COAL LAB MIDTERM FALL 2020

Ibadullah Shaikh

19k-0259

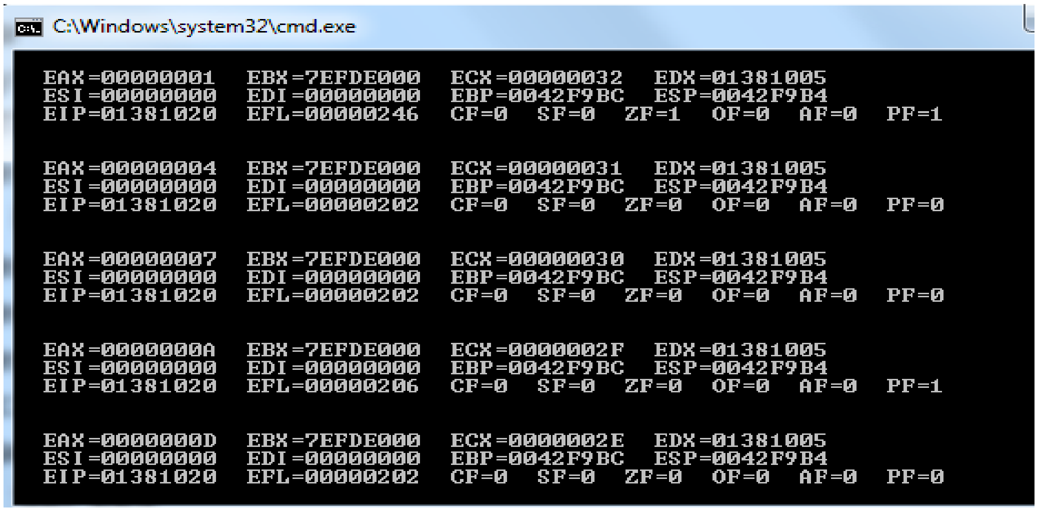
Section: G

Task 1 (a):

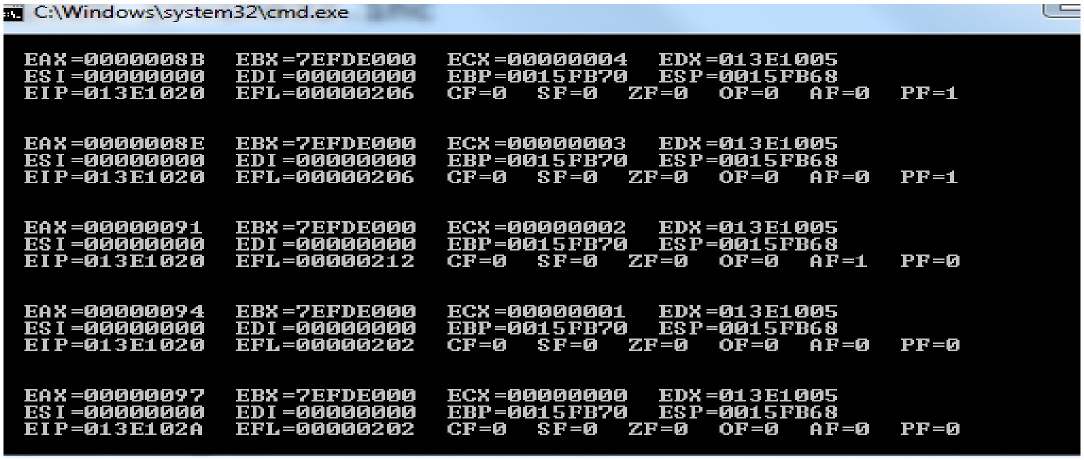
Code:

```
INCLUDE Irvine32.inc
.data
array BYTE 50 DUP (?)
n BYTE 50
.code
main PROC
mov eax, 1
mov ecx, DWORD PTR n
Series:
       call DumpRegs
       add eax, 3
Loop Series
call DumpRegs
exit
main ENDP
END main
```

Screenshot:



```
  EAX=00000001  EBX=7EFDE000  ECX=00000032  EDX=01381005
  ESI=00000000  EDI=00000000  EBP=0042F9BC  ESP=0042F9B4
  EIP=01381020  EFL=00000246  CF=0  SF=0  ZF=1  OF=0  AF=0  PF=1


  EAX=00000004  EBX=7EFDE000  ECX=00000031  EDX=01381005
  ESI=00000000  EDI=00000000  EBP=0042F9BC  ESP=0042F9B4
  EIP=01381020  EFL=00000202  CF=0  SF=0  ZF=0  OF=0  AF=0  PF=0


  EAX=00000007  EBX=7EFDE000  ECX=00000030  EDX=01381005
  ESI=00000000  EDI=00000000  EBP=0042F9BC  ESP=0042F9B4
  EIP=01381020  EFL=00000202  CF=0  SF=0  ZF=0  OF=0  AF=0  PF=0


  EAX=0000000A  EBX=7EFDE000  ECX=0000002F  EDX=01381005
  ESI=00000000  EDI=00000000  EBP=0042F9BC  ESP=0042F9B4
  EIP=01381020  EFL=00000206  CF=0  SF=0  ZF=0  OF=0  AF=0  PF=1


  EAX=0000000D  EBX=7EFDE000  ECX=0000002E  EDX=01381005
  ESI=00000000  EDI=00000000  EBP=0042F9BC  ESP=0042F9B4
  EIP=01381020  EFL=00000202  CF=0  SF=0  ZF=0  OF=0  AF=0  PF=0
```



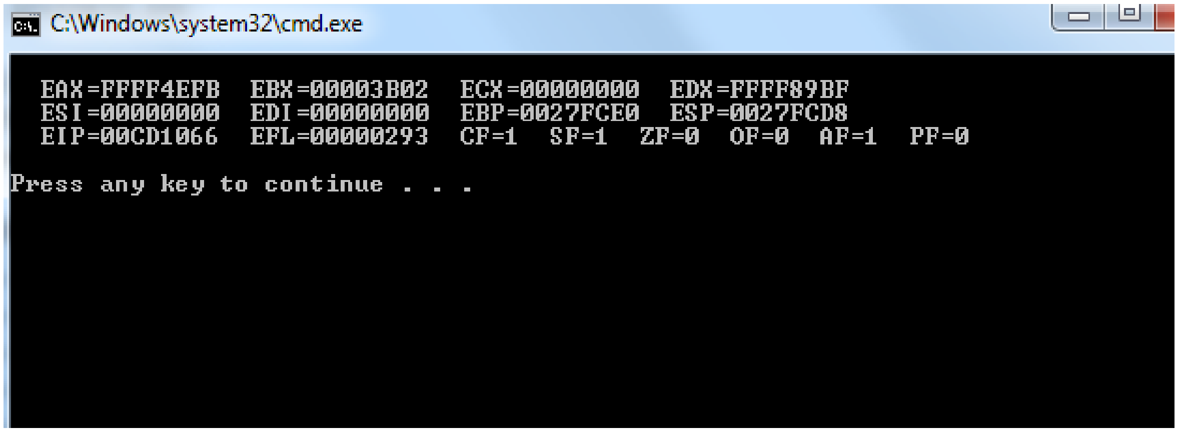
```
  EAX=0000008B  EBX=7EFDE000  ECX=00000004  EDX=013E1005
  ESI=00000000  EDI=00000000  EBP=0015FB70  ESP=0015FB68
  EIP=013E1020  EFL=00000206  CF=0  SF=0  ZF=0  OF=0  AF=0  PF=1

  EAX=0000008E  EBX=7EFDE000  ECX=00000003  EDX=013E1005
  ESI=00000000  EDI=00000000  EBP=0015FB70  ESP=0015FB68
  EIP=013E1020  EFL=00000206  CF=0  SF=0  ZF=0  OF=0  AF=0  PF=1

  EAX=00000091  EBX=7EFDE000  ECX=00000002  EDX=013E1005
  ESI=00000000  EDI=00000000  EBP=0015FB70  ESP=0015FB68
  EIP=013E1020  EFL=00000212  CF=0  SF=0  ZF=0  OF=0  AF=1  PF=0

  EAX=00000094  EBX=7EFDE000  ECX=00000001  EDX=013E1005
  ESI=00000000  EDI=00000000  EBP=0015FB70  ESP=0015FB68
  EIP=013E1020  EFL=00000202  CF=0  SF=0  ZF=0  OF=0  AF=0  PF=0

  EAX=00000097  EBX=7EFDE000  ECX=00000000  EDX=013E1005
  ESI=00000000  EDI=00000000  EBP=0015FB70  ESP=0015FB68
  EIP=013E102A  EFL=00000202  CF=0  SF=0  ZF=0  OF=0  AF=0  PF=0
```

(b):

Code:

```
INCLUDE Irvine32.inc
.data
x BYTE 2
y BYTE 59
result_ans DWORD ?
.code
main PROC
mov eax, 0
mov ebx, 0
mov eax, DWORD PTR y
mov ebx, DWORD PTR x
add eax, 5
add ebx, 2
sub eax, ebx
PUSH eax
mov eax, DWORD PTR y
mov ebx, DWORD PTR x
sub eax, ebx
mov edx, eax
mov eax, DWORD PTR y
mov ebx, DWORD PTR x
add eax, ebx
add y, 2
sub edx, eax
sub edx, DWORD PTR y
POP eax
add eax, edx
mov result_ans, eax
call DumpRegs
exit
main ENDP
END main
```

Screenshot:



```
  EAX=FFFF4EFB  EBX=00003B02  ECX=00000000  EDX=FFFF89BF
  ESI=00000000  EDI=00000000  EBP=0027FCE0  ESP=0027FCD8
  EIP=00CD1066  EFL=00000293  CF=1  SF=1  ZF=0  OF=0  AF=1  PF=0

Press any key to continue . . .
```

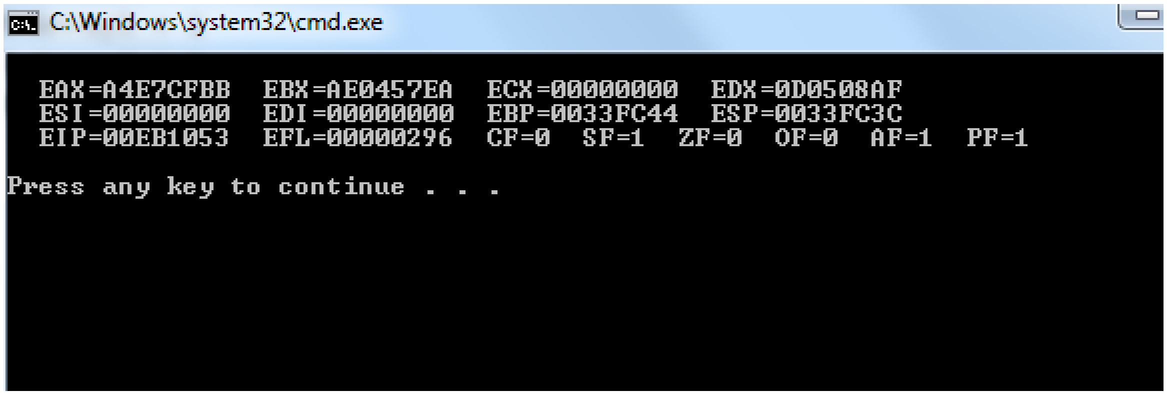
Task 2: (a)

Code:

```
INCLUDE Irvine32.inc
.data
Array1 BYTE 11, 22, 33
Array2 BYTE 111, 222, 233
Array3 WORD 1111, 2222, 3333
Res1 DWORD ?, ?, ?
.code
main PROC
mov eax, 0
mov ebx, 0
mov edx, 0
call AddArrays_0
call AddArrays_1
call AddArrays_2
call DumpRegs
exit
main ENDP
AddArrays_0 PROC
       mov eax, DWORD PTR Array1 + 0
       mov ebx, DWORD PTR Array2 + 0
       mov edx, DWORD PTR Array3 + 0
       inc eax
       inc ebx
       inc edx
       add eax, ebx
       add eax, edx
       mov res1+0, eax
       RET
AddArrays_0 ENDP

AddArrays_1 PROC
       mov eax, DWORD PTR Array1 + 1
       mov ebx, DWORD PTR Array2 + 1
```

```
        mov edx, DWORD PTR Array3 + 1
        inc eax
        inc ebx
        inc edx
        add eax, ebx
        add eax, edx
        mov res1+1, eax
        RET
AddArrays_1 ENDP

AddArrays_2 PROC
        mov eax, DWORD PTR Array1 + 2
        mov ebx, DWORD PTR Array2 + 2
        mov edx, DWORD PTR Array3 + 2
        inc eax
        inc ebx
        inc edx
        add eax, ebx
        add eax, edx
        mov res1+2, eax
        RET
AddArrays_2 ENDP

END main
```

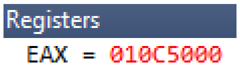
Screenshot:



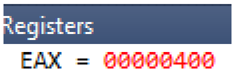
```
  EAX=A4E7CFBB  EBX=AE0457EA  ECX=00000000  EDX=0D0508AF
  ESI=00000000  EDI=00000000  EBP=0033FC44  ESP=0033FC3C
  EIP=00EB1053  EFL=00000296  CF=0  SF=1  ZF=0  OF=0  AF=1  PF=1

Press any key to continue . . .
```

Task 3:

a)

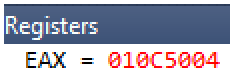


```
EAX = 010C5000
```

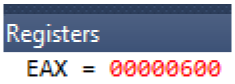


```
EAX = 00000400
```

b)

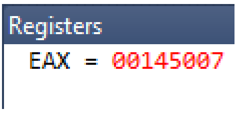


```
EAX = 010C5004
```

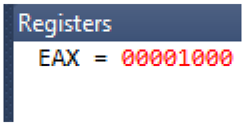


```
EAX = 00000600
```

c)

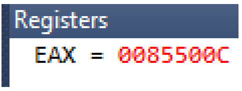


```
EAX = 00145007
```

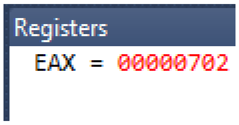


```
EAX = 00001000
```

d)

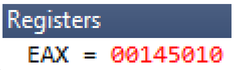


```
EAX = 0085500C
```

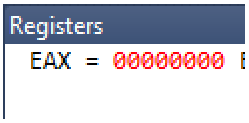


```
EAX = 00000702
```

e)



```
EAX = 00145010
```



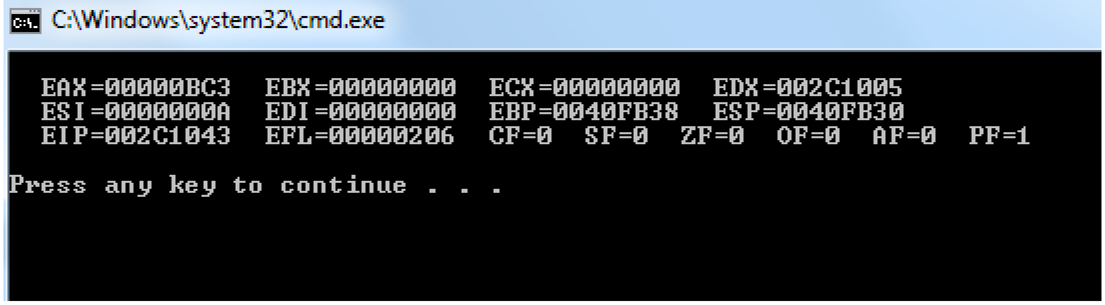
```
EAX = 00000000 E
```

Task 4:

Code:

```
INCLUDE Irvine32.inc
.data
arrayW DW 50,20,90,101,450
arrayB DB 10, 24,67,90,100
arraySum WORD ?,?,?,?,?
.code
main PROC
mov eax, 0
mov ebx, 0
mov esi, 0
mov ecx, LENGTHOF arrayW
Looopp:
       mov ax, arrayW[esi]
       add ax, WORD PTR arrayB[esi+1]
       mov arraySum[esi],ax
       add esi,2
Loop Looopp
call DumpRegs
exit
main ENDP
END main
```

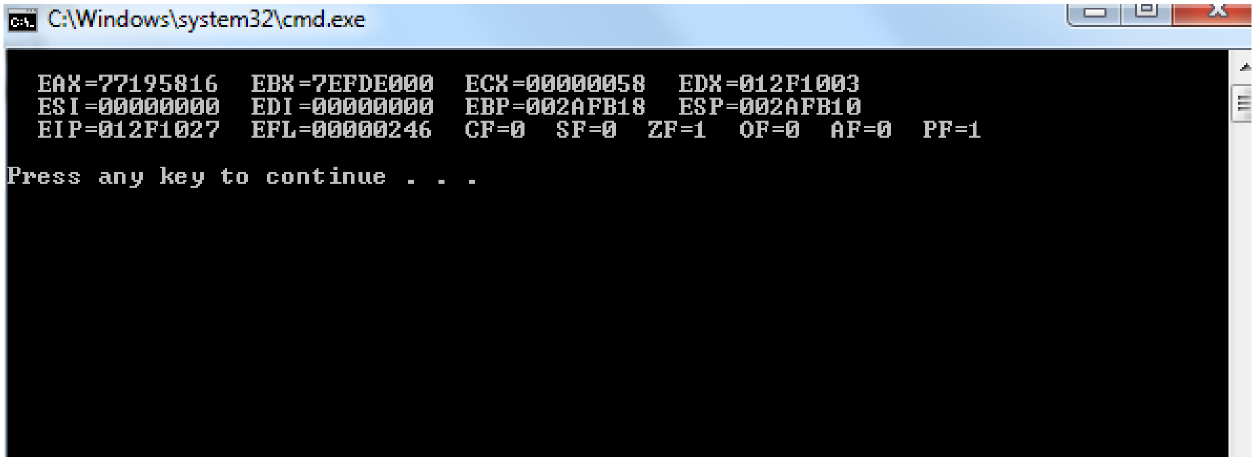
ScreenShot:



```
  EAX=00000BC3  EBX=00000000  ECX=00000000  EDX=002C1005
  ESI=0000000A  EDI=00000000  EBP=0040FB38  ESP=0040FB30
  EIP=002C1043  EFL=00000206  CF=0  SF=0  ZF=0  OF=0  AF=0  PF=1

Press any key to continue . . .
```

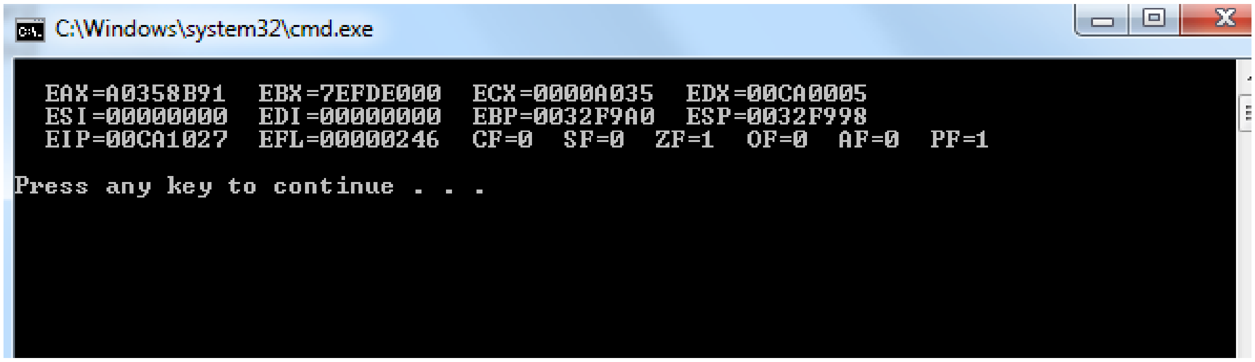
Task 5:

Screenshot (1):



```
  EAX=77195816  EBX=7EFDE000  ECX=00000058  EDX=012F1003
  ESI=00000000  EDI=00000000  EBP=002AFB18  ESP=002AFB10
  EIP=012F1027  EFL=00000246  CF=0  SF=0  ZF=1  OF=0  AF=0  PF=1

Press any key to continue . . .
```

Screenshot (2):



```
  EAX=A0358B91  EBX=7EFDE000  ECX=0000A035  EDX=00CA0005
  ESI=00000000  EDI=00000000  EBP=0032F9A0  ESP=0032F998
  EIP=00CA1027  EFL=00000246  CF=0  SF=0  ZF=1  OF=0  AF=0  PF=1

Press any key to continue . . .
```

Task 6:

Code:

```
INCLUDE Irvine32.inc
.data
array1 BYTE ?,?,?,?
array2 BYTE ?,?,?,?
array3 BYTE ?,?,?,?
array4 BYTE ?,?,?,?
.code
main PROC
mov ecx, 4
mov edi, OFFSET array2
mov eax, TYPE array 4
Outer:
       mov esi, OFFSET array1
       add eax, 4
       PUSH ecx
       mov ecx, 4
       Inner:
              mov ebx, eax
              add ebx, [esi]
              mov [edi], ebx
              add edi, 4
              add esi, 4
       Loop Inner
       POP ecx
Loop Outer
call DumpRegs
exit
main ENDP
END main
```

Screenshot: