

```
AREA question2, CODE, READWRITE
ENTRY
```

```
ADR          r0, STRING2
;Setting up String 2 to store the wanted bytes in and eventually store null in the end
```

```
LDR          r1, =STRING1
;put the string from memory location STRING 1 into r1 to read the string
```

```
LOOP LDRB    r2, [r1, r4]                                ;loads the
register 3 with a byte from the memory location at r1 pointing from register 4
```

```
CHECK1      CMP          r2, #0x74
;Subtract r2 from 0x74 which is "t" in Hexadecimal to see if they are equal
BNE          INSERT
;If they are not equal, then the value in r2 is not "t" and go to INSERT
```

```
            CMP          r4, #0
;Subtract r2 - 0 to see if they are equal to each other
            BEQ          CHECKH
;If they are equal check H now
            B            CHECKH
;If they are NOT equal then branch to CHECKT to see loop through the process to
clarify the first byte
```

```
CHECKT                                ;This will tell us if a space is preceding the first byte
in the string
```

```
            SUB          r11, r4, #1
;Do r4 - 1 and place it in register 11
            LDRB    r10, [r1, r11]                                ;loads
the register 10 with a byte from the memory location at r1 pointing from register 11
            CMP          r10, #0x20
;Subtract r10 from hexadecimal 20 to see if the byte in r10 is a space or not
            BNE          STORE
;If the byte in r10 is not equal to a " " then branch to STORE
            B            CHECKH
;otherwise this byte is equal to " " so branch to CHECKH
```

```
CHECKH                                ;Assuming that the first byte is valid, then continue
to "h" to check the next byte
```

```
            ADD          r4, r4, #1
;Increments r4.
```

```

        LDRB    r2, [r1, r4]                                ;loads
the register 2 with a byte from the memory location at r1 pointing from register 4

        CMP     r2, #0x68
;Compare r2 with hexadecimal 68 which is "h" by doing r2 - 0x68
        BEQ     CHECKE
;If the two are equal then branch to CHECKE to check the last byte in our pattern
        STRB    r7, [r0, r5]                                ;If they
are not equal store the byte in r0 pointed at by r5 in register 7
        ADD     r5, r5, #1
;Increase the pointer in register 5
        BNE     INSERT
;If it is not then it branches to INSERT

CHECKE   ADD     r4, r4, #1
;Increments r4.
        LDRB    r2, [r1, r4]                                ;Loads
a byte into r2 from the memory location appointed by r1 in position r4

        CMP     r2, #0x65
;Subtracts r2 by 0x65 to see if it is equal to "e"
        BEQ     INSERT                                     ;Not done
        STRB    r7, [r0, r5]
;Otherwise store the value in memory r0 pointed at by r5 in r7
        ADD     r5, r5, #1
;Increments r5
        STRB    r8, [r0, r5]                                ;Store
the byte containing the value of the character "h" in the memory
        ADD     r5, r5, #1
;Increments r5
        B       INSERT
;If it is not then it branches to INSERT

INSERT  NOP                                                ;still not done

DONE    STR     r2, [r0, r5]                                ;Stores
a the byte contained in r2, "null", into r0

```

```

        AREA    question2, DATA, READWRITE

```

```

        STRING1 DCB "and the man said they must go"        ;String1

```

```
EoS    DCB    0x00                                ;end of
string1
        align
STRING2 space    0xFF
        ;Just allocating 255 bytes
        END
```

