MIT Tutorial 1

SAHIL BONDRE: U18CO021

1. WALP to sort the characters of your roll no string (e.g. U18CO001) in ascending order (00118COU) in-place. Use ASCII values for comparison.

```
;<Bubble Sort>

jmp start

;data
name: db 85, 49, 56, 67, 79, 48, 50, 49

;code
start: lxi H, name
mvi D, 00H
; array size
mvi C, 8
dcr C
check: mov A, M
inx H
cmp M
; A <= [M]
jc next
jz next
; swap
mov B, M
mov M, A
dcx H
mov M, B
inx H
mvi D, 01H
next: dcr C
jnz check
mov A, D
cpi 01H
jz start
hlt
```

Address	Variable	Value	Value (Decimal)
▼ 4203	name	55h	85
4203	name + 0	55h	85
4204	name + 1	31h	49
4205	name + 2	38h	56
4206	name + 3	43h	67
4207	name + 4	4Fh	79
4208	name + 5	30h	48
4209	name + 6	32h	50
420A	name + 7	31h	49

Before Sorting

Address	Variable	Value	Value (Decimal)
▼ 4203	name	30h	48
4203	name + 0	30h	48
4204	name + 1	31h	49
4205	name + 2	31h	49
4206	name + 3	32h	50
4207	name + 4	38h	56
4208	name + 5	43h	67
4209	name + 6	4Fh	79
420A	name + 7	55h	85

After Sorting

2. What modifications will you do in (1.) to sort it in descending order? (UOC81100)

To sort in descending order, change the jump conditions before swapping:

```
;<Bubble Sort Descending>
; Initial Conditions
; A <= [M]
jc next
jz next

; Newer Conditions
; A >= [M]
jnc next
```

Address	Variable	Value	Value (Decimal)
▼ 4203	name	55h	85
4203	name + 0	55h	85
4204	name + 1	4Fh	79
4205	name + 2	43h	67
4206	name + 3	38h	56
4207	name + 4	32h	50
4208	name + 5	31h	49
4209	name + 6	31h	49
420A	name + 7	30h	48

Reverse sorted

3. How will you perform (1.) without changing the original data (not in-place) (Hint: use ds directive to reserve storage for the sorted output)?

Copy the contents and sort over the copied content

```
;<Copy and Sort>

jmp start

;data
name: db 85, 49, 56, 67, 79, 48, 50, 49
answer: ds 8

;code
start: nop
;copy
lxi H, name
lxi D, answer
mvi C, 08H
up: mov A, M
stax D
inx H
inx D
dcr C
jnz up
;sort

sort: lxi H, answer
mvi D, 00H
```

```
; array size
mvi C, 8
dcr C
check: mov A, M
inx H
cmp M
; A <= [M]
jc next
jz next
; swap
mov B, M
mov M, A
dcx H
mov M, B
inx H
mvi D, 01H
next: dcr C
jnz check
mov A, D
cpi 01H
jz sort
hlt
```

Address	Variable	Value	Value (Decimal)
4203	name	55h	85
4203	name + 0	55h	85
4204	name + 1	31h	49
4205	name + 2	38h	56
4206	name + 3	43h	67
4207	name + 4	4Fh	79
4208	name + 5	30h	48
4209	name + 6	32h	50
420A	name + 7	31h	49

420B	answer	30h	48
420B	answer + 0	30h	48
420C	answer + 1	31h	49
420D	answer + 2	31h	49
420E	answer + 3	32h	50
420F	answer + 4	38h	56
4210	answer + 5	43h	67
4211	answer + 6	4Fh	79
4212	answer + 7	55h	85

4. What would be the result in (1.) if the roll no string input was first converted to lowercase?

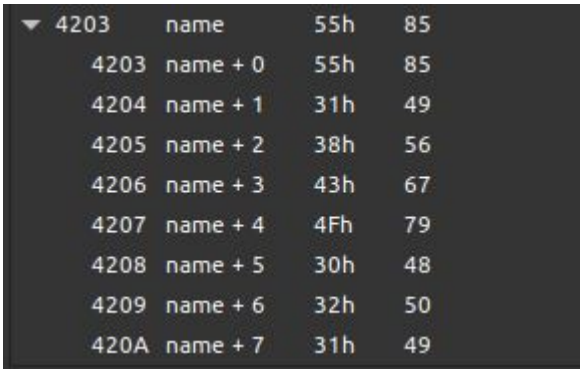
The result would still be the same ie. numbers in ascending order then letters in ascending order.

So: u18co021 => 01128cou

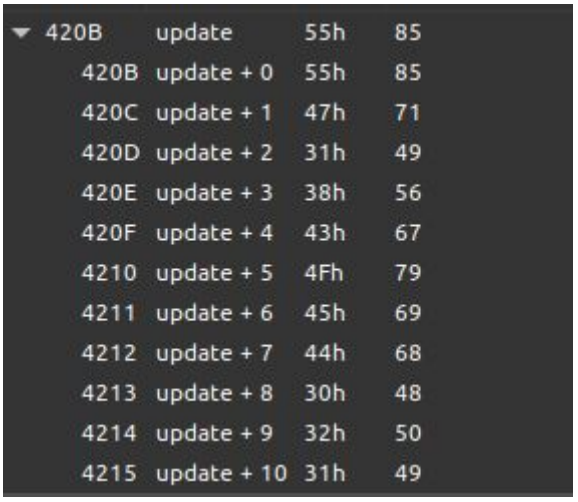
5. The management decided to change the prefix from U to UG and department code from CO to COED. WALP to convert your old roll no. (U18CO001) to the new format (UG18COED001) (Hint: use ds directive to reserve extra bytes, to avoid overwriting code segment)

```
;<Roll Number Change>
jmp start

;data
name: db 85, 49, 56, 67, 79, 48, 50, 49
update: ds 11


;code
start: nop
lxi H, name
lxi D, update
; u
mov A, M
stax D
inx H
inx D
; g
mvi A, 47H
stax D
inx D
; next 4: 18CO
mvi c, 04H
loop: nop
mov A, M
stax D
inx H
inx D
dcr C
jnz loop
; e
mvi A, 45H
stax D
inx D
; d
mvi A, 44H
stax D
inx D
; last 3: 021
```

```
mvi C, 03H
loop2: nop
mov A, M
stax D
inx H
inx D
dcr C
jnz loop2
hlt
```



4203	name	55h	85
4203	name + 0	55h	85
4204	name + 1	31h	49
4205	name + 2	38h	56
4206	name + 3	43h	67
4207	name + 4	4Fh	79
4208	name + 5	30h	48
4209	name + 6	32h	50
420A	name + 7	31h	49

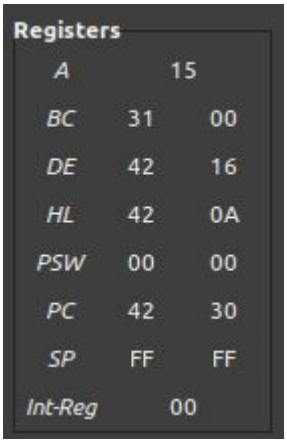


420B	update	55h	85
420B	update + 0	55h	85
420C	update + 1	47h	71
420D	update + 2	31h	49
420E	update + 3	38h	56
420F	update + 4	43h	67
4210	update + 5	4Fh	79
4211	update + 6	45h	69
4212	update + 7	44h	68
4213	update + 8	30h	48
4214	update + 9	32h	50
4215	update + 10	31h	49

6. In order to simplify a few tasks, sometimes, it is sufficient to work with just your serial no. instead of the entire roll no. string. WALP that does this conversion and stores the output in the accumulator. (U18CO010 -> A, U18CO016 -> 10, etc.)

```
;<Fetch Last 3 Numbers>

jmp start

;data
;U18CO021
name: db 85, 49, 56, 67, 79, 48, 50, 49

;code
start: nop
lxi H, name
inx H
inx H
inx H
inx H
inx H
; hundreds place
```

```
mvi C, 100
mvi A, 00H
loop100: nop
add M
; ascii
sui 48
dcr C
jnz loop100

; tens place
inx H
mvi C, 10
loop10: nop
add M
; ascii
sui 48
dcr C
jnz loop10

; units place
inx H
add M
; ascii
sui 48
hlt
```



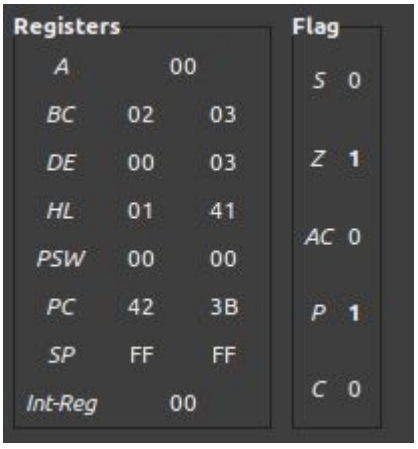
0x15 = 21 copied to A (u18co021)

7. Your program (6.) will fail for which roll nos? Modify it, so that it doesn't fail for the cases you've mentioned. Use a register pair for the output instead of the accumulator.

The above program will fail for roll numbers greater than 256. To overcome this we'll use 2 registers for storing the total value ie HL

```
;<Fetch Last 3 Numbers Advanced>

jmp start

;data
;U18CO321
name: db 85, 49, 56, 67, 79, 51, 50, 49

;code
start: nop
lxi H, name
inx H
inx H
inx H
inx H
inx H
mov A, M
sui 48
; hundreds place in C
mov C, A
inx H
mov A, M
sui 48
; tens place in B
mov B, A
inx H
; units place in A
mov A, M
sui 48
lxi H, 0000H
; 1
mvi D, 00H
mov E, A
dad D
; 10
mvi A, 10
mov E, B
loop10: nop
dad D
dcr A
jnz loop10

; 100
mvi A, 100
mov E, C
```

```
loop100: nop
dad D
dcr A
jnz loop100
hlt
```



0x0141 = 321 copied to HL (u18co321)