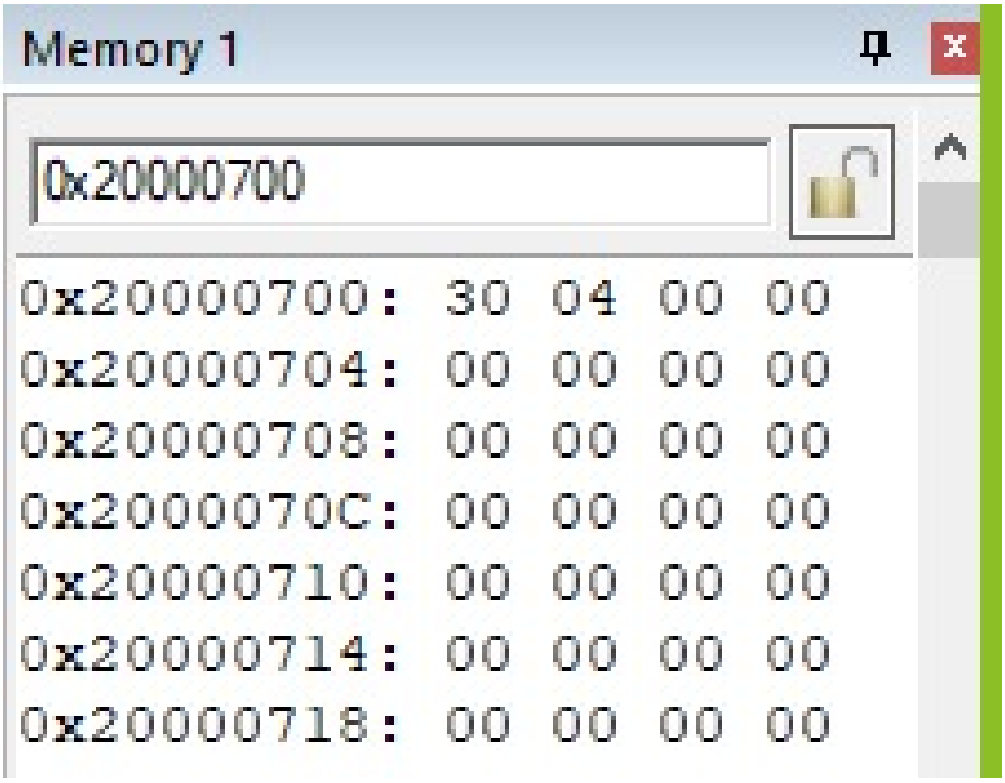


Figure 6:Memory and corresponding results for 0x0

Question 2) Calling function with user prompt

mainpart2.s CONVRT.s

```
;LABEL      DIRECTIVE   VALUE           COMMENT
           AREA        sdata, DATA, READONLY
           THUMB
ADRESS     EQU         0x20000700
NUM        DCD         356
;*****************************************************************
; Program section - Main Code for Preliminary Work 1 - Q2
;*****************************************************************
;LABEL      DIRECTIVE   VALUE           COMMENT
           AREA        main, READONLY, CODE
           THUMB
           EXTERN      OutStr          ;Subroutine - external reference source
           EXTERN      CONVRT          ;CONVRT subroutine - for conversion
           EXTERN      InChar
           EXPORT      __main

__main     PROC

LOOP1      MOV32       R4,#NUM
           LDR         R4,[R4]
           MOV32       R5,#ADRESS
           BL          InChar          ;Enter Character to Terminal
           BL          CONVRT
           B           LOOP1

           ENDP                        ; End of the program
           ALIGN
           END
```

Figure 7:Question 2 main code

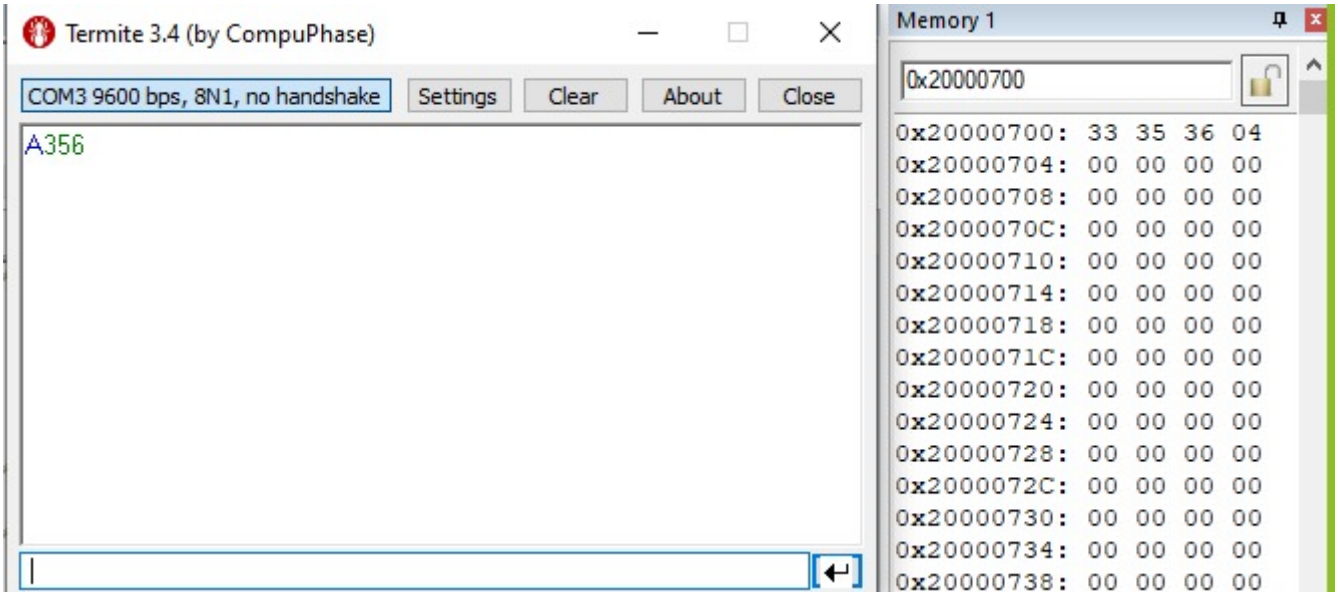


Figure 8: Termite screen for different values stored in DATA

Someone from figure 8 can see that, whenever we changed the data we get new decimal result according to data. Here we pass number to R4 register before the convert function to write the number in the desired memory location. We tried it with a new value and show it on figure 9.

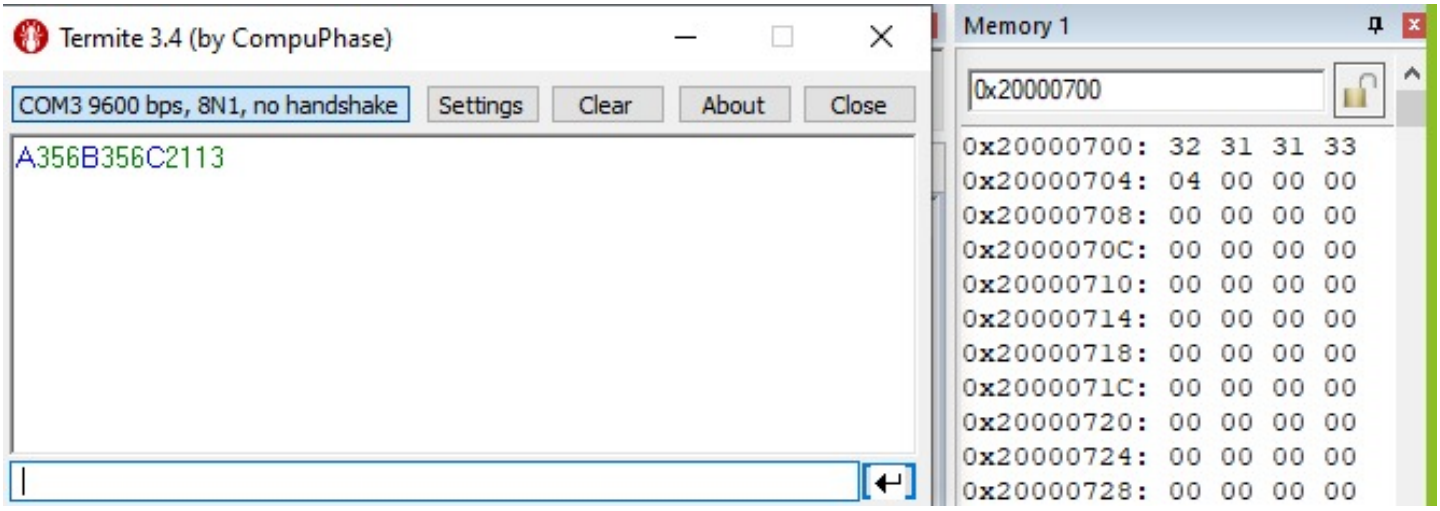


Figure 9: Termite screen for different values stored in DATA

Question 3)

Question 4) Modified Fibonacci

```
;LABEL      DIRECTIVE   VALUE           COMMENT
            AREA        sdata, DATA, READONLY
            THUMB
ADRESS      EQU         0x20000700
DTEN        EQU         0xA
NUM         DCD         2113
;*********************************************************************
; Program section - Main Code for Preliminary Work 1 - Q4
;*********************************************************************
;LABEL      DIRECTIVE   VALUE           COMMENT
            AREA        main, READONLY, CODE
            THUMB
            EXTERN      OutStr          ;Subroutine - external reference source
            EXTERN      CONVRT          ;CONVRT subroutine - for conversion
            EXTERN      InChar
            EXPORT      __main

__main      PROC

start       BL          InChar
            MOV32       R2,#DTEN        ;Decimal 10 move to R2
            MOV32       R5, #ADRESS     ;Initial adress for R5
            SUB         R1,R0,#0x30     ;From ASCII to Integer
            BL          InChar
            SUB         R0,#0x30
            MOV         R6,#0x0
            ADD         R6,R2,R0
            MUL         R1,R1,R6
            MOV         R7, R5

            MOV         R8, #0x1
            MOV         R4, R8
            BL          CONVRT
            MOV         R10, #0x1
```

Figure 9:Main code section1 for question 4

```
            MUL         R1,R1,R6
            MOV         R7, R5

            MOV         R8, #0x1
            MOV         R4, R8
            BL          CONVRT
            MOV         R10, #0x1
            MOV         R4, R10
            BL          CONVRT
            SUB         R1, #0x1

mfibon      SUBS        R1, #0x1
            BEQ         goto
            ADD         R8,R8
            ADD         R3, R10, R8
            MOV         R4, R3
            BL          CONVRT
            MOV         R8, R10
            MOV         R10, R3
            B           mfibon

goto        MOV         R9, #0x04
            STRB        R9, [R5]
            MOV         R0, R7
            BL          OutStr
            B           start

            ENDP
;*********************************************************************
; End of the program  section
;*********************************************************************
;LABEL      DIRECTIVE       VALUE                           COMMENT
            ALIGN
            END
```

Figure 10: Main code section2 for question 4

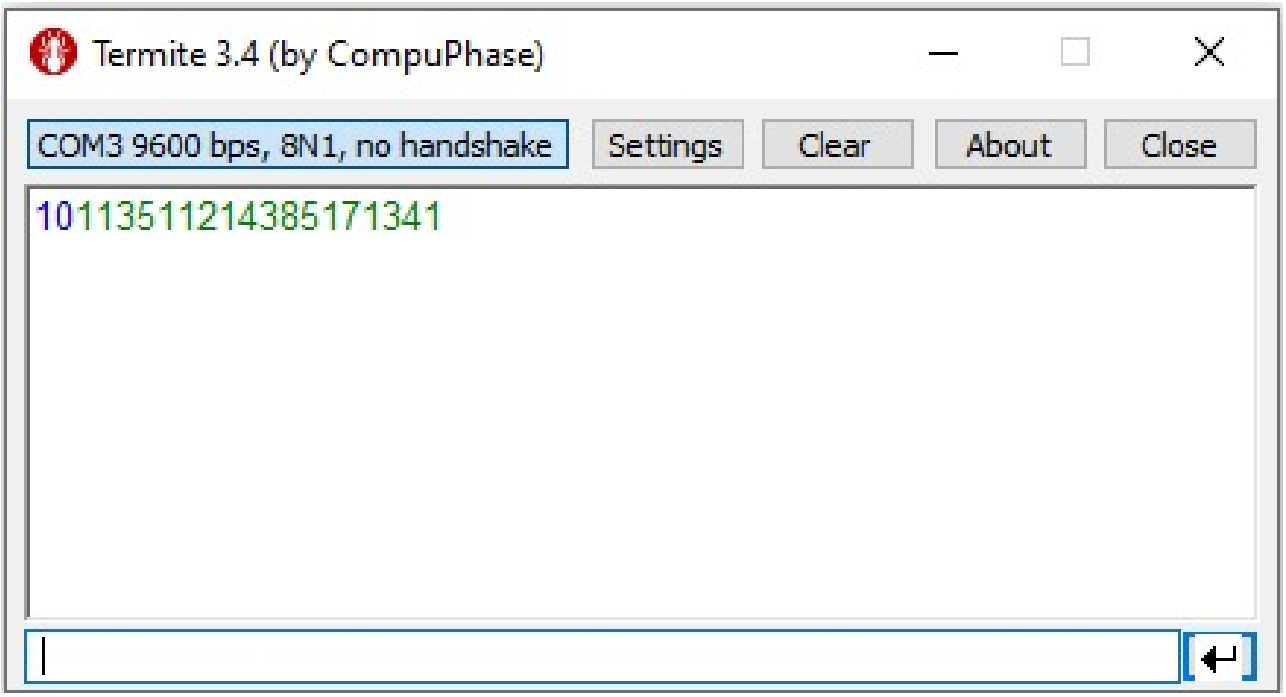


Figure 11: Termite part of the question 4