Question 2 [8 Marks]

Update Flag register value after execution of the CMP statement. Mark ✓ in taken or not taken box for each instruction respectively.

Instructions	Taken	Not Taken
Mov al,97		
Cmp al,-5		
Ja L1		✓
L1: jg L2	✓	
L2: jnb L3		✓
L3: jnge L4		✓
L4: jbe L5	✓	
L5: jng L7		✓
L6:jmp end	✓	
L7:		
End:		

Flags		
	Sign	0
	Zero	0
	Carry	1
	Overflow	0
	Parity	1
	Auxiliary	1

Calculations:

0110 0001
1111 1011
0110 0110

Question 3 [10 Marks]

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

1) 7CBDh + FFFEh **[3 Marks]**

7CBD
+ FFFE
7CBB

CF = 1
OF = 0
AC = 1
PF = 1
Z = 0

2) 15A0h - 8547h **[3 Marks]**

15A0
- 8547
9059

CF = 1
ZF = 0
OF = 1
PF = 1
AF = 1

(b) Write an assembly code that finds even parity for the most significant BYTE of si register, where si register is a 16-bit register. Write code after given lines that will update parity bit in the FLAGS register. [4 Marks]

mov si,0F798H

```
mov  AX, SI
mov  Bl, AH
add  Bl, 00
```

Question 4 [8 + 6=14 Marks]

(a) Consider the following data declaration and fill instruction after every instruction. Assume all registers have zero value at the start of execution of code [8 marks]

ef cd ab 00 01 0000 28

0 1 2 3 4 2

7×4=28 reserve 00 01 56

```
L1 LABEL BYTE
L2 LABEL WORD
L3 DD 0abcdefh,1,2,3,4,5,6
L4 LABEL BYTE
L5 WORD 1,2
L6 db sizeof L3 DUP(type L5 DUP(1))
```

L6 db 28 dup (2 dup(1))

Mov AL , L1	AL= ef
Mov AL , type L3	AL= 04
Mov AH , sizeof L3	AH= 1C (28)
Mov BL , lengthof L3	BL= 07
Mov BH , BYTE PTR (L3 +1)	BH= CD
Mov CL , L1	CL= EF
Mov BX , L2	BX= CDEF
Mov AX, sizeof L6	AX= 38h (56d)
Mov DX. (L2+4)	DX= 0001

(b) Find the values of SizeOf, LengthOf and Type operators. [6 marks]

.data	SizeOf	LengthOf	Type
V1 byte 11,22,33,44,55,66	6	6	1
V2 word 15 Dup(0),5,7,10	36	18	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

Question 5 [10 Marks]

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

```
.data
word2 dw -22
list1 BYTE 1,2
quad1 dq 23ABEF89AC123601h
list2 db 10, 041h, 'A', 00111111b
string BYTE 'ABC',0
list4 WORD 2 DUP(0AB12h)
quad3 QWORD 'EF'
```

	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	41	41	BF
0010	41	42	43	00	12	AB	12	AB	46	45	00	00	00	00	00	00
0020																

(b) Update value of registers after each line of code.

```
Bval1 db 034h, 012h
Wval2 dw 0ABCDh
Dval3 dd 0ABCDEF12h
```

12 EF CD AB
0 1 2

```
Mov ax, WORD PTR bval1
Mov al, BYTE PTR wval2
Mov bx, WORD PTR dval3
Mov cx, WORD PTR [dval3+2]
```

	H	L
Ax	12	34
Ax	12	CD
Bx	EF	12
Cx	AB	CD