

# Comp 3350: Computer Organization & Assembly Language

## HW # 10: Theme: Strings and Arrays

(All main questions carry equal weight. Credit awarded to only those answers for which work has been shown.)

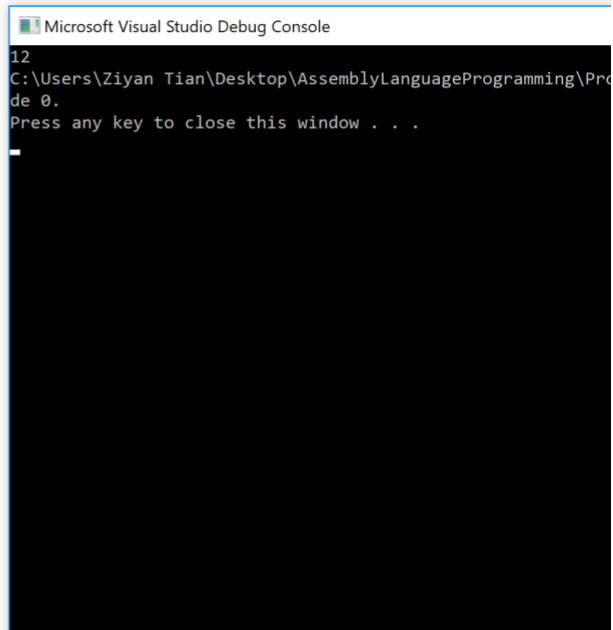
1. [Interrupts] What are hardware and software interrupts? Give examples of each. What are maskable and non-maskable interrupts? Provide examples of each. 6 points \* 4
  - Hardware interrupts  
A hardware interrupt is generated PIC, which signals the CPU to suspend execution of the current program and execute an interrupt service routine.  
Example, a keyboard character waiting at the input port would be lost if not saved by the CPU;
  - Software interrupts  
A software interrupt is a call to an operating system procedure.  
Example, INT <xx>, TRAP
  - Maskable interrupts  
A maskable interrupt is an interrupt that can be disabled by using standard interrupt-masking techniques  
Example, INT, INTR
  - Non-maskable interrupts  
A non-maskable interrupt is a hardware interrupt that standard interrupt-masking techniques in the system cannot ignore.  
Example, NMI, Reset
2. [Strings] Write a program that computes the number of characters in any string. Test the robustness of your program using different strings including those of size 0.

```
.data
str1    BYTE    "qwerfcxdftyu",0
.code
main PROC
    mov ecx, 0        ; counter
    mov eax, OFFSET str1
L1: mov ebx, [eax]
    cmp ebx, 0
    jz L2
    add ecx, 1
    add eax, 1
    jmp L1

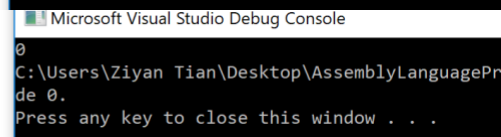
L2: mov eax, ecx
    call WriteDec

    invoke ExitProcess, 0
main endp
end main
```

```
.data
str1    BYTE    0
.code
main PROC
```



```
Microsoft Visual Studio Debug Console
12
C:\Users\Ziyan Tian\Desktop\AssemblyLanguageProgramming\Pro
de 0.
Press any key to close this window . . .
```



```
Microsoft Visual Studio Debug Console
0
C:\Users\Ziyan Tian\Desktop\AssemblyLanguageProgramming\Pro
de 0.
Press any key to close this window . . .
```

3. [Structures] Using the structure example discussed in the book and slides, write a program that displays the  $x$ -coordinates of several points given as an array of coordinates in the data segment. Unlike the example in the book you should use 3-dimensional points. Test your program with various  $(x, y, z)$  inputs. Use base-indexed addressing to implement the program.

```

COOR STRUCT
    x WORD ?
    y WORD ?
    z WORD ?
COOR ENDS

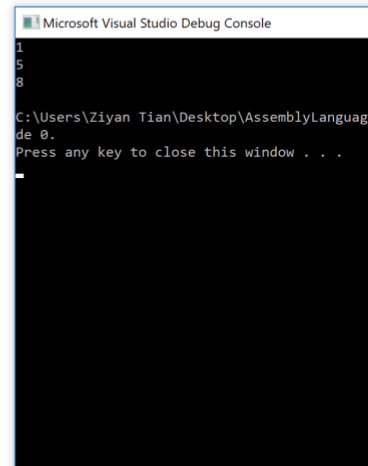
.data
points COOR <1, 2, 4>, <5, 6, 7>, <8, 9, 0>

.code
main PROC
    mov esi, OFFSET points
    mov edi, 0
    mov eax, 0
    mov ebx, 6
    mov ecx, 3

L1: mov ax, WORD PTR [esi + edi]
    call WriteDec
    call crlf
    add edi, ebx
    loop L1

    invoke ExitProcess, 0
main endp
end main

```



4. [Strings] Write a program that searches for a sequence of two characters in a string, e.g in the string “Folks, today is a full moon day in a full dark night” you can search for the first occurrence of “fu”. You should set the EDI pointer to point to the first character found. Test the program thoroughly using various strings. Provide screen shots of the runs along with your program. You must use string instructions in your program.

```

.data
str1 BYTE "Folks, today is a full moon day in a full dark night", 0

.code
main PROC
    mov esi, OFFSET str1
    mov edi, 0
    mov ecx, LENGTHOF str1
    sub ecx, 1

L1: mov eax, DWORD PTR [esi + edi]
    cmp al, 66h
    jnz L2
    mov eax, DWORD PTR [esi + edi + 1]
    cmp al, 75h
    jnz L2
    add edi, esi
    mov eax, edi
    call WriteHex
    call crlf
    mov eax, [edi]
    call WriteChar
    mov eax, [edi + 1]
    call WriteChar
    jmp L3
L2: add edi, 1
    loop L1

L3: invoke ExitProcess, 0
main endp
end main

```

