BLG 212E
Microprocessor Systems
Recitation 3

Altay Ünal, Kadir Özlem, Sadettin Fidan

November 11, 2024

Question 1

For a CPU with 8-bit data bus and 20-bit address bus, design a 24Kx8 memory using the following memory units.

Two 2Kx8

One 4Kx8

One 16Kx8

(a) Draw the memory map. Make sure to indicate the starting and ending addresses for each unit. Start with the address **$0A000** and **leave no spaces** between blocks.

(b) Write the simplified address decoder output functions (CEx) for each memory component in terms of the decoder inputs. (Note that the chip selection inputs are active zero (0). For example, memory unit M1 is selected when CE1 is 0 and all other CEx are 1.)

Solution 1a

0000 1010 0000 0000 0000: 0A000

⋮

0000 1010 0111 1111 1111: 0A7FF (M1-2K)

0000 1010 1000 0000 0000: 0A800

⋮

0000 1010 1111 1111 1111: 0AFFF (M2-2K)

0000 1011 0000 0000 0000: 0B000

⋮

0000 1011 1111 1111 1111: 0BFFF (M3-4K)

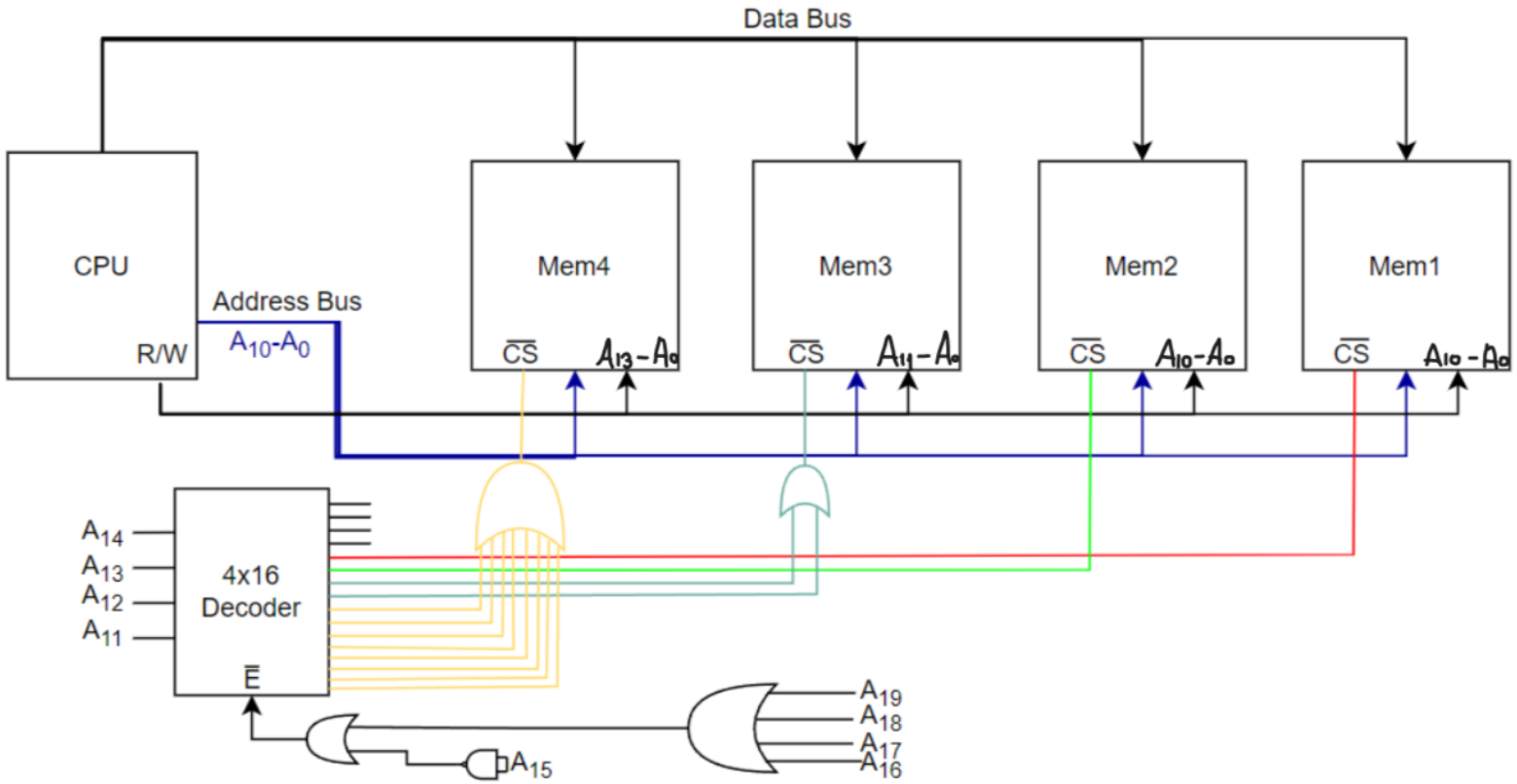
0000 1100 0000 0000 0000: 0C000

⋮

0000 1111 1111 1111 1111: 0FFFF (M4-16K)

Solution 1b

$$\overline{CE1} = A_{11} + A_{12} + \overline{A_{13}} + A_{14} + \overline{A_{15}} + A_{16} + A_{17} + A_{18} + A_{19}$$
$$\overline{CE2} = \overline{A_{11}} + A_{12} + \overline{A_{13}} + A_{14} + \overline{A_{15}} + A_{16} + A_{17} + A_{18} + A_{19}$$
$$\overline{CE3} = \overline{A_{12}} + \overline{A_{13}} + A_{14} + \overline{A_{15}} + A_{16} + A_{17} + A_{18} + A_{19}$$
$$\overline{CE4} = \overline{A_{14}} + \overline{A_{15}} + A_{16} + A_{17} + A_{18} + A_{19}$$



Question 2

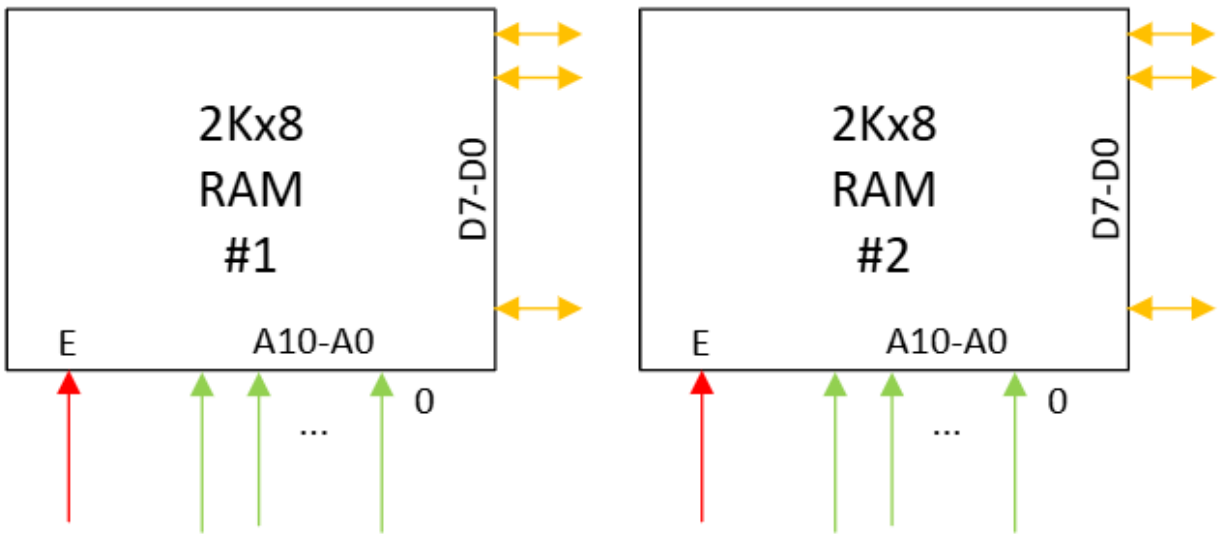
Connecting 2 Memory Units into a single Memory Unit

One 2Kx8,

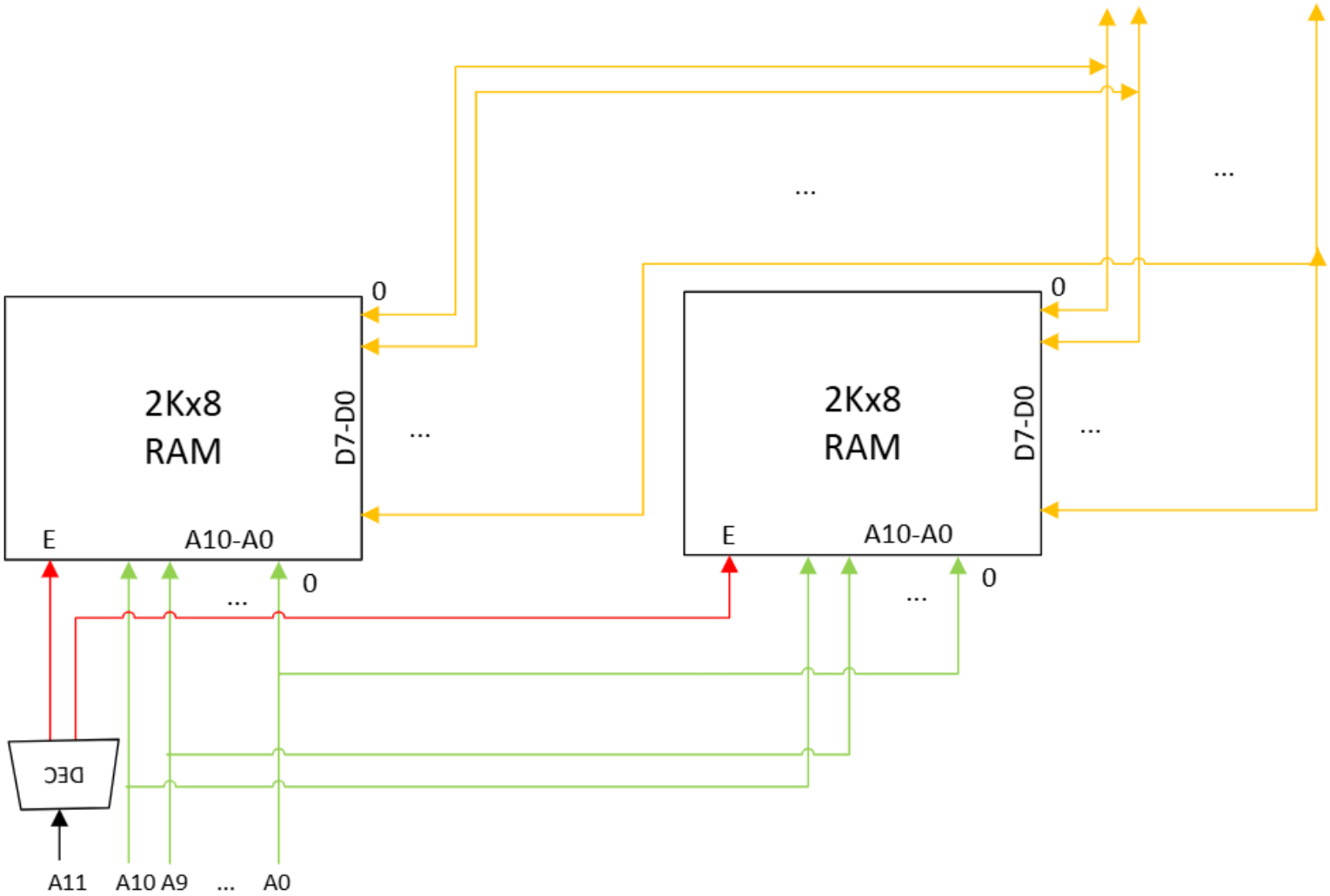
One more 2Kx8.

(a) Create a single 4Kx8

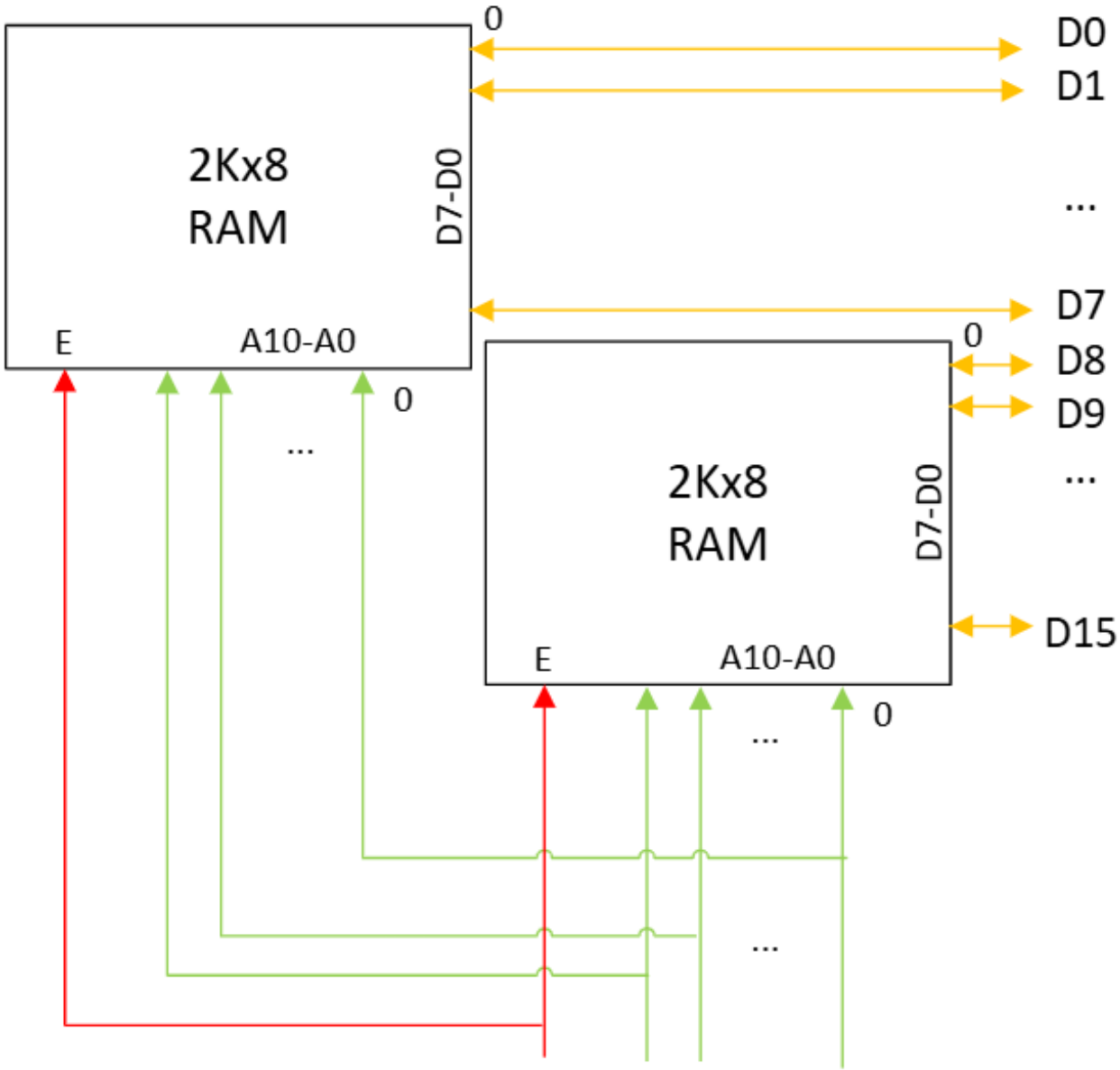
(b) Create a single 2Kx16



Solution 2a



Solution 2b


0
D0
D1
2Kx8
RAM
D7-D0
...
D7
E
A10-A0
0
0
D8
D9
...
...
2Kx8
RAM
D7-D0
D15
E
A10-A0
0
...
...

Question 3

Initial values for the following memory locations are given as (0x20000400)=0x99, (0x20000401)=0x11, (0x20000402)=0x22, (0x20000403)=0x88 and (0x20000404)=0xBA. An Assembly code is given below. What would be the final values stored in the used registers?

```
ADDRESS_A EQU 0x20000400 ; define ADDRESS_A 0x20000400
ADDRESS_B EQU 0x20000500 ; define ADDRESS_B 0x20000500
      LDR R0, =ADDRESS_A ; ptr1 = 0x20000400
      LDR R1, =ADDRESS_B ; ptr2 = 0x20000500
      LDRH R2, [R0]      ; val1 = *((short *)ptr1) // 0x1199
      LDRB R3, [R0, #2]  ; val2 = *((char *)ptr2 + 2) // 0x22
      ORRS R4, R2        ; val3 |= val1
      STR R4, [R1, #4]   ; *(ptr2 + 4) = val3 // ptr2[4] = 0x1199
      LDRH R6, [R0, #4]  ; val5 = *((short *)ptr1 + 4) // 0x00BA
      ANDS R3, R6        ; val2 &= val5 // 0x0022
      ADDS R5, R6, R3    ; val4 = val5 + val2 // 0x00DC
      ADDS R0, R0, #8    ; ptr1 += 8 // 0x20000408
      STR R5, [R0, #4]   ; *(ptr1 + 4) = val4 // ptr1[4] = 0x0x00D
```

Solution 3

R0	0x20000408
R1	0x20000500
R2	0x1199
R3	0x22
R4	0x1199
R5	0xDC
R6	0xBA

Question 4

Assuming the LSB (rightmost) is numbered as b_0, and the MSB (leftmost) as b_{31}, let R1 be $b_{31}b_{30}\cdots b_5b_4b_3b_2b_1b_0$. Write an Assembly program that will replace R1 with a value where the most significant 16 bits are repeated. Explain your program.
(*Hint:* Mirrored version for 8 bits: $b_7b_6b_5b_4b_3b_2b_1b_0$ becomes $b_7b_6b_5b_4b_7b_6b_5b_4$)

Solution 4

```
LDR R0, =0xFFFF0000; mask is set
MOVS R2, R1; R1 is saved to R2
LSRS R2, #16; Shifted to last 16 bits
ANDS R1, R0; Cleared last 16 bits
ORRS R1, R2; Updated last 16 bits
```

Question 5

Write ABI compliant function using Thumb assembly that performs the same operation as this C function.

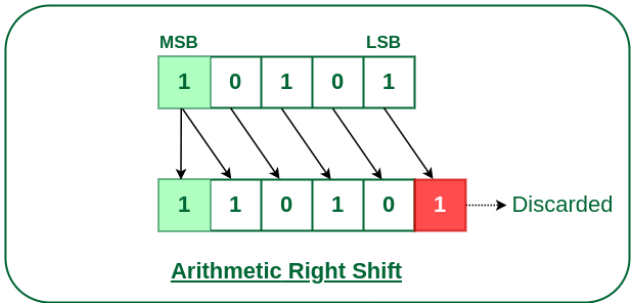
```
int f(int x)
{
    return ((x >> 4) * 7);
}
```

Solution 5

Argument/Scratch registers are not expected to be preserved upon returning from a called subroutine. (R0-R3)

Preserved ("variable") registers are expected to have their original values upon returning from a called subroutine. (R4-R8, R10-R11)

```
ASRS R0, R0, #4
MOVS R1, #7
MULS R0, R1, R0
BX LR
```



Question 6

Write a program in C that does the same thing as the following Arm Thumb assembly language code. Your C code must not exceed three lines.

```
ldr r0, =0x2000A500 ; int *r0 = 0x2000A500
ldr r1, [r0]        ; r1 = *(0x2000A500)
subs r1, r1, #5     ; r1 -= 5
str r1, [r0]        ; *r0 = r1
```

Solution 6

```
int *number = 0x2000A500;
*number = *number - 5;
```

Question 7 (Quiz2)

Examine the following ARM assembly code snippet, focusing on the sequence of function calls and the stack state during execution. Using the labeled instructions (I1–I19), complete the following tasks (Note: The main function will be executed once):

1. Instruction Execution Order: List the precise sequence in which each instruction (I1 to I19) executes.
2. Stack State Analysis: Draw the state of the stack after each PUSH and POP operation. For each step, show the stack's contents and indicate any changes made by each PUSH or POP instruction.
3. Register Values: At the end, what are the values of R0, R1, R2 registers?

Question 7 (Quiz2)

```
__main  FUNCTION
        EXPORT  __main
        BL        foo       ; I1
        B       __main      ; I2
foo
        PUSH {LR}           ; I3
        MOVS R0, #5         ; I4
        MOVS R1, #10        ; I5
        MOVS R2, #15        ; I6
        BL bar              ; I7
        POP {PC}            ; I8
tar
        PUSH {LR}           ; I9
        ADD R0, R0, R3      ; I10
        ADD R1, R1, R4      ; I11
        POP {PC}            ; I12
bar
        PUSH {LR}           ; I13
        PUSH {R0, R1, R2}   ; I14
        MOVS R3, #25        ; I15
        MOVS R4, #30        ; I16
        BL tar              ; I17
        POP {R0, R1, R2}    ; I18
        POP {PC}            ; I19
```

Question 7 (Solution 1)

1. I1	2. I3	3. I4	4. I5	5. I6	6. I7	7. I13	8. I14	9. I15	10. I16
11. I17	12. I9	13. I10	14. I11	15. I12	16. I18	17. I19	18. I8	19. I2	20.
21.	22.	23.	24.	25.	26	27.	28.	29.	30.

Question 7 (Solution 2)

EXECUTED INS. : I3

STACK CONTENT
&I2 (Address of instruction I2)

EXECUTED INS. : I13

STACK CONTENT
&I2 (Address of instruction I2)
&I8 (Address of instruction I8)

EXECUTED INS. : I14

STACK CONTENT
&I2 (Address of instruction I2)
&I8 (Address of instruction I8)
15 (Value of R2)
10 (Value of R1)
5 (Value of R0)

EXECUTED INS. : I9

STACK CONTENT
&I2 (Address of instruction I2)
&I8 (Address of instruction I8)
15 (Value of R2)
10 (Value of R1)
5 (Value of R0)
&I18 (Address of instruction I18)

EXECUTED INS. : I12

STACK CONTENT
&I2 (Address of instruction I2)
&I8 (Address of instruction I8)
15 (Value of R2)
10 (Value of R1)
5 (Value of R0)

EXECUTED INS. : I18

STACK CONTENT
&I2 (Address of instruction I2)
&I8 (Address of instruction I8)

EXECUTED INS. : I19

STACK CONTENT
&I2 (Address of instruction I2)

EXECUTED INS. : I8

STACK CONTENT

Question 7 (Solution 3)

R0	R1	R2
5	10	15