

COMP LAB 1

ADIT LUHADIA

190911112

IT A

Q1 A) Transfer 5 words of data in data segment, where blocks are a). Overlapping and b).non-overlapping.

```
DATA
SEGMENT

    SRC DW 11H, 22H, 33H, 44H, 55H
    DES DW 3 DUP (0)
    DES_START DW 3
    COUNT DB 5
DATA ENDS

CODE SEGMENT
    ASSUME CS: CODE, DS: DATA
START:
    MOV AX, DATA
    MOV DS, AX

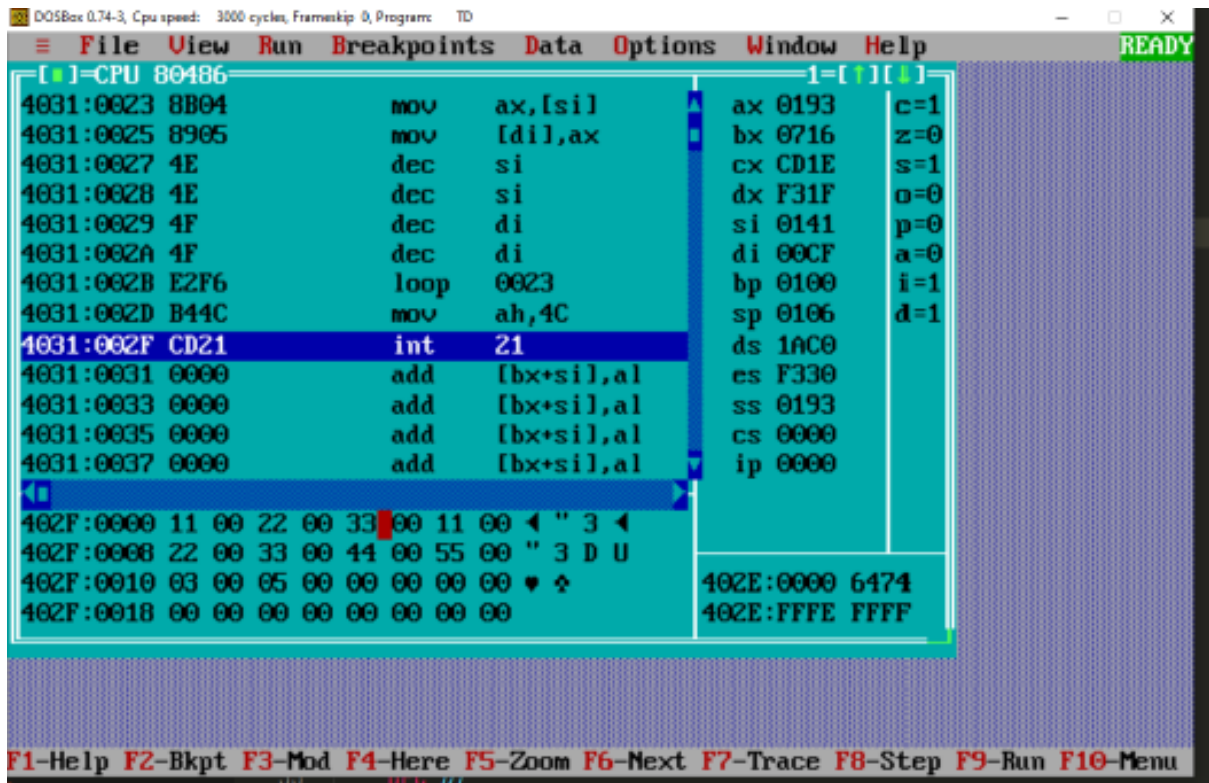
    MOV CH, 00H
    MOV CL, COUNT

    LEA SI, SRC
    ADD SI, CX
    ADD SI, CX
    DEC SI
    DEC SI
    LEA DI, DES
    ADD DI, DES_START
    ADD DI, DES_START
    DEC DI
    DEC DI

COPY:
    MOV AX, [SI]
    MOV [DI], AX
    DEC SI
    DEC SI
    DEC DI
    DEC DI
    LOOP COPY
```

```
MOV AH, 4CH
INT 21H
```

```
CODE ENDS
END START
```



Q1 B)

DATA

SEGMENT

```
SRC DW 15H, 25H, 35H, 45H, 55H
```

```
DES DW 5 DUP(0)
```

```
COUNT DB 5
```

DATA ENDS

CODE SEGMENT

```
ASSUME CS: CODE, DS: DATA
```

START:

```
MOV AX, DATA
```

```
MOV DS, AX
```

```
MOV CH, 00H
```

```
MOV CL, COUNT
```

```

LEA SI, SRC
LEA DI, DES

```

COPY:

```

MOV AX, [SI]
MOV [DI], AX
INC SI
INC SI
INC DI
INC DI
LOOP COPY

```

```

MOV AH, 4CH
INT 21H

```

CODE ENDS

END START

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...

File View Run Breakpoints Data Options Window Help		READY
[CPU 80486]		1
402F:000B 8D360000	lea si,[0000]	ax 0192 c=1
402F:000F 8D3E0A00	lea di,[000A]	bx 0716 z=0
402F:0013 8B04	mov ax,[si]	cx CD1C s=1
402F:0015 8905	mov [di],ax	dx F31D o=0
402F:0017 46	inc si	si 0141 p=0
402F:0018 46	inc si	di 00CF a=0
402F:0019 47	inc di	bp 0100 i=1
402F:001A 46	inc si	sp 0106 d=1
402F:001B E2F6	loop 0013	ds 1ABF
402F:001D B44C	mov ah,4C	es F32E
402F:001F CD21	int 21	ss 0192
402F:0021 0000	add [bx+si],al	cs 0000
402F:0023 0000	add [bx+si],al	ip 0000
402F:0025 0000	add [bx+si],al	
402F:0027 0000	add [bx+si],al	
402D:0000 15 00 25 00 35 00 45 00 5 5 E		402C:0000 6474
402D:0008 55 00 15 00 45 00 45 00 U 5 E E		402C:FFFE FFFF
402D:0010 00 00 00 00 05 00 00 00 *		402C:FFFC 7206
402D:0018 00 00 00 00 00 00 00 00		402C:FFFA 402F
402D:0020 B8 2D 40 8E D8 B5 00 8A 1-0A+ è		402C:FFF8 0021

1-Help F2-Bkpt F3-Mod F4-Here F5-Zoom F6-Next F7-Trace F8-Step F9-Run F10-Menu

Q2) Add two 32 - bit hexadecimal numbers in data segment and store the result in data segment.

DATA

SEGMENT

```
X DD 22099720H
```

```
Y DD 88745474H
```

```

RES DD ?
CARRY DB ?
DATA ENDS

CODE SEGMENT
    ASSUME CS: CODE, DS: DATA
START:
    MOV AX, DATA
    MOV DS, AX

    MOV AX, WORD PTR X
    MOV BX, WORD PTR Y
    ADD AX, BX
    MOV WORD PTR RES, AX
    MOV AX, WORD PTR X+2
    MOV BX, WORD PTR Y+2
    ADC AX, BX
    MOV WORD PTR RES+2, AX
    ADC CARRY, 0

    MOV AH, 4CH
    INT 21H
CODE ENDS
END START

```

The screenshot shows the DOSBox 0.74-3 interface. The title bar reads "DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...". The menu bar includes "File", "View", "Run", "Breakpoints", "Data", "Options", "Window", and "Help". The status bar at the bottom shows function key shortcuts: "F1-Help", "F2-Bkpt", "F3-Mod", "F4-Here", "F5-Zoom", "F6-Next", "F7-Trace", "F8-Step", "F9-Run", and "F10-Menu".

The main window is divided into three sections:

- Assembly Code:** A list of instructions with their addresses. The instruction at address `cs:0010 0800` is highlighted in blue. The instructions are:
 - `cs:0000 54 push sp`
 - `cs:0001 027615 add dh,[bp+15]`
 - `cs:0004 CE into`
 - `cs:0005 017615 add [bp+15],si`
 - `cs:0008 3802 cmp [bp+si],al`
 - `cs:000A 7615 jbe 0021`
 - `cs:000C D7 xlat`
 - `cs:000D 017615 add [bp+15],si`
 - `cs:0010 0800 or [bx+si],al`**
 - `cs:0012 7000 jo 0014`
 - `cs:0014 60 pusha`
 - `cs:0015 1000 adc [bx+si],al`
 - `cs:0017 F060 lock pusha`
 - `cs:0019 1000 adc [bx+si],al`
 - `cs:001B F060 lock pusha`
- Registers:** A table of register values on the right side:
 - ax 0192 c=1
 - bx 0716 z=0
 - cx CD1C s=1
 - dx F31D o=0
 - si 0141 p=0
 - di 00CF a=0
 - bp 0100 i=1
 - sp 0106 d=1
 - ds 1ABF
 - es F32E
 - ss 0192
 - cs 0000
 - ip 0000
- Memory Dump:** A table of memory values at the bottom:
 - ds:0000 00 00 00 00 54 75 72 62 Turb
 - ds:0008 6F 20 43 20 2D 20 43 6F o C - Co
 - ds:0010 70 79 72 69 67 68 74 20 pyright
 - ds:0018 31 39 38 39 20 42 6F 72 1989 Bor
 - ds:0020 6C 61 6E 64 20 49 6E 74 land Int
 - ss:010E 0002
 - ss:010C 2E8B
 - ss:010A 21CD
 - ss:0108 30B4
 - ss:0106 03FA

Q3) Subtract two 32 - bit numbers in data segment and store the result in data segment

DATA

SEGMENT

X DD 15489657H

Y DD 12487596H

RES DD ?

CARRY DB ?

DATA ENDS

CODE SEGMENT

ASSUME CS: CODE, DS: DATA

START:

MOV AX, DATA

MOV DS, AX

MOV AX, WORD PTR X

MOV BX, WORD PTR Y

SUB AX, BX

MOV WORD PTR RES, AX

MOV AX, WORD PTR X+2

MOV BX, WORD PTR Y+2

SBB AX, BX

MOV WORD PTR RES+2, AX

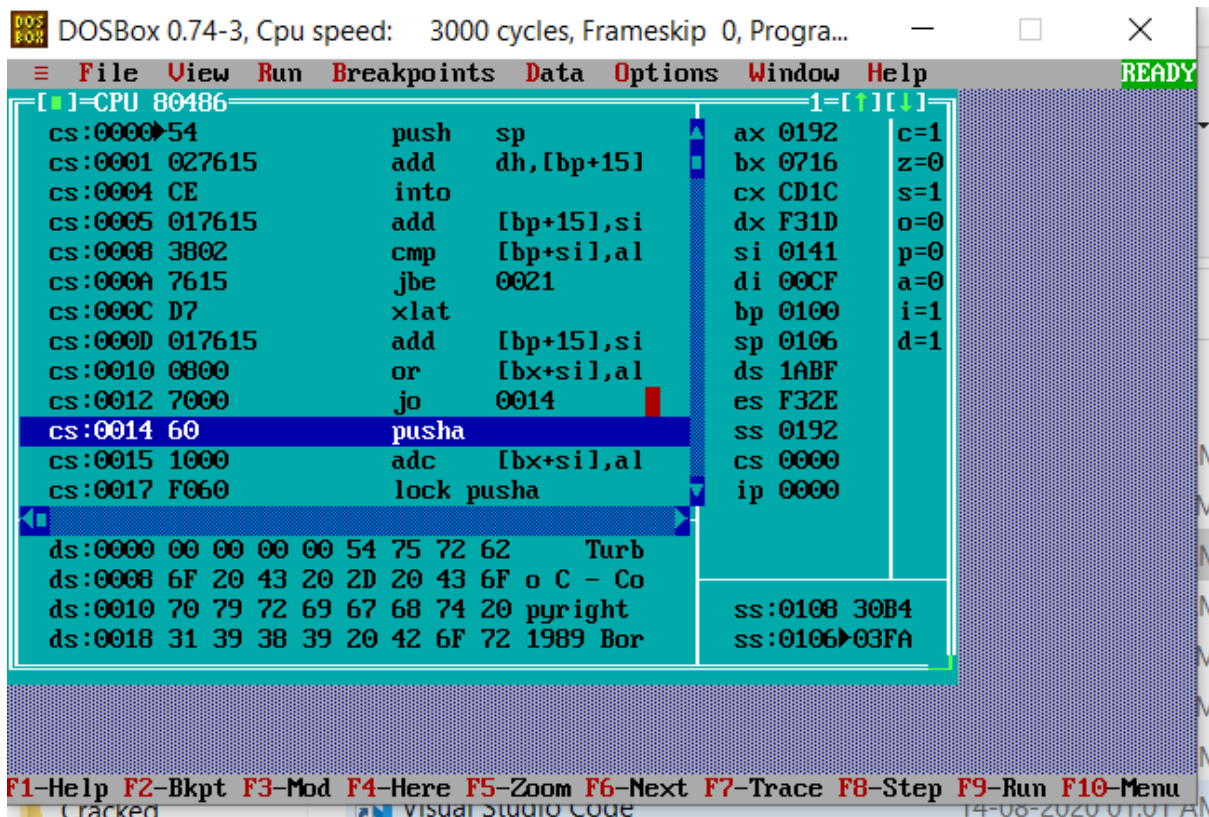
ADC CARRY, 0

MOV AH, 4CH

INT 21H

CODE ENDS

END START



Q4) Add two 64 – bit decimal numbers in data segment and store the result in data segment

DATA

SEGMENT

X DQ 9876543298765432H

Y DQ 1234567812345678H

RES DQ ?

CARRY DB ?

DATA ENDS

CODE SEGMENT

ASSUME CS: CODE, DS: DATA

START:

MOV AX, DATA

MOV DS, AX

MOV AX, WORD PTR X

MOV BX, WORD PTR Y

ADD AX, BX

MOV WORD PTR RES, AX

MOV AX, WORD PTR X+2

MOV BX, WORD PTR Y+2

ADC AX, BX

```

MOV WORD PTR RES+2, AX

MOV AX, WORD PTR X+4
MOV BX, WORD PTR Y+4
ADC AX, BX
MOV WORD PTR RES+4, AX

MOV AX, WORD PTR X+6
MOV BX, WORD PTR Y+6
ADC AX, BX
MOV WORD PTR RES+6, AX

ADC CARRY, 0

MOV AH, 4CH
INT 21H

CODE ENDS
END START

```

The screenshot shows the DOSBox 0.74-3 interface. The title bar reads "DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...". The menu bar includes File, View, Run, Breakpoints, Data, Options, Window, and Help. The status bar at the bottom shows function key shortcuts: F1-Help, F2-Bkpt, F3-Mod, F4-Here, F5-Zoom, F6-Next, F7-Trace, F8-Step, F9-Run, and F10-Menu.

The main window displays assembly code on the left and register values on the right. The code is as follows:

```

cs:0000 54      push    sp
cs:0001 027615   add     dh,[bp+15]
cs:0004 CE      into
cs:0005 017615   add     [bp+15],si
cs:0008 3802     cmp     [bp+si],al
cs:000A 7615     jbe     0021
cs:000C D7      xlat
cs:000D 017615   add     [bp+15],si
cs:0010 0800     or      [bx+si],al
cs:0012 7000     jo      0014
cs:0014 60      pusha
cs:0015 1000     adc     [bx+si],al
cs:0017 F060     lock pusha

```

The register window on the right shows the following values:

ax	0192	c	1
bx	0716	z	0
cx	CD1C	s	1
dx	F31D	o	0
si	0141	p	0
di	00CF	a	0
bp	0100	i	1
sp	0106	d	1
ds	1ABF		
es	F32E		
ss	0192		
cs	0000		
ip	0000		

Below the code, a memory dump is visible for the data segment (ds):

```

ds:0000 00 00 00 00 54 75 72 62      Turb
ds:0008 6F 20 43 20 2D 20 43 6F o C - Co
ds:0010 70 79 72 69 67 68 74 20 pyright
ds:0018 31 39 38 39 20 42 6F 72 1989 Bor

```

The status bar at the bottom right shows "READY".

Q5) Find the sum of 10 unsigned bytes in an array and store the result in data segment

DATA

SEGMENT

```

SRC DB 01H, 02H, 03H, 04H, 05H, 06H, 07H, 08H, 09H, 0AH
COUNT DB 10

```

```

        RES DW ?
DATA ENDS

CODE SEGMENT
        ASSUME CS: CODE, DS: DATA
START:
        MOV AX, DATA
        MOV DS, AX

        MOV CH, 00
        MOV CL, COUNT

        LEA SI, SRC
        MOV BX, 0

        CLC

SUM:
        mov AX, 0
        MOV AL, BYTE PTR [SI]
        ADC BX, AX
        INC SI
        LOOP SUM
        MOV RES, BX

        MOV AH, 4CH
        INT 21H

CODE ENDS
END START

```


DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...

File
View
Run
Breakpoints
Data
Options
Window
Help

[+] CPU 80486

cs:0000 54 push sp
cs:0001 027615 add dh,[bp+15]
cs:0004 CE into
cs:0005 017615 add [bp+15],si
cs:0008 3802 cmp [bp+si],al
cs:000A 7615 jbe 0021
cs:000C D7 xlat
cs:000D 017615 add [bp+15],si
cs:0010 0800 or [bx+si],al
cs:0012 7000 jo 0014
cs:0014 60 pusha
cs:0015 1000 adc [bx+si],al
cs:0017 F060 lock pusha
cs:0019 1000 adc [bx+si],al
cs:001B F060 lock pusha

ax 0192 c=1
bx 0716 z=0
cx CD1C s=1
dx F31D o=0
si 0141 p=0
di 00CF a=0
bp 0100 i=1
sp 0106 d=1
ds 1ABF
es F32E
ss 0192
cs 0000
ip 0000

ds:0000 00 00 00 00 54 75 72 62 Turb
ds:0008 6F 20 43 20 2D 20 43 6F o C - Co
ds:0010 70 79 72 69 67 68 74 20 pyright
ds:0018 31 39 38 39 20 42 6F 72 1989 Bor
ds:0020 6C 61 6E 64 20 49 6E 74 land Int

ss:0108 30B4
ss:0106 03FA
ss:0104 1689
ss:0102 2E1A
ss:0100 BFBA

F1-Help F2-Bkpt F3-Mod F4-Here F5-Zoom F6-Next F7-Trace F8-Step F9-Run F10-Menu