



Coal

LAB TASK 5

NAME SAJID ISLAM

ROLLNO 24p-0745

INSTRUCTOR Muhammad Hassan

Question 1

```
[org 0x0100]

mov cx, 7      ; loop counter
mov ax, 0      ; clear accumulator
mov bx, numbers ; point to numbers array

loop_start:
add ax, [bx]   ; add value to ax
add bx, 2      ; move to next word
dec cx        ; decrement counter
jnz loop_start ; loop until cx=0

mov [result], ax ; store result

mov ax, 0x4c00 ; exit program
int 0x21

numbers: dw 1,2,3,4,5,6,7
result:  dw 0
```

Output

DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: AFD

AX 0003SI 0000CS 19F5IP 010BStack +0 0000Flags 7204

BX 011CDI 0000DS 19F5+2 20CD

CX 0006BP 0000ES 19F5HS 19F5+4 9FFFOF DF IF SF ZF AF PF CF

DX 0000SP FFFESS 19F5FS 19F5+6 EA000 0 1 0 0 0 1 0

{R} reg=value

CMD >SS

0109 0307ADDAX,[BX]

010B 81C30200ADDBX,0002

010F 49DEC CX

0110 75F7JNZ 0109

0112 A32B01MOV[0128],AX

0115 B8004CMOVBX,4C00

0118 CD21INT 21

011A 0100ADD[BX+SI],AX

011C 0200ADDAL,[BX+SI]

101234567

DS:0000CD 20 FF 9F 00 EA F0 FEAD DE 1B 05 C5 06 00 00

DS:0008AD DE 1B 05 C5 06 00 00

DS:001018 01 10 01 18 01 92 01

DS:001801 01 01 00 02 FF FF FF

DS:0020FF FF FF FF FF FF FF

DS:0028FF FF FF FF EB 19 C0 11

DS:0030A2 01 14 00 18 00 F5 19

DS:0038FF FF FF FF 00 00 00 00

DS:004005 00 00 00 00 00 00 00

DS:004800 00 00 00 00 00 00 00

20123456789ABCDEF

DS:0000CD 20 FF 9F 00 EA F0 FEAD DE 1B 05 C5 06 00 00= f.Ω■ i |..†...

DS:001018 01 10 01 18 01 92 0101 01 01 00 02 FF FF FF.....ft.

DS:0020FF FF FF FF FF FF FFFF FF FF FF EB 19 C0 11δ.L.

DS:0030A2 01 14 00 18 00 F5 19FF FF FF FF 00 00 00 00ó.....J.

DS:004005 00 00 00 00 00 00 0000 00 00 00 00 00 00.....

Question 2

```
[org 0x0100]

xor ax, ax      ; clear ax
mov bx, 0       ; index

loop_start:
add ax, [data + bx] ; add value at data+bx to ax
add bx, 2        ; increment index
cmp bx, 20       ; check if end of array
jne loop_start   ; loop if not done

mov [result], ax ; store result

mov ax, 0x4c00   ; exit program
int 0x21

data: dw 1,2,3,4,5
result: dw 0
```

used CMP BX, 20 and JNE to stop when BX reaches the end of the array.

Output

Screen Shot after first iteration

DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: AFD

AX 0003 SI 0000 CS 19F5 IP 0105 Stack +0 0000 Flags 7285
BX 0004 DI 0000 DS 19F5 +2 20CD
CX 0027 BP 0000 ES 19F5 HS 19F5 +4 9FFF OF DF IF SF ZF AF PF CF
DX 0000 SP FFFE SS 19F5 FS 19F5 +6 EA00 0 0 1 1 0 0 1 1

S or SI or SYM
CMD >S

0111 75F2 JNZ 0105
0105 03871B01 ADD AX,[011B+BX]
0109 81C30200 ADD BX,0002
010D 81FB1400 CMP BX,0014
0111 75F2 JNZ 0105
0113 A32501 MOV [0125],AX
0116 B8004C MOV AX,4C00
0119 CD21 INT 21
011B 0100 ADD [BX+SI],AX

1 0 1 2 3 4 5 6 7
DS:0000 CD 20 FF 9F 00 EA F0 FE
DS:0008 AD DE 1B 05 C5 06 00 00
DS:0010 18 01 10 01 18 01 92 01
DS:0018 01 01 01 00 02 FF FF FF
DS:0020 FF FF FF FF FF FF FF FF
DS:0028 FF FF FF FF EB 19 C0 11
DS:0030 A2 01 14 00 18 00 F5 19
DS:0038 FF FF FF FF 00 00 00 00
DS:0040 05 00 00 00 00 00 00 00
DS:0048 00 00 00 00 00 00 00 00

2 0 1 2 3 4 5 6 7 8 9 A B C D E F
DS:0000 CD 20 FF 9F 00 EA F0 FE AD DE 1B 05 C5 06 00 00 = f.Ω≡ i |..†...
DS:0010 18 01 10 01 18 01 92 01 01 01 01 00 02 FF FF FFff.
DS:0020 FF FF FF FF FF FF FF FF FF FF FF FF EB 19 C0 11 δ.L.
DS:0030 A2 01 14 00 18 00 F5 19 FF FF FF FF 00 00 00 00 ó.....J.
DS:0040 05 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
.....

1 Step 2ProcStep 3Retrieve 4Help ON 5BRK Menu 6 7 up 8 dn 9 le 10 ri

Question 3

```
[org 0x0100]

mov ax, [array] ; moving element at first index into ax
mov bx, 2

loop_start:
cmp ax, [array + bx] ; compare ax with current element
jg no_swap           ; if ax > [array+bx], skip
jl swap             ; if ax < [array+bx], update max

swap:
mov ax, [array + bx] ; update ax with new max

no_swap:
add bx, 2            ; move to next word
cmp bx, 10           |
jne loop_start       ; loop if not done

mov [maximum], ax    ; store maximum value

mov ax, 0x4c00
int 0x21

array:  dw 100, 20, 30, 40, 50
maximum: dw 0
```

Output

Screenshot after storing the largest number in memory

```

DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: AFD
AX 0064 SI 0000 CS 19F5 IP 011F Stack +0 0000 Flags 7244
BX 000A DI 0000 DS 19F5 +2 20CD
CX 0000 BP 0000 ES 19F5 HS 19F5 +4 9FFF OF DF IF SF ZF AF PF CF
DX 0000 SP FFFE SS 19F5 FS 19F5 +6 EA00 0 0 1 0 1 0 1 0
S or SI or SYM
CMD >S
011C A32E01 MOV [012E],AX
011F B8004C MOV AX,4C00
0122 CD21 INT 21
0124 64 DB 64
0125 0014 ADD [SI],DL
0127 001E0028 ADD [2800],BL
012B 0032 ADD [BP+SI],DH
012D 006400 ADD [SI+00],AH
0130 07 POP ES
1 0 1 2 3 4 5 6 7
DS:0000 CD 20 FF 9F 00 EA FF FF
DS:0008 AD DE 1B 05 C5 06 00 00
DS:0010 18 01 10 01 18 01 92 01
DS:0018 01 01 01 00 02 FF FF FF
DS:0020 FF FF FF FF FF FF FF FF
DS:0028 FF FF FF FF EB 19 E6 11
DS:0030 A2 01 14 00 18 00 F5 19
DS:0038 FF FF FF FF 00 00 00 00
DS:0040 05 00 00 00 00 00 00 00
DS:0048 00 00 00 00 00 00 00 00
2 0 1 2 3 4 5 6 7 8 9 A B C D E F
DS:0000 CD 20 FF 9F 00 EA FF FF AD DE 1B 05 C5 06 00 00 = f.0 i |. + ...
DS:0010 18 01 10 01 18 01 92 01 01 01 01 00 02 FF FF FF .....ff. ....
DS:0020 FF FF FF FF FF FF FF FF FF FF FF FF FF EB 19 E6 11 ..... δ.μ.
DS:0030 A2 01 14 00 18 00 F5 19 FF FF FF FF 00 00 00 00 ó.....J. ....
DS:0040 05 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....

```

Question 4

Zero Flag

```

[org 0x0100]

mov ax, 99
sub ax, 99      ; ZF = 1

mov bx, 6
sub bx, 5      ; ZF = 0

mov ax, 0x4c00
int 0x21

```

Output

After executing the subtraction command we get the below results

```
DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: AFD
AX 0000 SI 0000 CS 19F5 IP 0106 Stack +0 0000 Flags 7244
BX 0000 DI 0000 DS 19F5      +2 20CD
CX 0000 BP 0000 ES 19F5 HS 19F5  +4 9FFF OF DF IF SF ZF AF PF 0
DX 0000 SP FFFE SS 19F5 FS 19F5  +6 EA00 0 0 1 0 1 0 1
S or SI or SYM
CMD >S
1
DS:0000 CD 20 FF 9F 00 EA FF
DS:0008 AD DE 1B 05 C5 06 00
DS:0010 18 01 10 01 18 01 92
0103 2D6300 SUB AX,0063
0106 BB0600 MOV BX,0006
```

Zero flag becomes zero again after executing the end command

```
CX 0000 BP 0000 ES 19F5 HS 19F5 +4 9FFF OF DF IF SF ZF AF PF
DX 0000 SP FFFE SS 19F5 FS 19F5 +6 EA00 0 0 1 0 0 0 0
```

Sign flag

[org 0x0100]

```
mov ax, 5
sub ax, 7      ; SF = 1

mov bx, 10
sub bx, 5      ; SF = 0

mov ax, 0x4c00
int 0x21
```

Output

AX	FFFE	SI	0000	CS	19F5	IP	0106	Stack	+0	0000	Flags	7291
BX	0000	DI	0000	DS	19F5				+2	20CD		
CX	0012	BP	0000	ES	19F5	HS	19F5		+4	9FFF	OF	DF IF SF ZF AF PF
DX	0000	SP	FFFE	SS	19F5	FS	19F5		+6	EA00	0	0 1 1 0 1 0
S or SI or SYM												
CMD >S								1			0	1 2 3 4 5 6
0103	2D0700			SUB	AX,0007				DS:0000	CD	20 FF 9F 00 EA F0	
0106	BB0A00			MOV	BX,000A				DS:0008	AD DE 1B 05 C5 06 00		
									DS:0010	18 01 10 01 18 01 92		

The sign flag again becomes zero at end command

AX	4C00	SI	0000	CS	19F5	IP	0110	Stack	+0	0000	Flags	7204
BX	0005	DI	0000	DS	19F5				+2	20CD		
CX	0012	BP	0000	ES	19F5	HS	19F5		+4	9FFF	OF	DF IF SF ZF AF P
DX	0000	SP	FFFE	SS	19F5	FS	19F5		+6	EA00	0	0 1 0 0 0 0
S or SI or SYM												
CMD >S								1			0	1 2 3 4 5
010D	BB004C			MOV	AX,4C00				DS:0000	CD	20 FF 9F 00 EA F	
0110	CD21			INT	21				DS:0008	AD DE 1B 05 C5 06 0		
									DS:0010	18 01 10 01 18 01 9		

Parity Flag

```
[org 0x0100]

mov al, 9
add al, 1      ; PF = 1 (even parity)

mov ax, 0x4c00
int 0x21
```

Output

AX 000A	SI 0000	CS 19F5	IP 0104	Stack +0 0000	Flags 7204
BX 0000	DI 0000	DS 19F5		+2 20CD	
CX 0009	BP 0000	ES 19F5	HS 19F5	+4 9FFF	OF DF IF SF ZF AF PF CF
DX 0000	SP FFFE	SS 19F5	FS 19F5	+6 EA00	0 0 1 0 0 0 1 0
S or SI or SYM					
CMD >S				1	0 1 2 3 4 5 6 7
				DS:0000	CD 20 FF 9F 00 EA F0 FE
0102 0401				DS:0008	AD DE 1B 05 C5 06 00 00
ADD AL,01					

Auxiliary Flag

```
[org 0x0100]

mov ax, 255
add ax, 1      ; AF = 1 (carry from lower nibble)

mov bx, 0x05
add bx, 1      ; AF = 0

mov ax, 0x4c00
int 0x21
```

Output

Auxiliary flag become on

AX 0100	SI 0000	CS 19F5	IP 0106	Stack +0 0000	Flags 7214
BX 0000	DI 0000	DS 19F5		+2 20CD	
CX 0000	BP 0000	ES 19F5	HS 19F5	+4 9FFF	OF DF IF SF ZF AF PF CF
DX 0000	SP FFFE	SS 19F5	FS 19F5	+6 EA00	0 0 1 0 0 1 1 0
S or SI or SYM					
CMD >S				1	0 1 2 3 4 5 6 7
				DS:0000	CD 20 FF 9F 00 EA FF FF
0103 050100				DS:0008	AD DE 1B 05 C5 06 00 00
ADD AX,0001					

Auxiliary flag become zero again

AX 0100	SI 0000	CS 19F5	IP 010D	Stack +0 0000	Flags 7204							
BX 0006	DI 0000	DS 19F5		+2 20CD								
CX 0000	BP 0000	ES 19F5	HS 19F5	+4 9FFF	OF DF IF SF ZF AF PF CF	0	0	1	0	0	0	1
DX 0000	SP FFFE	SS 19F5	FS 19F5	+6 EA00		0	0	1	0	0	0	1
S or SI or SYM												
CMD >S				1	0	1	2	3	4	5	6	7
				DS:0000	CD	20	FF	9F	00	EA	FF	FF
0109 81C30100				ADD	BX,0001							
010D B8004C				MOV	AX,4C00	DS:0008	AD	DE	1B	05	C5	06
						DS:0010	18	01	10	01	18	01

Question 5

```

[org 0x0100]          ; Origin address

mov bx, data           ; BX points to start of data array
mov dx, result         ; DX points to start of result array
mov cx, 6              ; We have 6 numbers to process

next_num:
    mov ax, [bx]        ;
    cmp ax, 0           ; Compare with 0
    jne non_zero        ; Jump if not equal to 0

; if the number is zero
    mov byte [dx], 0    ;
    jmp store_done

non_zero:
    mov byte [dx], 1    ; Store 1 in result array

store_done:
    add bx, 2           ; Go to next number (word = 2 bytes)
    inc dx              ; Next result (byte)
    loop next_num       ; Repeat until CX=0

mov ax, 0x4c00          ; DOS terminate
int 0x21

data:  dw 0, -5, 12, 7, -128, 255
result: db 0, 0, 0, 0, 0, 0

```

Output

This is the final output

```

DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: ARD
AX 0000 SI 0127 CS 19F5 IP 0109 Stack +0 0000 Flags 7200
BX 0000 DI 0132 DS 19F5 +2 20CD
CX 0005 BP 0000 ES 19F5 HS 19F5 +4 9FFF OF DF IF SF ZF AF PF CF
DX 0000 SP FFFE SS 19F5 FS 19F5 +6 EA00 0 0 1 0 0 0 0 0
S or SI or SYM
CMD >S
1 0 1 2 3 4 5 6 7
DS:0000 CD 20 FF 9F 00 EA F0 FE
DS:0008 AD DE 1B 05 C5 06 00 00
DS:0010 18 01 10 01 18 01 92 01
DS:0018 01 01 01 00 02 FF FF FF
DS:0020 FF FF FF FF FF FF FF FF
DS:0028 FF FF FF FF EB 19 C0 11
DS:0030 A2 01 14 00 18 00 F5 19
DS:0038 FF FF FF FF 00 00 00 00
DS:0040 05 00 00 00 00 00 00 00
DS:0048 00 00 00 00 00 00 00 00
2 0 1 2 3 4 5 6 7 8 9 A B C D E F
DS:0000 CD 20 FF 9F 00 EA F0 FE AD DE 1B 05 C5 06 00 00 = f.Ω≡ i|..+...
DS:0010 18 01 10 01 18 01 92 01 01 01 01 00 02 FF FF FF .....ff. ....
DS:0020 FF FF FF FF FF FF FF FF FF FF FF FF EB 19 C0 11 δ.L.

```