CS221 Final 472

Q1- True or False: [5 marks]

1) Accessing a register is faster than memory.	
2) Assembly language uses symbolic names rather than binary code.	
3) Assembly is a high-level language.	
4) Assembly language is more portable than high-level languages.	
5) Assembly is used for system software and embedded systems.	

Q2- Fill in the blanks: [5 marks]

1) …….. instruction is used to copy data from a source operand to a destination operand.
2) …….. instruction is used to exchange data between two operands.
3) …….. instruction is used to insert a new element on top of the stack.
4) …….. instruction is used to delete the top element from the stack.
5) …….. instruction is used to copy a smaller unsigned source operand into a larger destination register by filling the upper part with zeros.

Q3- If array values are 10,8,6,5,4,12,8,20. What is the output after executing the following program? (EBX=10) [5 marks]

Answer	The output is …….

Program	count PROC uses esi ecx ebx
	
	mov eax, 0
	L1:
	cmp [esi], ebx
	jae next
	inc eax
	next:
	add esi, 4
	loop L1
	
	ret ; was missing in the exam
	count ENDP

Q4- Give the values of registers after executing the instructions. [6 marks]

```
.data
arrayB BYTE 10h, 20h, 30h, 40h
arrayW WORD 100h, 200h, 300h, 400h
arrayD DWORD 10000h, 20000h, 30000h, 40000h

.code
mov esi , 2

mov al, arrayB[esi]

mov ax, arrayW[esi + 4]

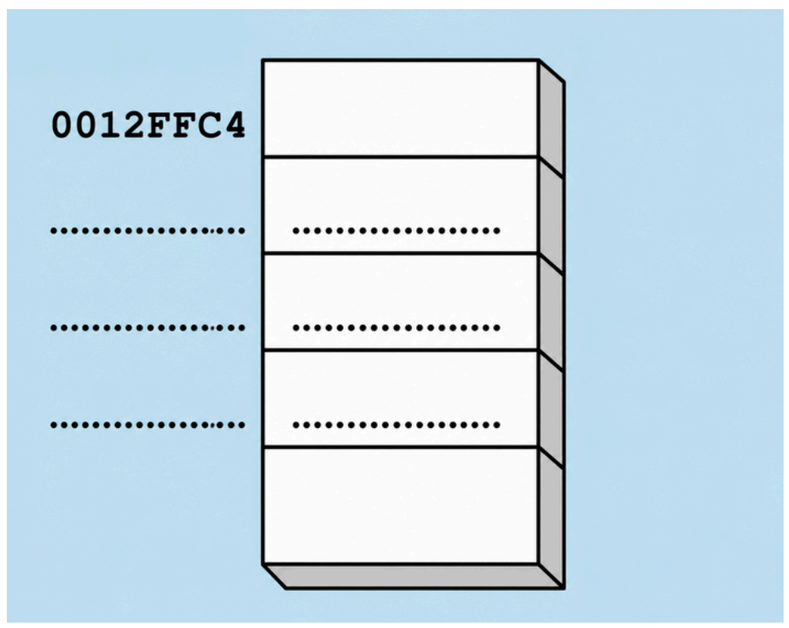
mov eax, arrayD[esi * 2]
```

Register	Value
al	
ax	
eax	

Q5- Give the values of ESP and fill the stack. [6 marks]

Suppose we execute the following:

```
PUSH EAX  ; EAX = 125C80FFh
PUSH EBX  ; EBX = 2Eh
PUSH ECX  ; ECX = 9B61Dh
```



ESP= .....

Q6- Fill in the blanks with the correct displacement values to access the fields of the structure. [4 marks]

```
.DATA
  mystruct WORD    12
           WORD    10
           DWORD   1981
           BYTE    'M'
.CODE
  mov ebx, OFFSET mystruct
  mov ax,   [ebx + .... ]    ; AX = 10
  mov eax, [ebx + .... ]     ; EAX = 1981
  mov al,   [ebx + .... ]    ; AL = 'M'
```

Q7- Write the following C++ code in Assembly language. [5 marks]

C++ Code	Assembly code

while (ebx <= var1)
{
ebx = ebx + 5;
var1 = var1 - 1
}	

Q8- Write an assembly code satisfty the following. [14 marks]

Write a program that prompts the user for five 32-bit unsigned integers, stores them in an array, calculates only the sum of the ODD values of the array, displays the sum on the screen.
Your code must be composed with the following procedures:

- **Main procedure**
- **Prompt user for multiple integers**
- **Calculate the sum of the ODD values and display it.**

```
INCLUDE Irvine32.inc
ArraySize EQU 5

.DATA
prompt1 BYTE "Enter an unsigned integer: ",0
prompt2 BYTE "The sum of the odd integers is: ",0

array DWORD ArraySize DUP(?)

.CODE
main PROC
call Clrscr        ; clear the screen
.....
.....
.....
.....
.....
exit
main ENDP
```

```
;-----------------------------------------------
; PromptForIntegers: Read input integers from the user
; Receives:
; ESI = pointer to the array
; ECX = array size
; Returns: Fills the array with the user input
;-----------------------------------------------
PromptForIntegers PROC USES .....
.....
.....
.....
.....
.....
.....
.....
.....
.....
PromptForIntegers ENDP
;-----------------------------------------------
; ArraySum: Calculates the sum of the Odd integers and display it
; Receives:
; ESI = pointer to the array
; ECX = array size
; Returns: Displays the sum after calculating it
;-----------------------------------------------
ArraySum PROC USES .....
.....
.....
.....
.....
.....
.....
.....
.....
.....
ArraySum ENDP
END main
```

Answers are on the next page

Answers

Question 1&2 answer:

Question 1		Question 2	
1	T	**1**	MOV
2	T	**2**	XCHG
3	F	**3**	PUSH
4	F	**4**	POP
5	T	**5**	MOVZX

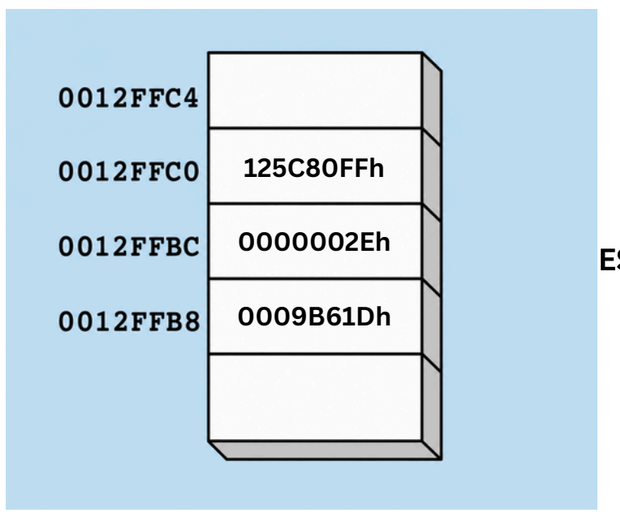
Question 3 answer:

Answer	**The output is: EAX = 5**

Question 4 answer:

Register	Value
al	30h
ax	400h
eax	20000h

Question 5 answer:



ESP= 0012FFB8

Question 6 answer:

Answer
[ebx + 2]
[ebx + 4]
[ebx + 8]

Question 7 answer:

C++ Code	Assembly code

while (ebx <= var1)
{
ebx = ebx + 5;
var1 = var1 - 1
}
	top:
	cmp ebx, var1
	ja next
	add ebx, 5
	dec var1
	jmp top
	next:

Question 8 answer:

MAIN Procedure

```
main PROC

call Clrscr
; clear the screen

mov esi, offset array
mov ecx, ArraySize

call PromptForIntegers
call ArraySum

exit
main ENDP
```

PromptForIntegers Procedure

```
PromptForIntegers PROC USES esi ecx

    mov edx, offset prompt1

l1:
    call WriteString
    call ReadDec

    mov [esi], eax
    add esi, 4
    loop l1

    ret
PromptForIntegers ENDP
```

ArraySum Procedure

```
ArraySum PROC USES esi ecx

    mov eax, 0
    mov edx, offset prompt2
l2:
    test DWORD PTR [esi], 1
    jz skip
    add eax, [esi]

skip:
    add esi, 4
    loop l2

    call WriteString
    call WriteDec
    call Crlf

    ret
ArraySum ENDP
```

Made By Omar Edilbi