

## National University of Computer and Emerging Sciences

FAST School of Computing

Spring-2023

Question 1 [10 Marks]

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

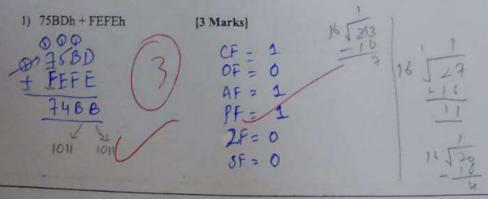
Instructions	Taken	Not Taken				Calculations: FF.
Mov al, 72		-		Latin		11 000 40
Cmp al,-15		11	Flags	Sign	0	FEEL
Ja L1	/	V,		Zero	0	
L1: jg L2	1	/		Carry	1	1 000+
L2: jnb L3		///	1	Overflow	0	010   011
L3: jnge L4		, _		Parity	0	
L4: jbe L5	1	- 1		Auxiliary	0	
L5; jng L7		1				
L6:jmp end	M			(1-)		
L7:				(10)		
End:						

ZF = 0 and SF = OF JG/INLE SF - OF JGE/JNL SF o OF JL / INGE (ZF = 1) or (SF <> OF) (CF and ZF) = 0 CF-0 CF = 1 JB/JNAZ/JC (CF or ZF) = 1 THE/TNA

Greater / Not Less nor Equal Grater or Equal / Not Less Less / Not Greater not Equal Less or equal / not greater re / Not Below or Equal ve or Equal / Not Below w / Not Above or Equal / Carry Below or Equal / Not Above

## Question 2 [6+10 Marks]

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



National University of Computer and Emerging Sciences 0090 Islamabad Campus

2) I2B0h - 8347h

[3 Marks] CF - 1 ZF = 0 6F = 1

(b) You may use jumps for the conversion of IF ELSE statement and WHILE loop. Write proper data segment for initializing variables used in code and code segment for writing code. [10 marks

Note: all variables are 16-bit unsigned integers.

```
Int array[]={10,60,20,33,72};
Int index=0;
Int sample=40;
array_size=sizeof array/sizeof sample;
while(index<array_size){
          if(array[index]<sample)
                    var1= var1+1;
                    index++; }
          else
                    Index--;
```

· Model Small · Stack 100h

> · date orray dw 10,60,20,33,72 sample du 40 var 1 dw 10

· code

jac execut southop.

jac execut southop.

jac execut add vow1, 1 odd irder, 2 my startloop

National University of Computer and Emerging Sciences Islamabad Campus FAST School of Computing | Spring-2023 bx STREOF array else cond: comp index bx... mov ax, HORO PTR (array + index) = " Tridex) adi cmp ax, sample jae else cood add var1, 1 wif part add index, 1 0 if part ing startloop Question 3 [20+6+4=30 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. Also fill memory of these data declaration [20 marks] LI LABEL BYTE L2 LABEL WORD 00 00 00 00 L3 DD 66ab77efh,2,3,4,5 L4 LABEL WORD Bytel SBYTE -4,-2,3,1 \_\_\_ DO DUP ( 1 PUP (11) L6 db sizeof L3 DUP(type Byte1 DUP(1)) Mov AL . L1 AL= Mov AX, L4 AX= Mov AX,(L4+2) AH= Mov BL, SIZEOF L3 BL= tin hear = 14) Mov BH, BYTE PTR (L3+1) BH= Mov CL, (L1+2) CL= ab Mov BX, L2 BX= Mov AX, sizeof L6 AX= Mov DX, (L2+4) DX= 00 02 Mov esi, Offset L3 AX= Mov ax, [esi-4]

est= 0000 +00

## National University of Computer and Emerging Sciences 0090

FAST School of Computing Spring-2023 Islamabad Campus 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E OF 0000 4 02 00 00 00 64 50 00 00 50 00 50 0010 8 50 03 01 61 01 61 01 01 01

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

.data	SizeOf	LengthOf	Туре
V1 byte 11,22,33,44,55	5	5	1
V2 word 15 Dup(0),5,7	34	17	2/
V3 dword 4 Dup(10 Dup(4))	160	40	4_
V4 word 1,2,3,4,5,6, 7,8,8 Word 7,8,6,9,8,9	18	9	2

(c) Write a code that adds v1 and v2 and store result in sum. Your code should be x86 architecture.

[4 marks]

.data

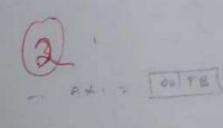
V1 db -5

V2 dw OFFFCh

Sum dw 0

add ax, V2

mov sum, ax



Bvall db 034h, 022h

Wval2 dw 0ABCDh

Dval3 dd 0AB22EFI2h

Married Street
100
100
0
200
100
60
101
929
0.000
Distance of the last
100
5391
0
Pages 1
70
1
Page 1
Property /
Hein
Section 1
20
-
Cale
100
100
50
Total Control
1
1

Mov al, BYTE PTR wval2

Mov bx, WORD PTR dval3

Mov cx, WORD PTR [dval3+2]

Ax

H

22	20
6	34

Bx

Ax

7	B	EF	لمد	22	
	20	74	6	34	

OV 03 10 0 AB 6 34 T COAFA B1152