

Program 1

Source Code

```
.syntax unified
.section .data
.global main

.align 4
STATUS: .byte 0x97
.align 4
CONTROL: .byte 0xEC
.align 4
PERIPH: .byte 0x15

.text
main:
    // Initialize variables
    ldr r0, =STATUS
    ldr r1, =CONTROL
    ldr r2, =PERIPH

    // Load values from memory
    ldrb r3, [r0]

    // If bit 3 of STATUS is 1, set bit 4 of CONTROL to 1, otherwise set bit 5 of
    // CONTROL to 0.
    tst r3, #8
    it eq
    movseq r4, #0x20
    it ne
    movne r4, #0x10
    ldrb r5, [r1]
    orr r5, r4
    strb r5, [r1]

    // If bit 5 of STATUS is 0 and bit 6 of STATUS is 0, set bit 2 of CONTROL to 0,
    // otherwise set bit 0 of CONTROL to 1.
    tst r3, #0x20
    bne op2_bit6
    tst r3, #0x40
    it eq
    beq op2_bit6
    tst r3, #0x40
    it eq
    movseq r4, #0xFC
    it ne
    movne r4, #0x01
    orr r5, r4
    strb r5, [r1]

op2_bit6:
```

```
// Replace bits 5-2 of PERIPH with the hex digit A, without changing the other four bits of PERIPH.
```

```
mov r4, #0xA0
ldrb r6, [r2]
and r6, #0x03
orr r6, r4
strb r6, [r2]
```

```
// Effectively halts the program.
```

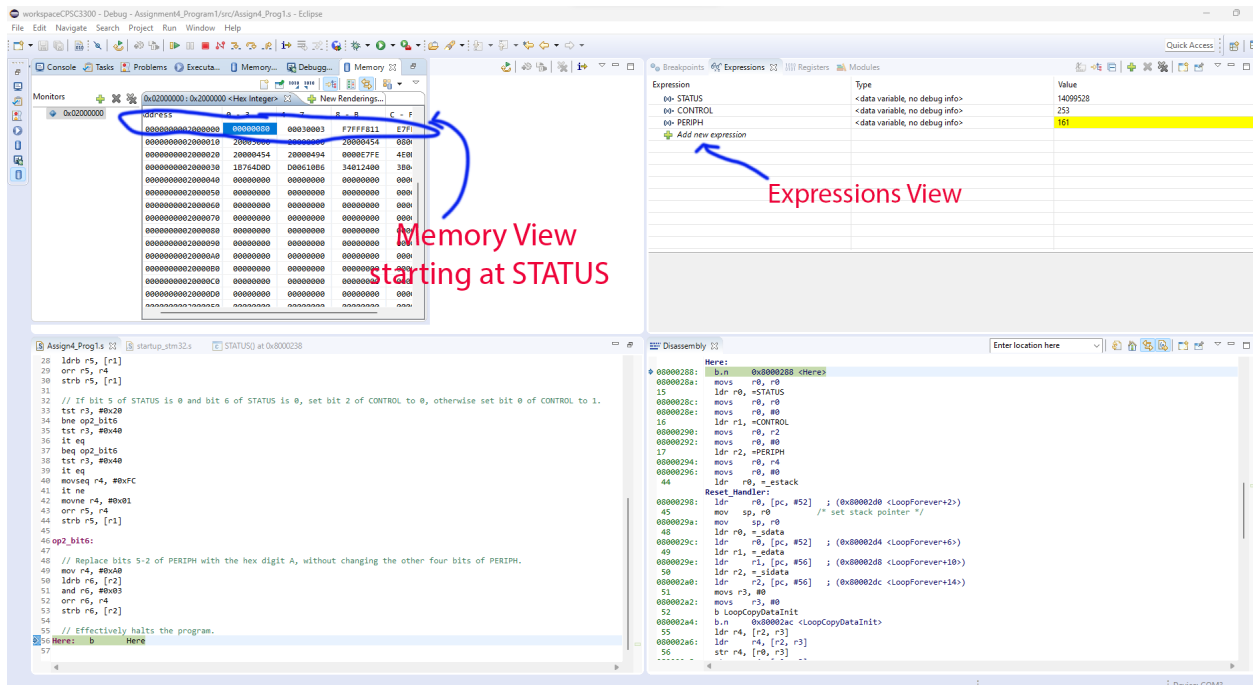
Here: b Here

The screenshot shows the Eclipse IDE interface with the following views:

- Memory view:** Displays memory addresses and their corresponding values. A blue circle highlights the address 0x02000000, which contains the value 0x00000000. A red arrow points from the text "Memory view" to this circle.
- Expressions view:** Displays the values of variables. The PERIPH variable is highlighted in yellow, showing a value of 161. A red arrow points from the text "Expressions View" to this view.
- Disassembly view:** Shows the assembly code for the program. The instruction at address 0x00002026 is highlighted, which is the instruction that effectively halts the program.

The assembly code in the Disassembly view is as follows:

```
0x00002026: movs r0, r0
0x00002027: ldr r0, =STATUS
0x00002028: movs r0, r0
0x00002029: ldr r1, =CONTROL
0x0000202a: movs r0, r0
0x0000202b: ldr r1, =PERIPH
0x0000202c: movs r0, r0
0x0000202d: ldr r2, =_edata
0x0000202e: ldr r2, [pc, #56] ; (0x0000202c <LoopForever+14>)
0x0000202f: movs r3, r0
0x00002030: b LoopCopyDataInit
0x00002031: b.n 0x0000202c <LoopCopyDataInit>
0x00002032: ldr r4, [r2, r3]
0x00002033: str r4, [r0, r3]
```



Program 2

Source Code

.syntax unified

```
.equ NULL, 0x00
.equ MAX_LENGTH, 20
.equ ASCII_UPPERCASE_A, 'A'
.equ ASCII_UPPERCASE_Z, 'Z'
.equ ASCII_LOWERCASE_A, 'a'
.equ ASCII_LOWERCASE_Z, 'z'
```

.global main

main:

```
// Initialize registers
ldr r0, =0x02000000 // Address of the string
mov r1, #0 // Index of the current character
```

convert_loop:

```
// Load the current character
ldrb r2, [r0, r1]
```

```
// Check if the character is null
cmp r2, #NULL
beq exit
```

```
// Check if the character is lowercase
cmp r2, #ASCII_LOWERCASE_A
blt increment_index
cmp r2, #ASCII_LOWERCASE_Z
bgt increment_index
```

```

// Convert the character to uppercase
sub r2, r2, #'a'-'A'
strb r2, [r0, r1]

increment_index:
// Increment the index
add r1, r1, #1

// Check if the maximum length has been reached
cmp r1, #MAX_LENGTH
bge exit

// Continue to the next character
b convert_loop

// Effectively halts the program.
Here:    b        Here

```

Run number 1 with this set: Includes the null char and 19 other char

set {char[20]}0x02000000 = "This is ninTeEnChar"

Run number 2 with this set: Includes the null char and 12 other char

set {char[13]}0x02000000 = "ThirTEenChar"

Program 3

Source Code

```

.syntax unified
.data
PACK:  .word 0x13587609
STRG:  .byte 8

.align 4
.text
.global main
main:

LDR    r0, =PACK
LDR    r1, =STRG

```



```
str r0, [r1]
```

```
// End the program
```

```
mov r0, #0
```

```
bx lr
```

```
// Subroutine to count the number of non-zero positive numbers in an array of signed  
32-bit integers
```

```
count_nonzero_positives:
```

```
mov r2, #0
```

```
loop:
```

```
ldr r3, [r0], #4 // Load the next element of array. increment the pointer next  
element
```

```
cmp r3, #0
```

```
// Compare the loaded value with zero
```

```
blt skip
```

```
// If it's negative, skip to the next iteration
```

```
add r2, r2, #1
```

```
skip:
```

```
subs r1, r1, #1 // Decrement the size counter
```

```
bne loop
```

```
// If there are still elements left, repeat the loop
```

```
mov r0, r2
```

```
bx lr
```

The screenshot displays the Eclipse IDE interface for debugging an ARM assembly program. The **Expressions View** on the right shows two expressions: `NPOS1` and `NPOS2`, both with a value of 12. The **Memory View** at the bottom shows a memory dump starting at address 0x02000000. A blue circle highlights a specific memory location, and a blue arrow points to it from the label "Memory View".

Expressions View

Expression	Type	Value
NPOS1	<data variable, no debug info>	12
NPOS2	<data variable, no debug info>	12

Memory View

Address	Hex	ASCII	Comment
0x02000000	00000000		
0x02000001	00000000		
0x02000002	00000000		
0x02000003	00000000		
0x02000004	00000000		
0x02000005	00000000		
0x02000006	00000000		
0x02000007	00000000		
0x02000008	00000000		
0x02000009	00000000		
0x0200000A	00000000		
0x0200000B	00000000		
0x0200000C	00000000		
0x0200000D	00000000		
0x0200000E	00000000		
0x0200000F	00000000		
0x02000010	00000000		
0x02000011	00000000		
0x02000012	00000000		
0x02000013	00000000		
0x02000014	00000000		
0x02000015	00000000		
0x02000016	00000000		
0x02000017	00000000		
0x02000018	00000000		
0x02000019	00000000		
0x0200001A	00000000		
0x0200001B	00000000		
0x0200001C	00000000		
0x0200001D	00000000		
0x0200001E	00000000		
0x0200001F	00000000		
0x02000020	00000000		
0x02000021	00000000		
0x02000022	00000000		
0x02000023	00000000		
0x02000024	00000000		
0x02000025	00000000		
0x02000026	00000000		
0x02000027	00000000		
0x02000028	00000000		
0x02000029	00000000		
0x0200002A	00000000		
0x0200002B	00000000		
0x0200002C	00000000		
0x0200002D	00000000		
0x0200002E	00000000		
0x0200002F	00000000		
0x02000030	00000000		
0x02000031	00000000		
0x02000032	00000000		
0x02000033	00000000		
0x02000034	00000000		
0x02000035	00000000		
0x02000036	00000000		
0x02000037	00000000		
0x02000038	00000000		
0x02000039	00000000		
0x0200003A	00000000		
0x0200003B	00000000		
0x0200003C	00000000		
0x0200003D	00000000		
0x0200003E	00000000		
0x0200003F	00000000		
0x02000040	00000000		
0x02000041	00000000		
0x02000042	00000000		
0x02000043	00000000		
0x02000044	00000000		
0x02000045	00000000		
0x02000046	00000000		
0x02000047	00000000		
0x02000048	00000000		
0x02000049	00000000		
0x0200004A	00000000		
0x0200004B	00000000		
0x0200004C	00000000		
0x0200004D	00000000		
0x0200004E	00000000		
0x0200004F	00000000		
0x02000050	00000000		
0x02000051	00000000		
0x02000052	00000000		
0x02000053	00000000		
0x02000054	00000000		
0x02000055	00000000		
0x02000056	00000000		
0x02000057	00000000		
0x02000058	00000000		
0x02000059	00000000		
0x0200005A	00000000		
0x0200005B	00000000		
0x0200005C	00000000		
0x0200005D	00000000		
0x0200005E	00000000		
0x0200005F	00000000		
0x02000060	00000000		
0x02000061	00000000		
0x02000062	00000000		
0x02000063	00000000		
0x02000064	00000000		
0x02000065	00000000		
0x02000066	00000000		
0x02000067	00000000		
0x02000068	00000000		
0x02000069	00000000		
0x0200006A	00000000		
0x0200006B	00000000		
0x0200006C	00000000		
0x0200006D	00000000		
0x0200006E	00000000		
0x0200006F	00000000		
0x02000070	00000000		
0x02000071	00000000		
0x02000072	00000000		
0x02000073	00000000		
0x02000074	00000000		
0x02000075	00000000		
0x02000076	00000000		
0x02000077	00000000		
0x02000078	00000000		
0x02000079	00000000		
0x0200007A	00000000		
0x0200007B	00000000		
0x0200007C	00000000		
0x0200007D	00000000		
0x0200007E	00000000		
0x0200007F	00000000		
0x02000080	00000000		
0x02000081	00000000		
0x02000082	00000000		
0x02000083	00000000		
0x02000084	00000000		
0x02000085	00000000		
0x02000086	00000000		
0x02000087	00000000		
0x02000088	00000000		
0x02000089	00000000		
0x0200008A	00000000		
0x0200008B	00000000		
0x0200008C	00000000		
0x0200008D	00000000		
0x0200008E	00000000		
0x0200008F	00000000		
0x02000090	00000000		
0x02000091	00000000		
0x02000092	00000000		
0x02000093	00000000		
0x02000094	00000000		
0x02000095	00000000		
0x02000096	00000000		
0x02000097	00000000		
0x02000098	00000000		
0x02000099	00000000		
0x0200009A	00000000		
0x0200009B	00000000		
0x0200009C	00000000		
0x0200009D	00000000		
0x0200009E	00000000		
0x0200009F	00000000		
0x020000A0	00000000		
0x020000A1	00000000		
0x020000A2	00000000		
0x020000A3	00000000		
0x020000A4	00000000		
0x020000A5	00000000		
0x020000A6	00000000		
0x020000A7	00000000		
0x020000A8	00000000		
0x020000A9	00000000		
0x020000AA	00000000		
0x020000AB	00000000		
0x020000AC	00000000		
0x020000AD	00000000		
0x020000AE	00000000		
0x020000AF	00000000		
0x020000B0	00000000		
0x020000B1	00000000		
0x020000B2	00000000		
0x020000B3	00000000		
0x020000B4	00000000		
0x020000B5	00000000		
0x020000B6	00000000		
0x020000B7	00000000		
0x020000B8	00000000		
0x020000B9	00000000		
0x020000BA	00000000		
0x020000BB	00000000		
0x020000BC	00000000		
0x020000BD	00000000		
0x020000BE	00000000		
0x020000BF	00000000		
0x020000C0	00000000		
0x020000C1	00000000		
0x020000C2	00000000		
0x020000C3	00000000		
0x020000C4	00000000		
0x020000C5	00000000		
0x020000C6	00000000		
0x020000C7	00000000		
0x020000C8	00000000		
0x020000C9	00000000		
0x020000CA	00000000		
0x020000CB	00000000		
0x020000CC	00000000		
0x020000CD	00000000		
0x020000CE	00000000		
0x020000CF	00000000		
0x020000D0	00000000		
0x020000D1	00000000		
0x020000D2	00000000		
0x020000D3	00000000		
0x020000D4	00000000		
0x020000D5	00000000		
0x020000D6	00000000		
0x020000D7	00000000		
0x020000D8	00000000		
0x020000D9	00000000		
0x020000DA	00000000		
0x020000DB	00000000		
0x020000DC	00000000		
0x020000DD	00000000		
0x020000DE	00000000		
0x020000DF	00000000		
0x020000E0	00000000		
0x020000E1	00000000		
0x020000E2	00000000		
0x020000E3	00000000		
0x020000E4	00000000		
0x020000E5	00000000		
0x020000E6	00000000		
0x020000E7	00000000		
0x020000E8	00000000		
0x020000E9	00000000		
0x020000EA	00000000		
0x020000EB	00000000		
0x020000EC	00000000		
0x020000ED	00000000		
0x020000EE	00000000		
0x020000EF	00000000		
0x020000F0	00000000		
0x020000F1	00000000		
0x020000F2	00000000		
0x020000F3	00000000		
0x020000F4	00000000		
0x020000F5	00000000		
0x020000F6	00000000		
0x020000F7	00000000		
0x020000F8	00000000		
0x020000F9	00000000		
0x020000FA	00000000		
0x020000FB	00000000		
0x020000FC	00000000		
0x020000FD	00000000		
0x020000FE	00000000		
0x020000FF	00000000		