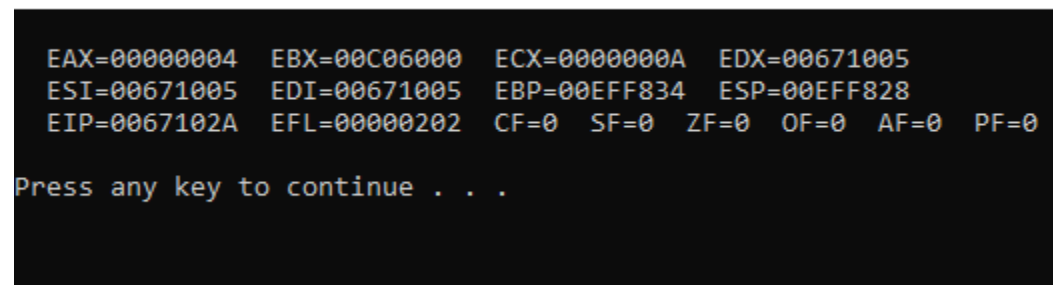# COAL LAB TASKS 7

Task 1:

Code:

```
INCLUDE Irvine32.inc
.data
table DWORD 4
.code
main PROC
mov eax, table
mov ecx, 10
PUSHAD
One:
      mov ebp, esp
      mov eax, [ebp + 28]
      mul ecx
Loop One
POPAD
call DumpRegs
exit
main ENDP
END main
```

Screenshot:

C:\WINDOWS\system32\cmd.exe

```
  EAX=00000004  EBX=00C06000  ECX=0000000A  EDX=00671005
  ESI=00671005  EDI=00671005  EBP=00EFF834  ESP=00EFF828
  EIP=0067102A  EFL=00000202  CF=0  SF=0  ZF=0  OF=0  AF=0  PF=0

Press any key to continue . . .
```
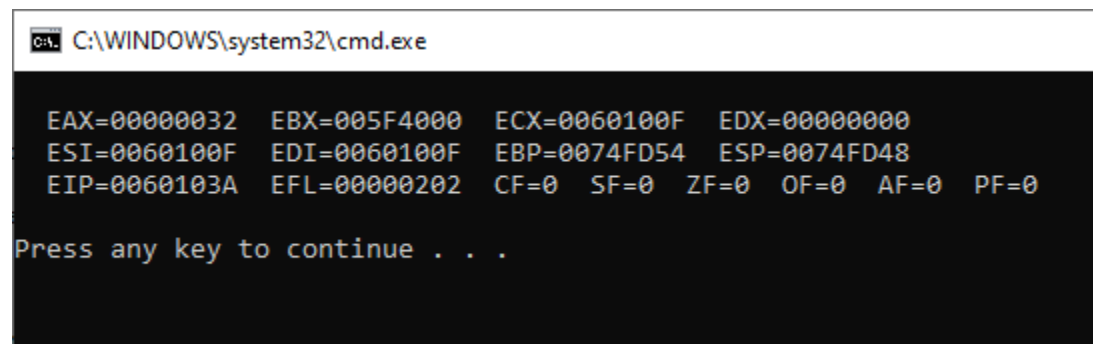
Task 2:

Code:

```
INCLUDE Irvine32.inc
.data
val1 DWORD 10
val2 DWORD 5
addi DWORD ?
subt DWORD ?
mult DWORD ?
.code
main PROC
mov eax, 0
mov ebx, 0
call AddTwo
call DumpRegs
exit
main ENDP
AddTwo PROC
       mov eax, val1
       add eax, val2
       mov addi, eax
       call SubTwo
       ret
AddTwo ENDP
SubTwo PROC
       mov eax, val1
       sub eax, val2
       mov subt, eax
       call MulTwo
       ret
SubTwo ENDP
MulTwo PROC
       mov eax, val1
       Mul val2
       mov mult, eax
       ret
MulTwo ENDP
END main
```

Screenshot:


```
C:\WINDOWS\system32\cmd.exe

 EAX=00000032  EBX=005F4000  ECX=0060100F  EDX=00000000
 ESI=0060100F  EDI=0060100F  EBP=0074FD54  ESP=0074FD48
 EIP=0060103A  EFL=00000202  CF=0  SF=0  ZF=0  OF=0  AF=0  PF=0

Press any key to continue . . .
```
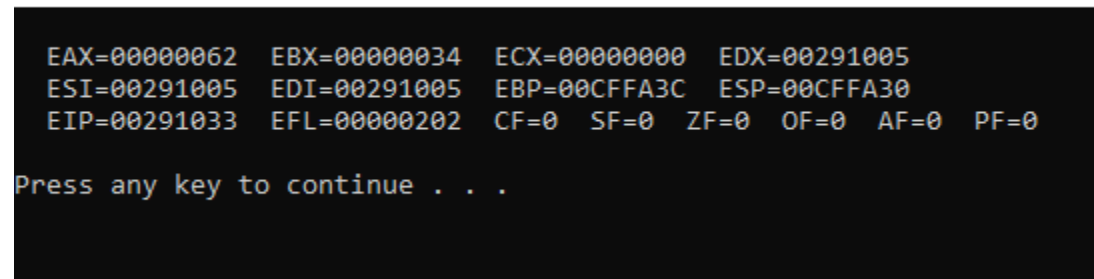
Task 3:

Code:

```
INCLUDE Irvine32.inc
.data
.code
main PROC
mov eax, 70h
mov ebx, 50h
mov ecx, 7
One:
        sub eax, 2
        PUSH ecx
        mov ecx, 4
        Two:
                DEC ebx
        Loop two
        POP ecx
Loop One
call DumpRegs
exit
main ENDP
END main
```

Screenshot:

C:\WINDOWS\system32\cmd.exe

```
 EAX=00000062   EBX=00000034   ECX=00000000   EDX=00291005
 ESI=00291005   EDI=00291005   EBP=00CFFA3C   ESP=00CFFA30
 EIP=00291033   EFL=00000202   CF=0   SF=0   ZF=0   OF=0   AF=0   PF=0

Press any key to continue . . .
```
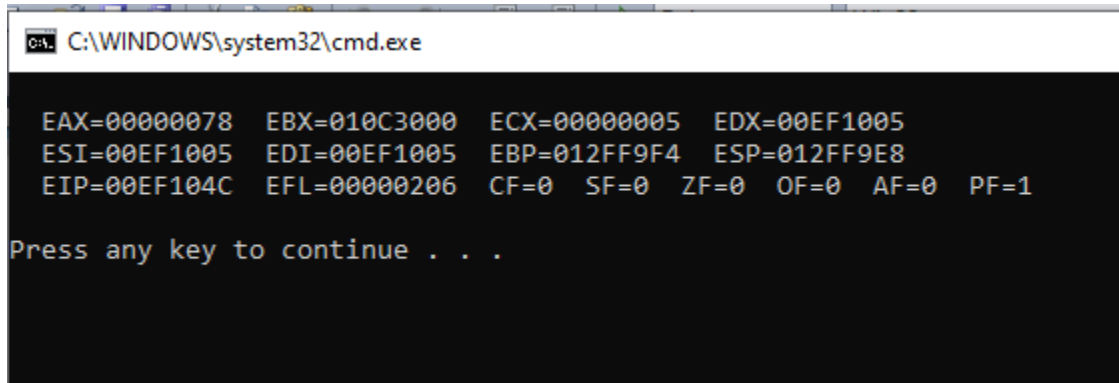
Task 4:

Code:

```
INCLUDE Irvine32.inc
.data
fact WORD 5h
Res WORD 1h
.code
main PROC
mov eax, 0
mov ecx, 0
mov cx, fact
PUSHA
mov ebp, esp
mov esi, [ebp + 4]
mov ax, [ebp + 28]
mov cx, [ebp + 24]
F:
        mov eax, ecx
        mul Res
        mov Res, ax
Loop F
POPA
mov ax, Res
call DumpRegs
exit
main ENDP
END main
```

Screenshot: