AuBuchon Lab-14 (50 points)

Write assembly program using system calls that implement the following tasks:

1. Define and declare a structure Point that has x and y coordinates
2. Define an array of 5 structures of Point type
3. Use loop 1 to ask a user to input x and y values for the 5 structures using system calls
4. Use a loop 2 to print out the x and y values for the array of structure using the system calls

;array of structures

struc Point

.x**:** resd 1

.y**:** resd 1

.size**:**

endstruc

section .data

msg1**:** db"Enter the x and y values for the five points"**,**10**,**0

msgL1**:** equ **$-**msg1

msg2**:** db"Printing the point coordinates"**,**10**,**0

msgL2**:** equ **$-**msg2

P**:**ISTRUC Point

AT Point.x**,** dd 0

AT Point.y**,** dd 0

IEND

section **.bss**

counter resd 1

PtArr**:** RESB Point.size**\***5

ArrCount**:** equ**($-**PtArr**)/**Point.size

section **.text**

global main

main**:**

**push** **ebp**

**mov** **ebp,** **esp**

**mov** **ecx,** ArrCount

**mov** **esi,** PtArr

L1**:**

**mov** **ecx,** msg1

**mov** **edx,** msgL1

**call** printString

;create system call for input for array

**lea** **ecx,** **[esi** **+** Point.x**]**

**call** standardInput

;change ascii string into decimal value

**mov** **eax,** **[esi** **+** Point.x**]**

**sub** **eax,** '0'

**mov** **DWORD[esi** **+** Point.x**],** **eax**

**mov** **ecx,** msg1

**mov** **edx,** msgL1

**call** printString

;create system call for input for array

**lea** **ecx,** **[esi** **+** Point.y**]**

**call** standardInput

;change ascii string into decimal value

**mov** **eax,** **[esi** **+** Point.y**]**

**sub** **eax,** '0'

**mov** **DWORD[esi** **+** Point.y**],** **eax**

**add** **esi,** Point.size

;increase counter for loop 5 times

**inc** **dword[**counter**]**

**cmp** **dword[**counter**],** 5

**jne** L1

**mov** **ecx,** msg2

**mov** **edx,** msgL2

**call** printString

**mov** **ecx,** ArrCount ;ecx has the count of array elements

**mov** **esi,** PtArr ;esi has the address for the first structure in the array

L2**:**

**mov** **eax,[esi** **+** Point.size**]**

**call** PrintDec

**call** println

**mov** **eax,[esi** **+** Point.y**]**

**call** PrintDec

**call** println

**add** **esi,** Point.size ;move to next structure in array

**loop** L2

**mov** **esp,** **ebp**

**pop** **ebp**

**ret**

standardInput**:**

**pusha**

**mov** **eax,** 3

**mov** **ebx,** 0

**mov** **edx,** 1

**int** 80h

**popa**

**ret**

PrintDec**:**

section **.bss**

decstr resb 10

ct1 resd 1 ;keep track of string size

section **.text**

**pusha**

**mov** **dword[**ct1**],**0 ;assume initially 0

**mov** **edi,**decstr ;edi points to dec-string in memory

**add** **edi,**9 ;mov the last element of string

**xor** **edx,** **edx** ;clear out edx for 64 bit division

whileNotZero**:**

**mov** **ebx,** 10 ;store 10 for division

**div** **ebx** ;divide by 10

**add** **edx,**'0' ;convert to ascii char

**mov** **byte[edi],dl** ;move to string

**dec** **edi** ; mov to next char in string

**inc** **dword[**ct1**]** ;increment char counter

**xor** **edx,** **edx** ;clear edx

**cmp** **eax,** 0 ;is remainder 0

**jne** whileNotZero ; if not keep looping

**inc** **edi** ;conversion, finish, bring edi

**mov** **ecx,** **edi** ;back to beginning of string, make ecx

**mov** **edx,** **[**ct1**]** ;point to it, and edx gets # chars

**mov** **eax,** 4 ;and print! to the stndout

**mov** **ebx,** 1

**int** 80h

**popa**

**ret**

printString**:**

;save register values of the called function

**pusha**

;string is pointed by ecx, edx has it's length

**mov** **eax,** 4

**mov** **ebx,** 1

**int** 80h

;return old register values of the called function

**popa**

**ret**

println**:**

section .data

n1 db 10

section **.text**

**pusha**

**mov** **ecx,** n1

**mov** **edx,** 1

**call** printString

**popa**

**ret**