

Question 2 [2+2+4+1+1+3=13Marks]

ive Short answers:

1. .data

[2 marks]

arr1 dw 1,2,3

If we want to move to the second index of this array i.e. the value 2. How will we write it in:

- Indexed addressing format: mov esi, arr1[esi] (1)
- Indirect addressing format: mov esi, offset arr1 (1)

2. In 16-bit indexed operands what registers are used? Name all four registers. [2 marks]

SI, DI, BX, BP. 0.5 each

3. Highlight if there is any error in the below mentioned code and also correct them. Writing errors unnecessarily will result in negative marking.

0.5 .data
Var1 db 256 out of range
Var2 dw ?
Var3 dd ?

[2 marks]

- a. Mov al, var1 ✓
- 0.25 Push al ✗ push ax / min 16-bits
- 0.5 Mov ah, var2 ✗ size mismatch
- 0.25 Push ah push ax min 16 bits
- 0.25 Pop al pop ax
- 0.25 Pop ah pop ax

b. Check if the code mentioned below is correct or not. You cannot delete anything in this code. [2 marks]

However you may add any line of code.

JMP main

proc1 proc

Push cx

mov cx,3

call proc3

l1:

add ax,1

LOOP l1

RET

proc1 ENDP

proc3 proc

l3:

add dx,1

loop l3

RET

proc3 ENDP

```
main proc
    mov ax,@data
    mov ds,ax
    mov ax,0
    mov bx,0
    mov dx,0
    mov cx,2
    call proc1
main EndP
```

4. If in a procedure "ret 4" is written what will it do? R. Address + 4 bytes [1 mark]
5. If ECX is initialized to zero before beginning a loop, how many times will the LOOP instruction repeat? (Assume ECX is not modified by any other instructions inside the loop.) infinite [1 mark]
6. Find the values of SizeOf, LengthOf and Type operators. [3 marks]

.data	SizeOf	LengthOf	Type
V1 byte 11,22,33,44,55	5	5	1
V2 word 15 Dup(0),5,7,8,10	10 38	5 19	2
V3 dword 4 Dup(10 Dup(4))	160	40	4
V4 word 1,2,3,4,5,6, 7,8,8,7 Word 7,8,6,9,8,9	20	10	2

0.25 each

Question 3 [15 Marks]

Write the values of different registers in Hexadecimal and also fill memory.

myBytes BYTE 10h, 'A', 11001101b, 40h
myWords WORD 1Ah, 3Bh, 72h, 44h, 66h
myDoubles DWORD 1,2,3,4,5
varD LABEL DWORD
varW LABEL WORD
myBytes2 BYTE 54, 67, 80, 0ABh
myBytes3 SBYTE -4, -2, 3, 1
myWords2 WORD 3 DUP(?), 2000h
myString db sizeof myWords2 DUP(type myDoubles DUP(1))

CP
8meh

10-A
11-B
12-C
13-D
14-E
15-F

→ 1 mark extra.

	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
0000	10	41h	CD	40	1A	00	3B	00	72	00	44	00	66	00	01	00
0010	00	00	02	00	00	00	03	00	00	00	04	00	00	00	05	00
0020	00	00	36h	43h	50h	ABh	FEh	FEh	03	01	??	??	??	??	??	??

JA/JNBE

(CF and ZF) = 0

Above / Not Below or Equal

IAE/JNB

CF = 0

Above or Equal / Not Below

JB/JNAE/JC

CF = 1

Below / Not Above or Equal / Carry

JBE/JNA

(CF or ZF) = 1

Below or Equal / Not Above

JE/JZ

ZF = 1

Equal / Zero

Question 5 [6 Marks]

(a) Update flags(carry, overflow, auxiliary, parity and zero flag) after following operations:

1) 7456h + FFFh

[3 Marks]

$$\begin{array}{r} \textcircled{1} \textcircled{0} \textcircled{0} \\ 7456 \\ + FFF \\ \hline 7454 \end{array}$$

CF = 1

ZF = 0

~~SF~~ SF = 0

OF = 0

PF = 0

AF = 1

2) 15A0h - 8588h

[3 Marks]

$$\begin{array}{r} 15A0 \\ - 8588 \\ \hline 9018 \end{array}$$

~~15~~

-0.5 if not in hex

CF = 1

ZF = 0

opt. SF = 1

OF = 1

PF = 1

AF = 1

Question 6[22 Marks]

Consider following code and complete the missing code in underscored areas, then **FILL STACK**. After executing complete code, update registers. Also show how **STACK** will look after execution. Data segment starts (Array offset) at **0000900DH**. **NOTE: Consider Line number as Instruction address. Clearly mention starting address on which line number.**

```

1 .model small
2 .stack 0100h
3 .data
4 ary dd 2,3,4
5 .code
6 jmp main
7 SumArray proc
8     push ebp
9     mov ebp,esp
10    mov esi, [ebp+12] ;offset of ary
11    mov ecx, [ebp+8] ;lengthof ary
12    cmp ecx,0
13    je L1
14    add eax,[esi]
15    dec ecx
16    add esi,4 ;update esi to move to
    next index
17    push esi
18    push ecx
19    call SumArray
20    L1:
21        pop ebp
22    ret 8
23 SumArray endp
24 main proc
25    mov ax,@data
26    mov ds,ax
27    push edx
28    push edi
29    push ebx
30    mov eax, offset ary
31    push eax
32    mov eax,lengthof ary
33    push eax
34    mov eax,0
35    mov esi,0
36    mov ecx,0
37    call SumArray
38 main endp
    
```

EAX=	ECX=	ESI=	EBP=
00000009	0	00000010	0

Offset	Stack filling	Stack at the end
00000FB0		
00000FB4	FC4	
00000FB8	RA (20)	
00000FBC	0	
00000FC0	19	
00000FC4	FD4	
00000FC8	RA 20	
00000FCC	1	
00000FD0	15	
00000FD4	FE4	
00000FD8	RA (20)	
00000FDC	2	
00000FE0	11	
00000FE4	0	
00000FE8	RA 38	
00000FEC	3	
00000FF0	0000000D	
00000FF4	0000	0000
00000FF8	0000	0000
00000FFC	0000	0000

Question 7 [8 Marks]

Consider the following data declaration. Fill in the given memory in hexadecimal:

.data
 word2 dw -22
 .tbl BYTE 1,2
 quad1 dq 23ABEF89AC123601h
 list2 db 10, 041h, 'A', 00111111b
 string BYTE 'ABC', 0
 list4 WORD 2 DUP(0AB12h)
 quad3 QWORD 'EF'

} 1 mem each

	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
0000	EA	FF	01	02	01	36	12	AC	89	ef	ab	23	0A	41h	41h	3F
0010	41h	42h	43	0	12	AB	12	AB	46	45	00	00	00	00	00	00
0020																