K.Yashwanth 2021A7PS0136U

Birla Institute of Technology and Science Pilani, Dubai Campus

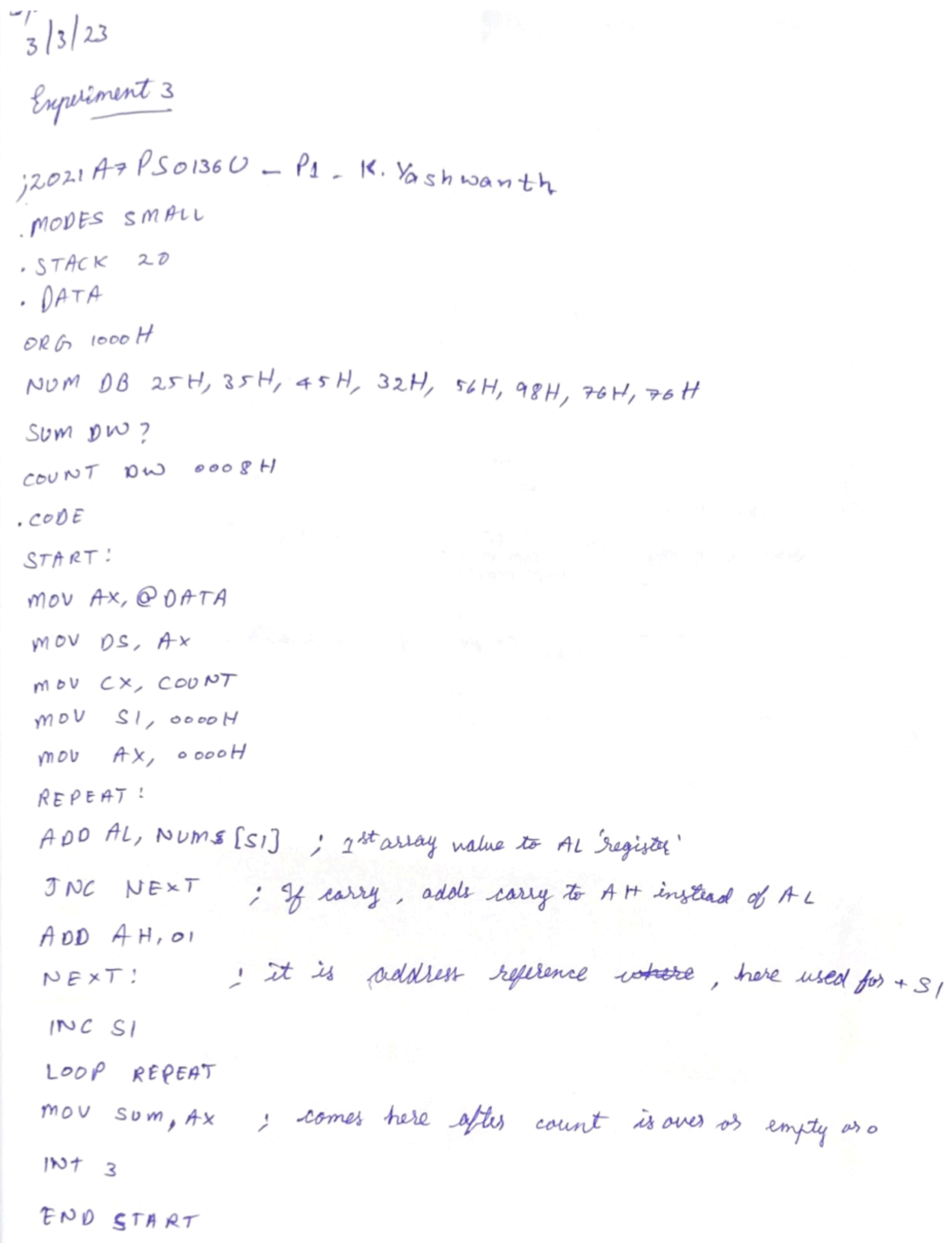
Dubai International Academic City

CS/ECE/INSTR/EEE F241
MICROPROCESSORS AND INTERFACING
LABORATORY MANUAL
II Semester 2021-22

EXPERIMENT-3
Working with Number Arrays

P1. Write a program to add an array of eight 2-digit hexadecimal numbers stored in memory and store the result in memory.

Handwritten Program:



3/3/23

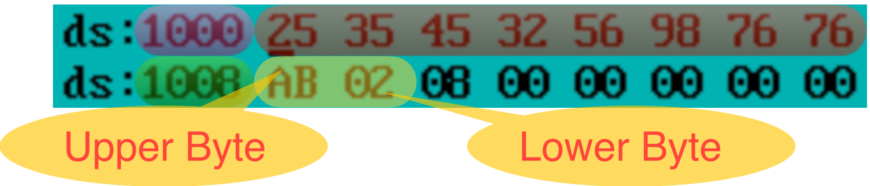
Experiment 3

```
;2021A7PS0136U - P1 - K. Yashwanth
.MODES SMALL
.STACK 20
.DATA
ORG 1000H
NUM DB 25H, 35H, 45H, 32H, 56H, 98H, 76H, 76H
SUM DW ?
COUNT DW 0008H
.CODE
START:
MOV AX,@DATA
MOV DS, AX
MOV CX, COUNT
MOV SI, 0000H
MOV AX, 0000H
REPEAT:
ADD AL, NUMS[SI]   ; 1st array value to AL 'register'
JNC NEXT           ; If carry, adds carry to AH instead of AL
ADD AH,01
NEXT:              ; it is address reference ~~where~~, here used for +SI
INC SI
LOOP REPEAT
MOV SUM,AX         ; comes here after count is over or empty or 0
INT 3
END START
```

K.Yashwanth 2021A7PS0136U

Solution ScreenShot:

🟪 →	*Array starting index*	🟥 →	*Given array of numbers*
🟩 →	*Result starting index*	🟨 →	*Sum of array of numbers*



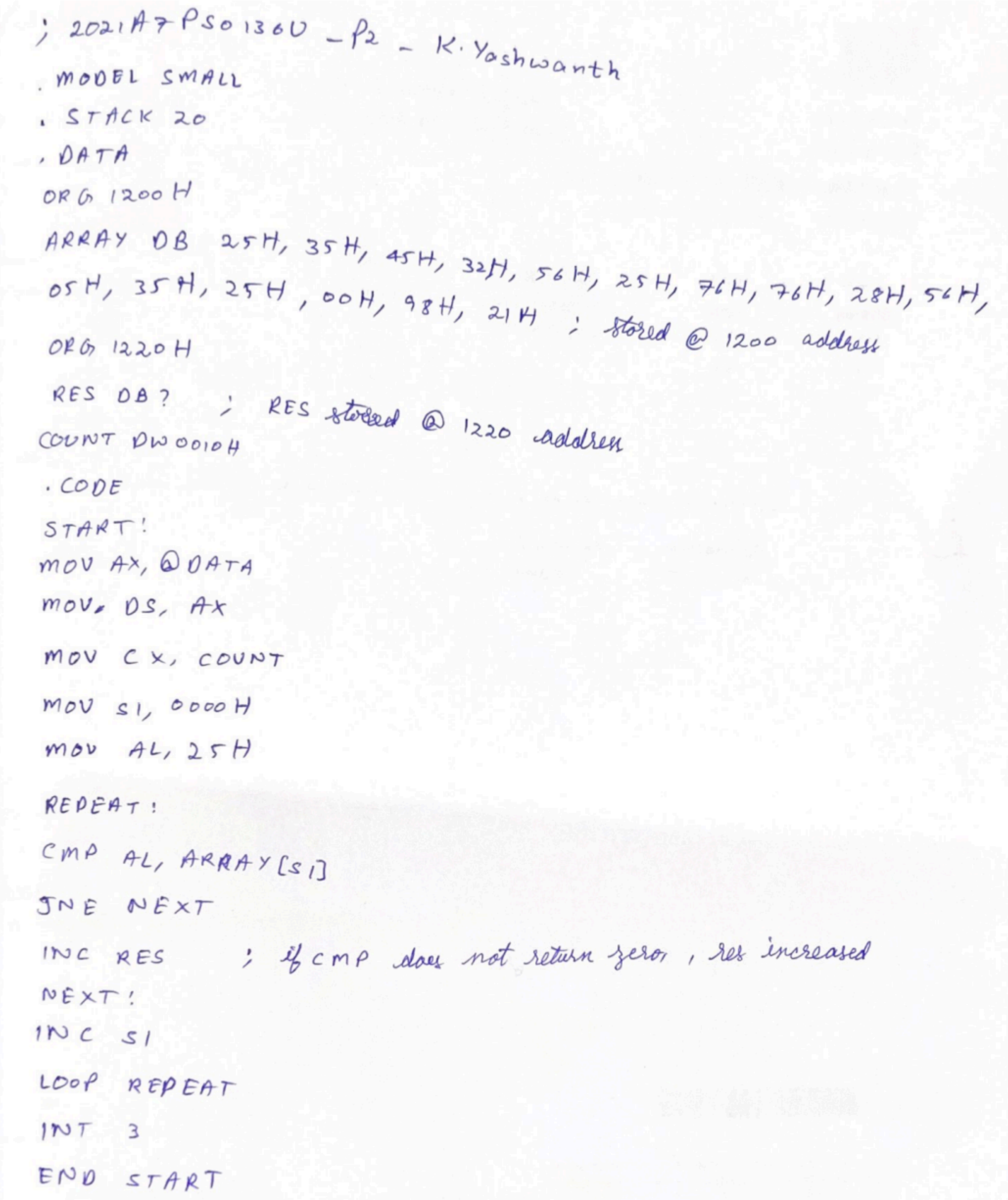
```
ds:1000 25 35 45 32 56 98 76 76
ds:1008 AB 02 08 00 00 00 00 00
```

Upper Byte

Lower Byte

P2. Write a program to count number of occurrences of the byte 25H in the given array of 16-bytes stored starting from 1200H. Also store the result in 1220H memory location.

Handwritten Program:



```
; 2021A7PS0136U - P2 - K.Yashwanth
.MODEL SMALL
.STACK 20
.DATA
ORG 1200H
ARRAY DB 25H, 35H, 45H, 32H, 56H, 25H, 76H, 76H, 28H, 56H,
05H, 35H, 25H , 00H, 98H, 21H ; stored @ 1200 address
ORG 1220H
RES DB ?   ; RES stored @ 1220 address
COUNT DW 0010H
.CODE
START:
MOV AX, @DATA
MOV, DS, AX
MOV CX, COUNT
MOV SI, 0000H
MOV AL, 25H
REPEAT:
CMP AL, ARRAY[SI]
JNE NEXT
INC RES     ; if cmp does not return zero, res increased
NEXT:
INC SI
LOOP REPEAT
INT 3
END START
```

K.Yashwanth 2021A7PS0136U

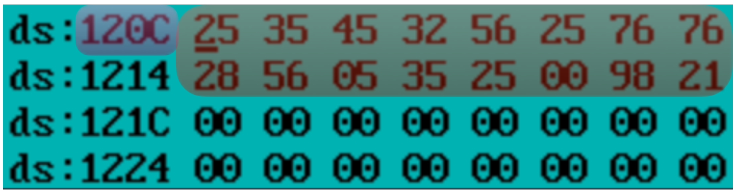
Solution ScreenShot:

🟪 → *Array starting index*

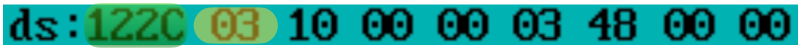
🟥 → *Given array of numbers*

🟩 → *Result address index*

🟨 → *Number of occurrences*



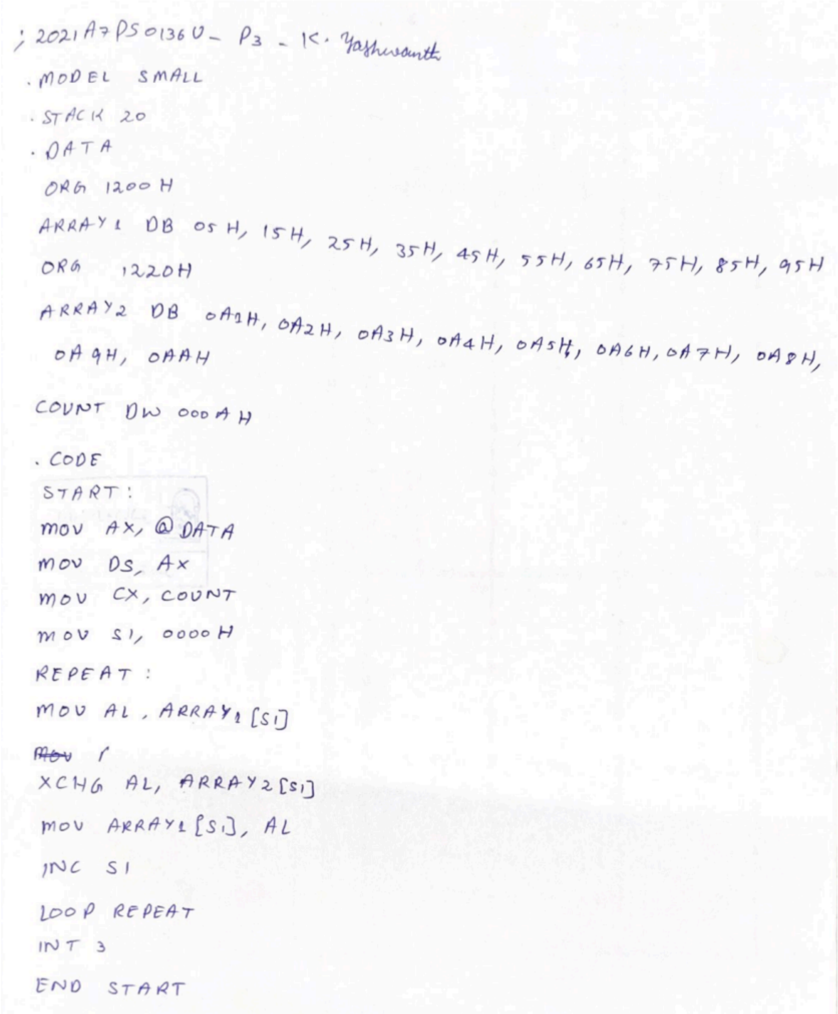
```
ds:120C 25 35 45 32 56 25 76 76
ds:1214 28 56 05 35 25 00 98 21
ds:121C 00 00 00 00 00 00 00 00
ds:1224 00 00 00 00 00 00 00 00
```



```
ds:122C 03 10 00 00 03 48 00 00
```

P3. Write a program to exchange two data blocks of length 10-bytes stored in memory starting from 1200H and 1220H respectively.

Handwritten Program:



```
; 2021A7PS0136U - P3 - K. Yashwanth
.MODEL SMALL
.STACK 20
.DATA
ORG 1200H
ARRAY1 DB 05H, 15H, 25H, 35H, 45H, 55H, 65H, 75H, 85H, 95H
ORG 1220H
ARRAY2 DB 0A1H, 0A2H, 0A3H, 0A4H, 0A5H, 0A6H, 0A7H, 0A8H,
 0A9H, 0AAH

COUNT DW 000AH

.CODE
START:
mov AX, @DATA
mov DS, AX
mov CX, COUNT
mov SI, 0000H
REPEAT:
MOV AL, ARRAY1[SI]
~~Mov~~ /
XCHG AL, ARRAY2[SI]
mov ARRAY1[SI], AL
INC SI
LOOP REPEAT
INT 3
END START
```

K.Yashwanth 2021A7PS0136U

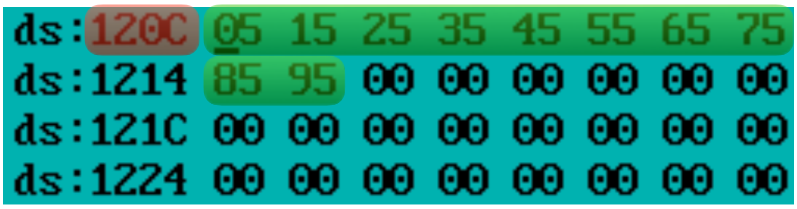
Solution ScreenShot:

🟥 → *Array 1 address index*

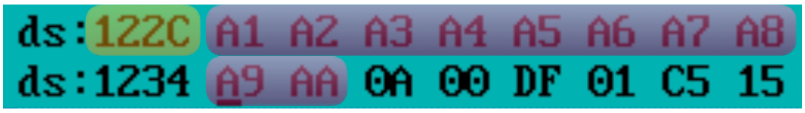
🟨 → *Array 2 address index*

🟩 → *Array 1 contents*

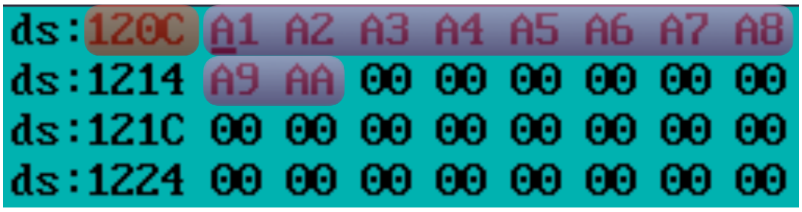
🟪 → *Array 2 contents*



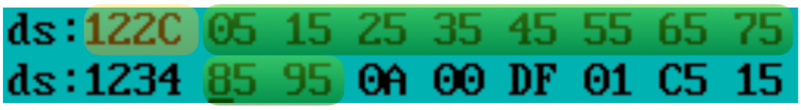
```
ds:120C 05 15 25 35 45 55 65 75
ds:1214 85 95 00 00 00 00 00 00
ds:121C 00 00 00 00 00 00 00 00
ds:1224 00 00 00 00 00 00 00 00
```



```
ds:122C A1 A2 A3 A4 A5 A6 A7 A8
ds:1234 A9 AA 0A 00 DF 01 C5 15
```



```
ds:120C A1 A2 A3 A4 A5 A6 A7 A8
ds:1214 A9 AA 00 00 00 00 00 00
ds:121C 00 00 00 00 00 00 00 00
ds:1224 00 00 00 00 00 00 00 00
```



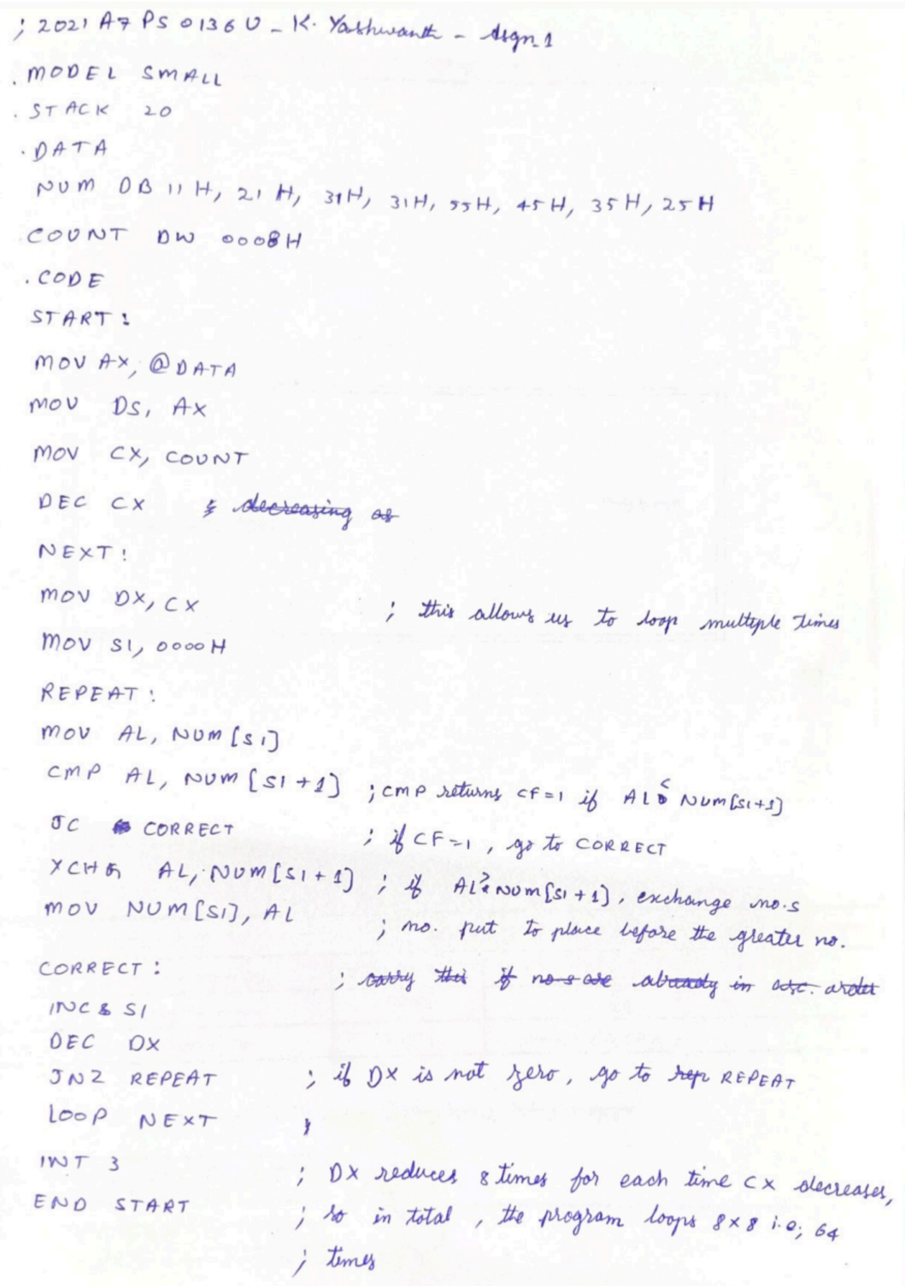
```
ds:122C 05 15 25 35 45 55 65 75
ds:1234 85 95 0A 00 DF 01 C5 15
```

K.Yashwanth 2021A7PS0136U

ASSIGNMENT/EXERCISE QUESTION:

Q1. Write a program to arrange the given array of 8-bit binary numbers stored in the memory in ascending order.
NUM DB 95H, 85H, 75H, 65H, 55H, 45H, 35H, 25H

Handwritten Program:



```
; 2021A7PS0136U - K. Yashwanth - Asgn 1
.MODEL SMALL
.STACK 20
.DATA
 NUM DB 11H, 21H, 31H, 31H, 55H, 45H, 35H, 25H
COUNT DW 0008H
.CODE
START:
MOV AX, @DATA
MOV DS, AX
MOV CX, COUNT
DEC CX        ; ~~decreasing as~~
NEXT:
MOV DX, CX                  ; this allows us to loop multiple times
MOV SI, 0000H
REPEAT:
MOV AL, NUM[SI]
CMP AL, NUM[SI+1]  ; CMP returns CF=1 if AL < NUM[SI+1]
JC  CORRECT                 ; if CF=1, go to CORRECT
XCHG AL, NUM[SI+1]  ; if AL > NUM[SI+1], exchange no.s
MOV NUM[SI], AL             ; no. put to place before the greater no.
CORRECT:                   ; ~~carry this if no.s are already in asc. order~~
INC SI
DEC DX
JNZ REPEAT            ; if DX is not zero, go to ~~rep~~ REPEAT
LOOP NEXT             ;
INT 3
                      ; DX reduces 8 times for each time CX decreases,
END START             ; so in total, the program loops 8x8 i.e; 64
                      ; times
```

K.Yashwanth 2021A7PS0136U

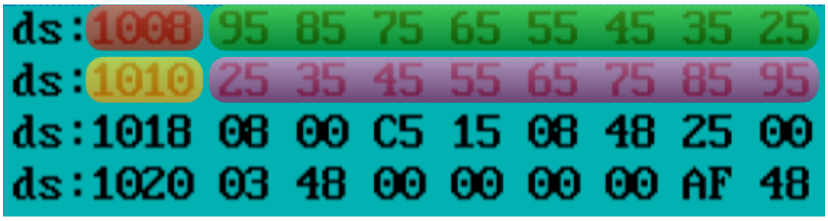
Solution ScreenShot:

🟥 → *Array 1 address index*

🟨 → *Array 2 address index*

🟩 → *Before arranging*

🟪 → *After arranging*



```
ds:1008 95 85 75 65 55 45 35 25
ds:1010 25 35 45 55 65 75 85 95
ds:1018 08 00 C5 15 08 48 25 00
ds:1020 03 48 00 00 00 00 AF 48
```