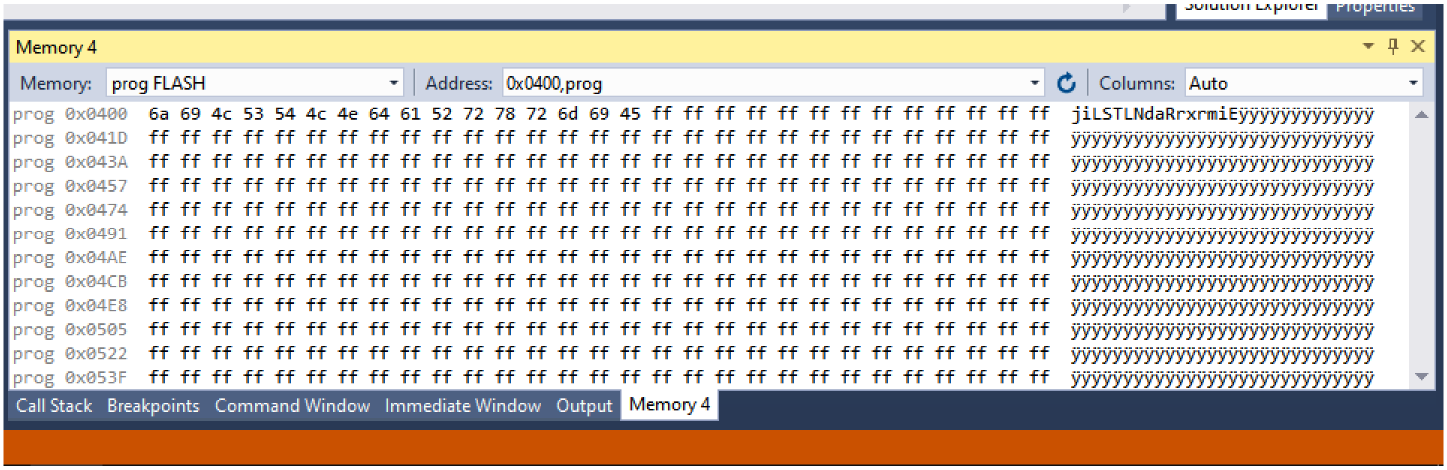
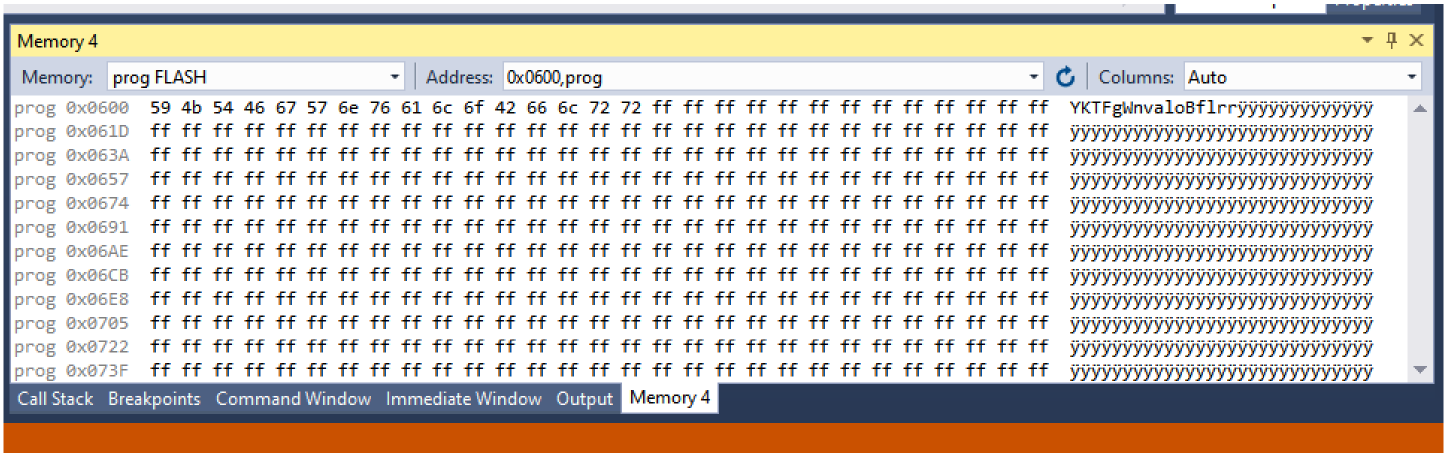


Memory: prog FLASH　Address: 0x0400,prog　Columns: Auto

```
prog 0x0400  6a 69 4c 53 54 4c 4e 64 61 52 72 78 72 6d 69 45 ff ff ff ff ff ff ff ff ff ff ff ff ff  jiLSTLNdaRrxrmiEÿÿÿÿÿÿÿÿÿÿÿÿÿ
prog 0x041D  ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff  ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
prog 0x043A  ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff  ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
prog 0x0457  ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff  ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
prog 0x0474  ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff  ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
prog 0x0491  ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff  ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
prog 0x04AE  ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff  ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
prog 0x04CB  ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff  ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
prog 0x04E8  ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff  ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
prog 0x0505  ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff  ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
prog 0x0522  ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff  ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
prog 0x053F  ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff  ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
```



Memory: prog FLASH　Address: 0x0600,prog　Columns: Auto

```
prog 0x0600  59 4b 54 46 67 57 6e 76 61 6c 6f 42 66 6c 72 72 ff ff ff ff ff ff ff ff ff ff ff ff ff  YKTFgWnvaloBflrrÿÿÿÿÿÿÿÿÿÿÿÿÿ
prog 0x061D  ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff  ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
prog 0x063A  ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff  ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
prog 0x0657  ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff  ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
prog 0x0674  ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff  ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
prog 0x0691  ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff  ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
prog 0x06AE  ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff  ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
prog 0x06CB  ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff  ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
prog 0x06E8  ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff  ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
prog 0x0705  ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff  ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
prog 0x0722  ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff  ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
prog 0x073F  ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff  ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
```

```
;
; DA1.asm
;
; Created: 1/30/2025 12:12:03 PM
; Author : venki
;

.ORG 0
   ; Initialize the stack pointer
   LDI R20, HIGH(RAMEND)
   OUT SPH, R20
   LDI R20, LOW(RAMEND)
   OUT SPL, R20

   ; Initialize X-pointer for EEPROM Key retrieval
   LDI XH, HIGH(0x00)
   LDI XL, LOW(0x00)

   ; Initialize the Y-pointer for Decrypted Message starting @ 0x200
   LDI YH, HIGH(0x200)
   LDI YL, LOW(0x200)

   ; Initialize the Z-pointer for Raw message starting @ 0x400
   LDI ZH, HIGH(KEY<<1)    ; Point to KEY in Program Memory
   LDI ZL, LOW(KEY<<1)

   LDI R16, 16          ; Limit the loop to 16 bytes for key

L1:
   LPM R20, Z+

   CALL STORE_IN_EEPROM   ; R20 has the EEPROM Data

   INC XL              ; increment to next EEPROM Location

   DEC R16

   BRNE L1

   LDI R16, 16         ; Reset counter


LOAD_KEY:

   CALL LOAD_FROM_EEPROM

   MOV R10, R20   ; k[0]

   INC XL        ; Move to next EEPROM byte


   CALL LOAD_FROM_EEPROM

   MOV R11, R20   ; k[1]

   INC XL


   CALL LOAD_FROM_EEPROM

   MOV R12, R20   ; k[2]

   INC XL


   CALL LOAD_FROM_EEPROM

   MOV R13, R20   ; k[3]

   INC XL


   DEC R16       ; counter--

   BRNE LOAD_KEY   ; repeat until all 16 bytes of key are loaded


   LDI R16, 32    ; 32 iterations
```

```
    ; TEA algorithm initialization
    LDI R20, 0x9E   ; Delta = 0x9E3779B9
    LDI R21, 0x37
    LDI R22, 0x79
    LDI R23, 0xB9   ; Delta = R23:R22:R21:R20

    LDI R24, 0x00   ; sum = 0
    LDI R25, 0x00
    LDI R26, 0x00
    LDI R27, 0x00

LOOP:
    ; sum += delta
    ADD R24, R20
    ADC R25, R21
    ADC R26, R22
    ADC R27, R23

    ; Load y (32-bit) from SRAM into R2-R5
    LD  R2, Y+    ; y[0]
    LD  R3, Y+    ; y[1]
    LD  R4, Y+    ; y[2]
    LD  R5, Y+    ; y[3]

    ; Load z (32-bit) from SRAM into R6-R9
    LD  R6, Y+    ; z[0]
    LD  R7, Y+    ; z[1]
    LD  R8, Y+    ; z[2]
    LD  R9, Y+    ; z[3]
```

```
; (z << 4) on lower 16 bits of z (R6, R7)
MOV R18, R6
MOV R19, R7
LSL R18
ROL R19
LSL R18
ROL R19
LSL R18
ROL R19
LSL R18
ROL R19
ADD R18, R10   ; Add key[0]
ADC R19, R11

; (z >> 5) on lower 16 bits of z (R6, R7)
MOV R14, R6
MOV R15, R7
LSR R14
LSR R14
LSR R14
LSR R14
LSR R14
ADD R14, R6   ; Add key[1]
ADC R15, R7

; XOR for y update: update lower 16 bits of y (R2,R3)
EOR R18, R6
EOR R19, R7
EOR R18, R14
EOR R19, R15

; Store updated y lower half
ST  Y+, R18
ST  Y+, R19

; (y << 4) for z update on lower 16 bits of y (R2, R3)
MOV R30, R2
MOV R31, R3
LSL R30
ROL R31
LSL R30
ROL R31
LSL R30
ROL R31
LSL R30
ROL R31
ADD R30, R12   ; Add key[2]
ADC R31, R13

; (y >> 5) for z update on lower 16 bits of y (R2, R3)
MOV R0, R2
MOV R1, R3
LSR R0
LSR R0
LSR R0
LSR R0
LSR R0
```

```
   ADD R0, R2   ; Add key[3]
   ADC R1, R3

   ; XOR for z update: combine results with original y lower half (R2,R3)
   EOR R30, R2
   EOR R31, R3
   EOR R30, R0
   EOR R31, R1

   ; Store updated z lower half
   ST  Y+, R30
   ST  Y+, R31

   ; Decrement counter
   DEC R16
   CPI R16, 0
   BREQ EXIT     ; Exit when 32 rounds are done
   JMP LOOP      ; Jump back to start of loop

EXIT:
   RET

LOAD_FROM_EEPROM:
   SBIC EECR, EEPE
   RJMP LOAD_FROM_EEPROM
   OUT EEARH, XH
   OUT EEARL, XL
   SBI EECR, EERE
   IN R20, EEDR
   RET

STORE_IN_EEPROM:
   SBIC EECR, EEPE
   RJMP STORE_IN_EEPROM
   OUT EEARH, XH
   OUT EEARL, XL
   OUT EEDR, R20
   SBI EECR, EEMPE
   SBI EECR, EEPE
   RET

; Message: "jiLSTLNdaRrxrmiElGjSeiZBNSIrXEOInKAljICoLQvnCSTuTqApIrpqhyjBNAYy"

.ORG 0x200
MESSAGE: .DB 0x6a, 0x69, 0x4c, 0x53, 0x54, 0x4c, 0x4e, 0x64, 0x61, 0x52, 0x72, 0x78, 0x72, 0x6d, 0x69, 0x45
;.DB 6c, 47, 6a, 53, 65, 69, 5a, 42, 4e, 53, 49, 72, 58, 45, 4f, 49
;.DB 6e, 4b, 41, 6c, 6a, 49, 43, 6f, 4c, 51, 76, 6e, 43, 53, 54, 75
;.DB 54, 71, 41, 70, 49, 72, 70, 71, 68, 79, 6a, 42, 4e, 41, 59, 79

; KEY: "YKTFgWnvaloBflrr"
.ORG 0x300
KEY: .DB 0x59, 0x4b, 0x54, 0x46, 0x67, 0x57, 0x6e, 0x76, 0x61, 0x6c, 0x6f, 0x42, 0x66, 0x6c, 0x72, 0x72
```