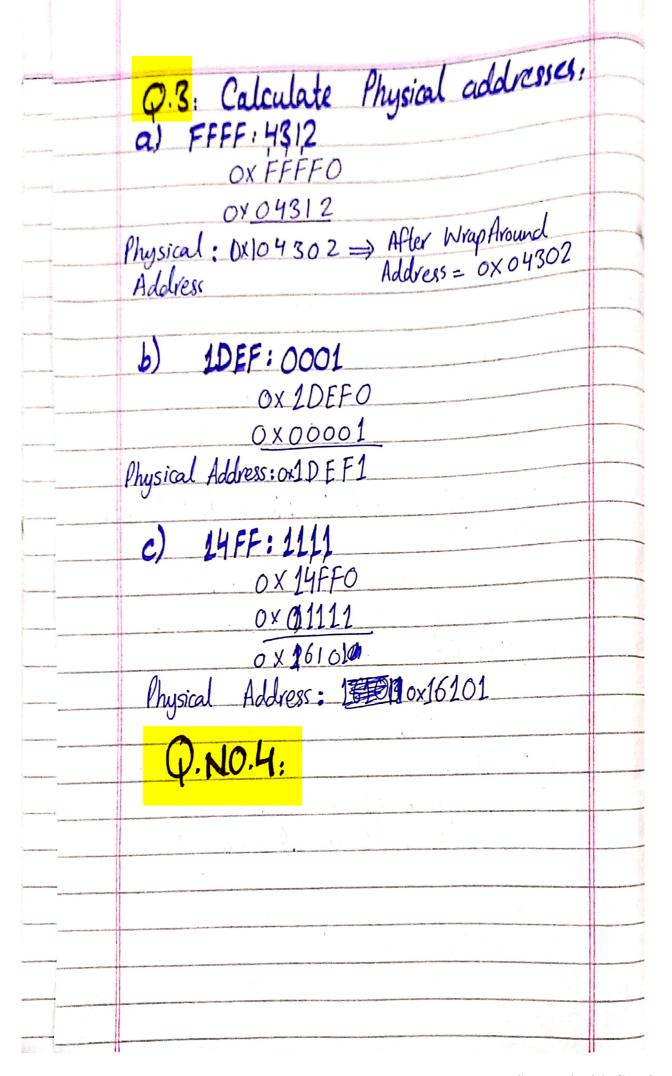
COMPUTER ORGANIZATION	
COMPUTER	
AND ASSEMBLY LANGUAGE:	
Assegnment-I;	
STUDENT AME:	
SECTION: BCS-3B	
Roy No .:	
	K . I
Submission Date: 16-9-2022	
VDI WSOLDSW VI VE	
Q.1. Ax= 0x334A, Bx = 0x45F1	
$C_{x} = 0 \times 8934$	
a) add any bx	
0011 0011 0100 1010	
0100 0101 1111 0001	
0111 1001 0011 1011	
CF = 0, $OF = 0$, $SF = 0$, $ZF = 0$	
b) add ex, bx	2 M 5
1000 1001 0011 0100	
0100 0101 1111 0001	
1100 1111 0010 0101	

	CF = O $OF = O$ $SF = 100$, $ZF = O$	
	c) sub bx, 6	
Timesel Colonia Coloni	6 = 0000 0000 0000 0110	
tonesco	-6 = 1111 111 111 1010	
5	0100 0101 1111 0001	
	1111 1111 1010	
	1 0100 0101 1110 1011	-
Почения выполнения почения в почения	CF = 0, $OF = 0$, $SF = 0$, $ZF = 0$	
	Q.2: Storing words in Big Enclian Form	t:
Maker appetite the control of the Consular Method has the Consular	a) 0x 6900	
	In Lower Memory Address: 69	
7	In Higher Memory Address: 00	
	b) 0x4567	and the state of t
	In Lower Memory Address: 45	and an all the second and the second
	In Lower Memory Address: 45 In Higher Memory Address: 67 c) OXAA99	
	c) OXAA99	
	In Lower Memory Address: AA	
	In Lower Memory Address: AA In Higher Memory Address: 99	
		1000
		and the second
		Pales



```
[org 0x0100]
 1
 2
 3
  mov ax, 45
   mov bx, -7
 4
 5
   mov cx, 0
   mov word[result], 0
 6
 7
8 cmp ax, 0
                              ; if (ax < 0)
 9 jns skip2
                              ; {
10 neg ax
                                  ax = -ax
11 inc cx
                                 CX++
12 skip2:
                               }
13
                              ; if (bx < 0)
14 \text{ cmp bx}, 0
15 jns skip3
                               {
16 neg bx
                                  bx = -bx
17 inc cx
                                cx++
18 skip3:
                               }
19
20 cmp ax, bx
                              ; if (ax < bx)
21 jg skip1
                               {
22 mov dx, ax
                                temp = ax
23 mov ax, bx
                                  ax = bx
```

```
24 mov bx, dx
                                bx = temp
25 skip1:
26
27 loopStart:
28 cmp bx, 0
                                if(bx == 0)
29 je loopEnd
                                    break;
30
31 add word[result], ax
                            ; result = result + ax
32
33 sub bx, 1
                                bx--
34
35 jmp loopStart
36 loopEnd:
37
38 cmp cx, 1
                            ; if (cx == 1)
39 jne skip4
                            ; result = -result
40 neg word[result]
41 skip4:
42
43 mov ax, 0x4C00
44 int 0x21
```

The second second second second	Q.5: Calculate Effective Address:	en meneral aggi gar all angele di meneral aggi en meneral aggi en meneral aggi gar angele aggi en meneral aggi Permanyan
au Citrick Providence	Bx = 0x0034, Si = 0x0110, Di = 0x1101	j
	BP = 0x0220, SP = 0x FFFF	grammers in a graph of the first him is a grammer of processing of the contract of the contrac
	a) ho-di (XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	gazinas joju arabani del mogli oprilitika kar pozicijajani je e umas. Kiroma
	a) bp-di (Subtraction Thyallid It walls Invalid (is not allowed) thyallid It walls	11646x)
	b) bp + si d) bx + bp	
Control Marie Control		gazaran Sprigerenten den dia Parlista mendelah persentan dia persentan dan persentan dan dia persentan dan dia Persentan
MICE AND SHAPE		
отположения в при	UVO Base reg-	and or operating the property of the control of the
	Effective: 0 x 0 3 3 0 - isters cannot Address be added	amendar yan madi anti inti data an mada dataka ka ka mada na mada ka
Conference of the 100	De douce	व व्य वनार १९४० १५४० १५४४ म्या स्थापन स्यापन स्थापन स्था
	e) bx +ip f) bx +di	n Britis (Michigae Britis (Alberta)
The second second	Invalid Because 0x0034	
and page 17 beautiful	ip register cannot be 0x1101	According to the second control of the secon
manustra and	accessed. 0x0034 0X1135	n-class en years substitutes son version at a medical a version at the contract of the contra
on the second of the second	c) bx-0x12-0x0012 Effective: 0x1135 Effective 0x0022 Address	
The state of the same	Haavess . Numer	
	Q:6 CS: 0x4582 SS: 0x4582	Money
-	Bx:0x22AA IP:0x0580 DI:0x4247	Principal Control of C
	BP: 0x4700 SI: 0x FFFF	en massatti grasili in galetteria consumento de en entretanno sugar Protes
- Company of the Comp	a) [cs: bx+si]	The section of the se
	bx: 0 x 22 A A.	
	Si: OX FEEF	The Second Contract of
	0×12199	Man
	Offset avalor com t	and the second s
	Offset 0x 2199 (segment wraparound)
Margin Stages and Address of the Stage of th		
· Immediate in the second		

	CS: 0 x 45820
	bx + si 0 x 0 2 1 9 9
	0 x 479 B 9
	Physical Allen ox 47989
The state of the s	Physical Address: 0x 47989 Wrap Around: Segment. Wrap Around
	Trap Hounds segment
	b) [1
The second secon	b) [bp+di+10]
	bp 0 x 4700
	di 0 x 4247
Millianic or recognitional and recoding to the second second	10 0 x 000A
Decrees on the Control of the Contro	Offset: 0 x 8 951
NORTH AND AND PROPERTY OF THE	SS: 0×45820
NAMES AND ADDRESS OF THE STATE	bp+di+10 0x 08951
	0x 4 E171
pagewormhalms.sunmacranisapormand vasioning and com-	Physical Address: 0x4E171 Wrap around: No Wrap around:
	Wrap around: No Wrap around:
	AB
Million control particle and administration and the control of the desired	Q.7: a) mov ip, bx
SERVICE TO THE SERVICE STATE OF THE SERVICE STATE STATE OF THE SERVICE STATE OF THE SERVICE STATE OF THE SERVICE S	ip register cannot be madified.
the county is the facilities when the common description when it can	mov bp, bx
The unbook sign consensus and a large sign of the sign	mor any bx
	b) mov byte bx, [ip]
	ip register cannot accessed and
	ip register cannot accessed and byte size cannot be mentioned with by register
	bx register
Constitution of the Consti	

	Alternate Solutions:	
	mor 6l, [bp]	
and the second of the second o	mov bz, [si]	
	c) mov si, al	·
	Si is a 16 bits register while al	
	is a 8 bits register creating size	
	mismatch error.	
	Alternate Solutions:	
	mer sig bx	
	mov Sig ax	
		The state of the s
and the second s	d) mov ax, [bx+bp+100]	
	Invalid because two base	
	registers cannot be added in effective	
	address.	
	Alternate Solutions:	
	mov an, [bx+si+100]	
	mor any [bp+di+100]	-
	Q.8) Flags After running the Instructions.	
	instructions	
	Zero = O Courry Flag = O Parity Flag = O	
	Flag	
	Overflow Plag=0 Sign Plag=1	
	July July July July July July July July	

There is not any apparent logical error in code of question no. 8 except that the result stored in num+4 address is overwritten in case the sum 18 greater than 4 so there should be an entra register or memory used to store the sum of first four? numbers of number.

```
[org 0x0100]
2
   ; write your code here to swap the contents of even index with the odd ones
   ; that is, swap 1 & amp; 2, 3 & amp; 4
   start:
   mov si, 0 		; si = 0
   mov di, 1 		 ; di = 1
8
   swapStart: ; {
   cmp si, [siz] ; if (si >= siz)
10
11
                            break
   jge swapEnd
12
13
   mov al, [num1 + si]; al = num1[si]
   mov bl, [num1 + di]; bl = num1[di]
14
15
16
  mov [num1 + si], bl; num1[si] = bl
  mov [num1 + di], al; num1[di] = al
17
18
19 add si, 2 ; si = si + 2
   add di, 2; di = di + 2
20
21
22
   jmp swapStart
23
   swapEnd:
                     ; }
24
25
   end:
26
   mov ax, 0x4c00 ; terminate program
27
   int 0x21
28
  num1: db 1, 2, 3, 4
29
30
   siz: dw 4
```



```
[org 0x0100]
 1
 2
 3
    ; write your code here to find max in an array
    ; and store it min
 4
 5
    start:
   mov si, 2
 6
                             ; si = 2
   mov ax, [array1]
 7
                             ; max = array1[0]
                             cx = 1
 8
   mov cx, 1
 9
10
   loopStart:
                               {
                             ; if (cx >= siz)
11 cmp cx, [siz]
    jge loopEnd
12
                                    break
13
14
    cmp [array1 + si], ax
                             ; if (arrayl[si] > max)
15
    jle skip1
                             ; {
   mov ax, [arrayl + si]
16
                                max = arrayl[si]
17
    skip1:
                             ; }
18
    add si, 2
19
                                 si = si + 2
20
    add cx, 1
                              cx = cx + 1
21
22
    jmp loopStart
23
    loopEnd:
                             ; }
24
25
   mov [min], ax
26
27
  end:
   mov ax, 0x4c00
28
                             ; terminate program
29
    int 0x21
30
   array1: dw 1, 3, -8, 7, 5
31
    siz: dw 5
32
33
   min: dw 0
```