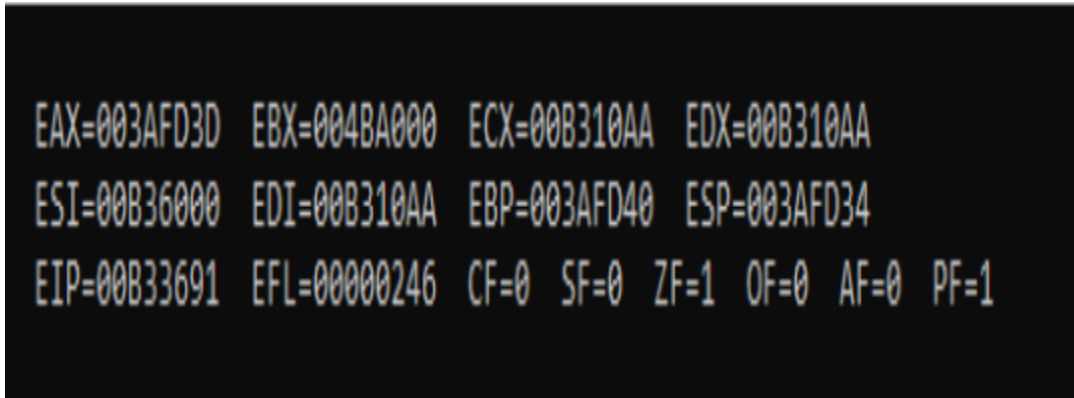


### Question 3:

```
INCLUDE Irvine32.inc
.data
arrb BYTE 61,43,11,52,25
arrayB byte 5 DUP(?)
.code
main PROC
mov esi, offset arrb
mov al, [esi + 2]
mov [arrayB], al
mov al, [esi + 4]
mov [arrayB + 1], al
mov al, [esi + 1]
mov [arrayB + 2], al
mov al, [esi + 3]
mov [arrayB + 3], al
mov al, [esi]
mov [arrayB + 4], al
call DUMPREGS
exit
main ENDP
END main
```



A screenshot of a debugger window with a black background and yellow text. It displays the current state of the CPU registers and status flags. The registers are arranged in three rows: EAX, EBX, ECX, EDX; ESI, EDI, EBP, ESP; and EIP, EFL, CF, SF, ZF, OF, AF, PF. The values are hexadecimal, except for the status flags which are binary.

Register/Flag	Value
EAX	003AFD3D
EBX	004BA000
ECX	00B310AA
EDX	00B310AA
ESI	00B36000
EDI	00B310AA
EBP	003AFD40
ESP	003AFD34
EIP	00B33691
EFL	00000246
CF	0
SF	0
ZF	1
OF	0
AF	0
PF	1

#### Question 4:

```
INCLUDE Irvine32.inc
.data
arrayB BYTE 10, 20, 30
arrayW WORD 150, 250, 350
arrayD DWORD 600, 1200, 1800
SUM1 DWORD 0
SUM2 DWORD 0
SUM3 DWORD 0
.code
main PROC
mov eax, 0
movzx eax, arrayB[0]
movzx ebx, arrayW[0]
add eax, ebx
add eax, arrayD[0]
mov SUM1, eax
movzx eax, arrayB[1]
movzx ebx, arrayW[1]
add eax, ebx
add eax, arrayD[1]
mov SUM2, eax
movzx eax, arrayB[2]
movzx ebx, arrayW[2]
add eax, ebx
add eax, arrayD[2]
mov SUM3, eax
call DUMPREGS
exit
main ENDP
END main
```

```
EAX=04B00118  EBX=000000FA  ECX=00B810AA  EDX=00B810AA
ESI=00B810AA  EDI=00B810AA  EBP=00EFFEC8  ESP=00EFFEBC
EIP=00B836BB  EFL=00000206  CF=0   SF=0   ZF=0   OF=0   AF=0   PF=1
```

### Question 5:

```
INCLUDE Irvine32.inc
.data
array1 BYTE 10, 20, 30, 40
array2 BYTE 4 DUP (?)
.code
main PROC
mov esi, offset array1
mov edi, offset array2
add edi, 3
mov al, [esi]
mov [edi], al
inc esi
dec edi
mov al, [esi]
mov [edi], al
inc esi
dec edi
mov al, [esi]
mov [edi], al
inc esi
dec edi
mov al, [esi]
mov [edi], al
call DUMPREGS
exit
main ENDP
END main
```

```
EAX=006FFF28  EBX=005A3000  ECX=00BD10AA  EDX=00BD10AA
ESI=00BD6003  EDI=00BD6004  EBP=006FFEE4  ESP=006FFED8
EIP=00BD3688  EFL=00000202  CF=0   SF=0   ZF=0   OF=0   AF=0   PF=0
```

### Question 6:

```
INCLUDE Irvine32.inc
.data
arrayD DWORD 1000, 2000, 3000, 4000, 5000
result DWORD 0
.code
main PROC
mov esi, offset arrayD
mov eax, [esi]
add esi, 4
sub eax, [esi]
add esi, 4
sub eax, [esi]
add esi, 4
sub eax, [esi]
add esi, 4
sub eax, [esi]
mov result, eax
call DUMPREGS
exit
main ENDP
END main
```

```
EAX=FFFFCD38  EBX=008F5000  ECX=007610AA  EDX=007610AA
ESI=00766010  EDI=007610AA  EBP=00AFFD58  ESP=00AFFD4C
EIP=00763685  EFL=00000292  CF=0  SF=1  ZF=0  OF=0  AF=1  PF=0
```

### Question 7:

```
INCLUDE Irvine32.inc
.data
arrayB BYTE 60, 70, 80
arrayW WORD 150, 250, 350
arrayD DWORD 600, 1200, 1800
.code
main PROC
MOV ESI,offset arrayB
mov dl,[esi]
add dl,[esi +2 *TYPE ArrayB]
MOV ESI,offset arrayw
mov bx,[esi]
add bx,[esi +2 *TYPE Arrayw]
MOV ESI,offset arrayd
mov eax,[esi]
add eax,[esi +2 *TYPE Arrayw]
call DUMPREGS
exit
main ENDP
END main
```

```
EAX=00000708  EBX=00F901F4  ECX=00C510AA  EDX=00C5108C
ESI=00C56009  EDI=00C510AA  EBP=00D5F82C  ESP=00D5F820
EIP=00C53685  EFL=00000202  CF=0   SF=0   ZF=0   OF=0   AF=0   PF=0
```