

Design a flowchart (or use C) and an assembly program to convert the value of a decimal number represented by a null-terminated ASCII string at label TEXT into an unsigned 32-bit binary number. This ASCII string contains only the ASCII characters "0" through "9". The string may contain leading zeros, and it will only contain valid values between 0 and 4294967295 ($2^{32}-1$). Store the result in memory at label NUMBER. For the example given below, your program must place 0x00007fff at label NUMBER. Similarly, the text "0123" must result in the value 0x0000007b stored at NUMBER. Hint: convert character "0" to number 0 by subtracting "0", ie "0" - "0" = 0.

```
.include "ubc-delmedia-macros.s"
.global _start

.data
TEXT:
.asciz "32767"
ZERO:
.ascii "0"
NUMBER:
.word 0

.text
_start:
```

Consider two separate null-terminated strings starting at labels NEEDLE and HAYSTACK. Design a flowchart (or use C) and an assembly program to find the starting memory location of the first occurrence of the NEEDLE in the HAYSTACK. Store this location in memory at label MATCH_ADDR. If NEEDLE is not present, store the value 0xFFFFFFFF instead. Do not exceed the length of the strings. For example, if HAYSTACK starts at 0x00001000 and contains "The Search is Over" and NEEDLE contains "ear", the program would write 0x00001005 to MATCH_ADDR.

```
.include "ubc-delmedia-macros.s"
.global _start

.data
HAYSTACK:
.asciz "The Search is Over"
NEEDLE:
.asciz "ear"
MATCH_ADDR:
.word 0xFFFFFFFF

.text
_start:
```

Hints

① • examine TEXT, character at a time

• convert character to digit, eg $'3' - '0' = 3$
 $\uparrow \quad \uparrow$
 ASCII: 51 48

• go to next digit

$$\Rightarrow 3 \times 10 = 30$$

$$+ \begin{array}{r} '2' - '0' \\ \hline \end{array} = +2$$

$$\Rightarrow 32 \times 10 = 320$$

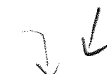
$$+ \begin{array}{r} '7' - '0' \\ \hline \end{array} = \quad 7$$

$$327$$

etc.

$$\downarrow$$

$$\text{num} = 0$$



$$\text{num} = \text{num} \times 10$$

$$\text{digit} = '3' - '0'$$

$$\text{num} = \text{num} + \text{digit}$$

② H: "The search is over" \emptyset
 \uparrow

N: "ear" \emptyset
 \uparrow

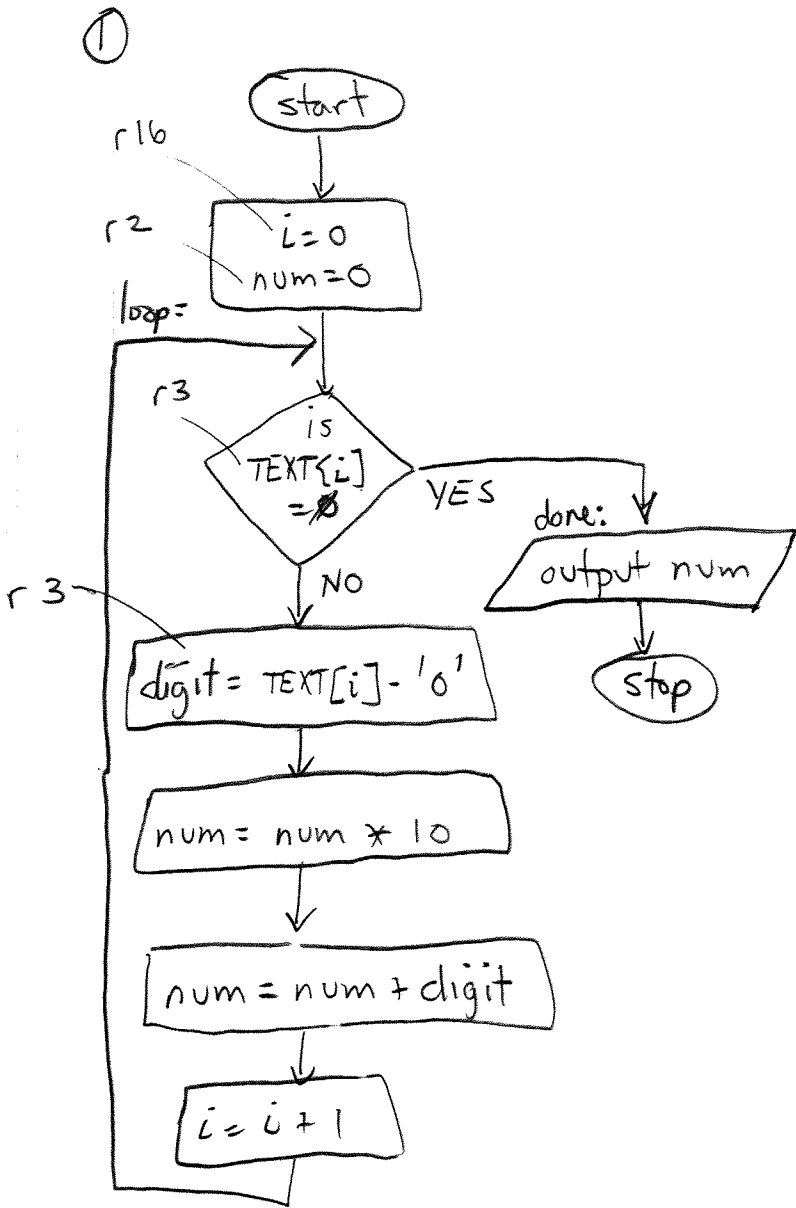
look at H from first to last character
 for each position, check if it matches all chars in N
 endloop subroutine

corner cases: "ear" ① reach char in H and \emptyset in N \Rightarrow match

"ver" ② reach \emptyset in H and \emptyset in N \Rightarrow match

"era" ③ reach \emptyset in H and char in N \Rightarrow no match

other cases: make sure you find "ear" in "ear".



```

.text
.global _start
_start: movia r16, TEXT
        movi  r2, 0
loop:   ldb  r3, 0(r16)
        beg  r3, r0, done
        subi r3, r3, '\0'
        muli r2, r2, 10
        add  r2, r2, r3
        addi r16, r16, 1
        br  loop
  
```

```

done:   movia r16, NUMBER
        stw  r2, 0(r16)
stop:   br  stop
  
```

.data

```

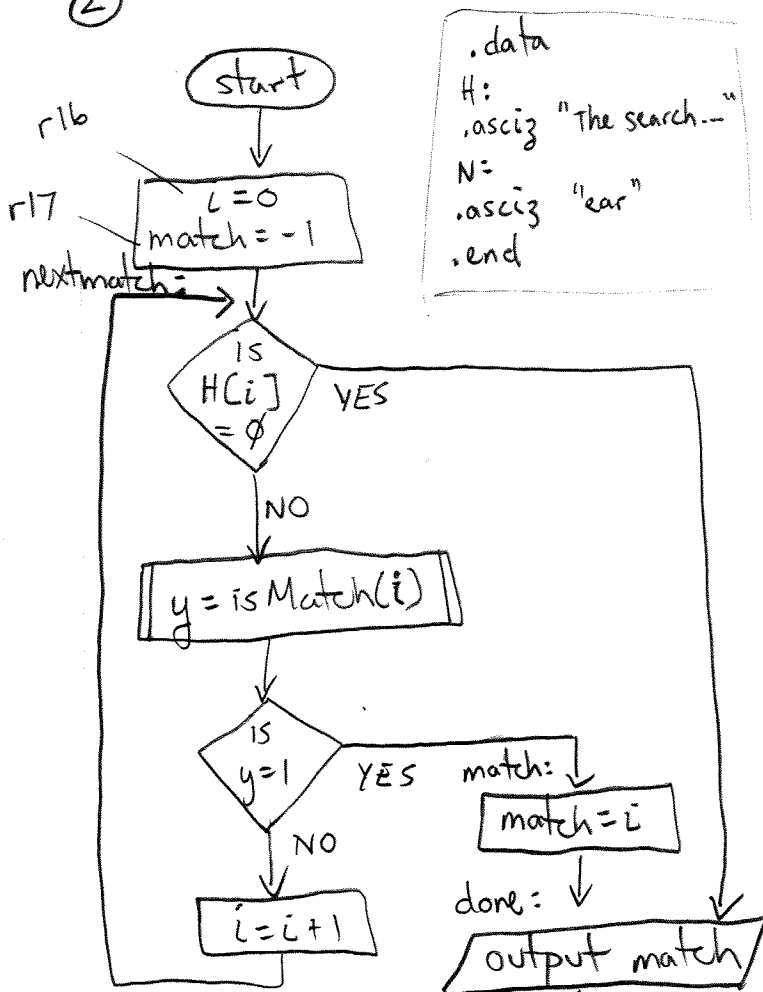
TEXT:
.asciz "32767"
  
```

```

NUMBER:
.word 0
  
```

rend

2



```

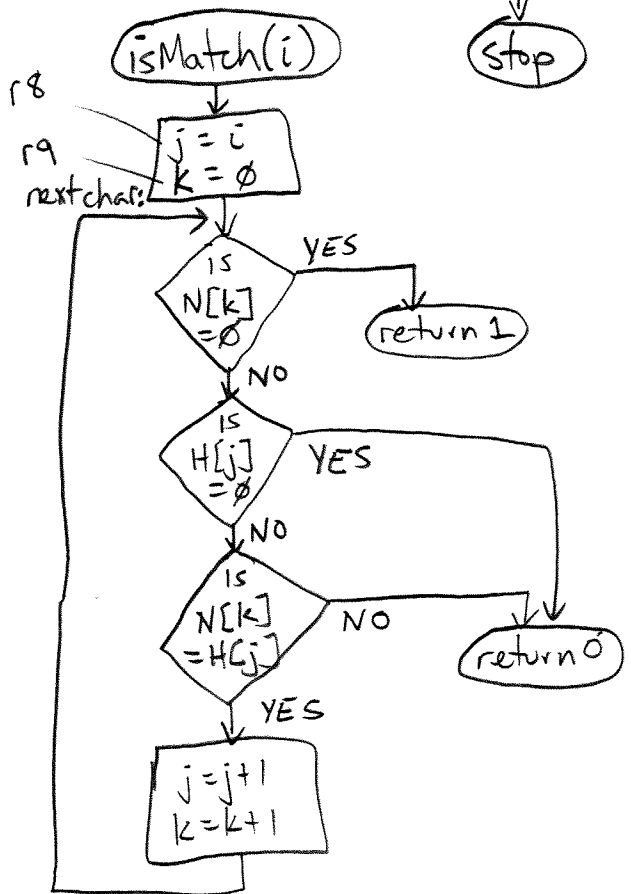
.data
H:
.asciz "The search..."
N:
.asciz "ear"
.end
    
```

```

.text
.global -start
-start: movia r16, H
       movi  r17, -1

nextmatch: ldb  r2, 0(r16)
           beq  r2, r0, done
           mov  r4, r16
           call isMatch
           bne  r2, r0, match
           addi r16, r16, 1
           br  nextmatch

match:    mov  r17, r16
done:     movia r18, MATCH_ADDR
         stw  r17, 0(r18)
stop:    br  stop
    
```



```

isMatch: mov  r8, r4
         movia r9, N

nextchar: ldb  r2, 0(r9)
         beq  r2, r0, return1
         ldb  r3, 0(r8)
         beq  r3, r0, return0
         bne  r2, r3, return0
         addi r8, r8, 1
         addi r9, r9, 1
         br  nextchar

return1: movi r2, 1
         ret

return0: movi r2, 0
         ret
    
```