

Name : Ibadullah

Roll No : 19k-0259

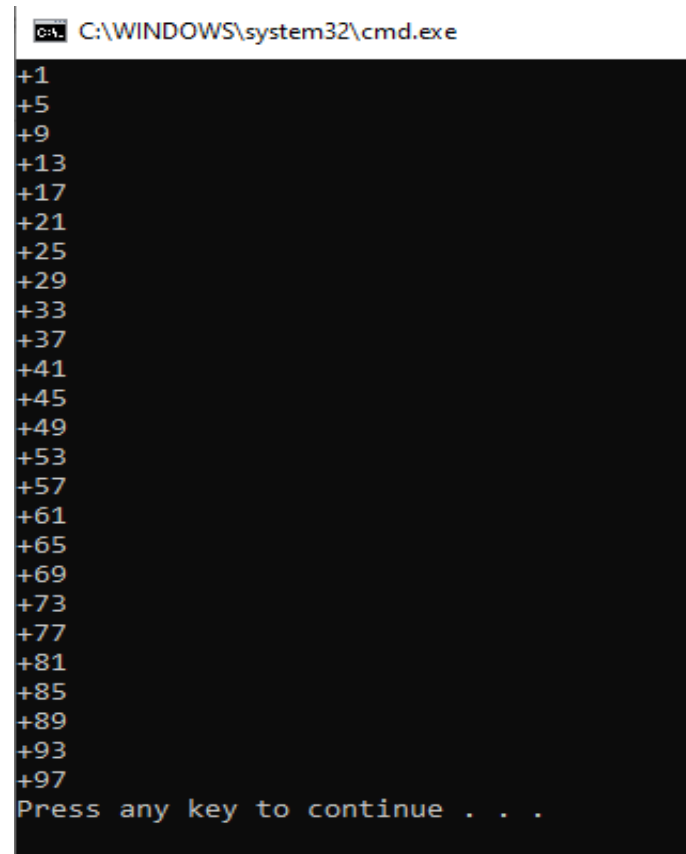
Assignment: 02

Task 1 (a)

Code:

```
INCLUDE Irvine32.inc
.code
main PROC
mov eax, 1
mov ecx, 25
l1:
    call WriteInt
    call crlf
    add eax, 4
Loop l1
exit
main ENDP
END main
```

Screenshot:



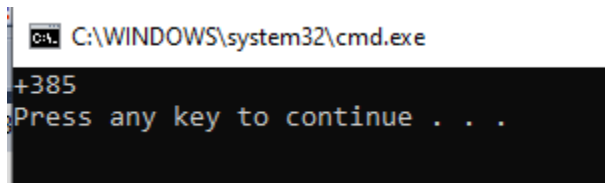
```
C:\WINDOWS\system32\cmd.exe
+1
+5
+9
+13
+17
+21
+25
+29
+33
+37
+41
+45
+49
+53
+57
+61
+65
+69
+73
+77
+81
+85
+89
+93
+97
Press any key to continue . . .
```

Task 1 (b)

Code:

```
INCLUDE Irvine32.inc
.data
array BYTE 1,2,3,4,5,6,7,8,9,10
Sum WORD 0
.code
main PROC
mov eax, 0
mov esi, 0
mov ecx, LENGTHOF array
l1:
    movzx eax, array[esi* TYPE array]
    mul eax
    add DWORD PTR sum, eax
    mov eax, 0
    INC esi
Loop l1
movzx eax, Sum
call WriteInt
call crlf
exit
main ENDP
END main
```

Screenshot:

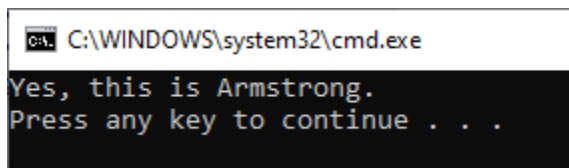


Task 2:

Code:

```
INCLUDE Irvine32.inc
.data
var1 BYTE 1
var2 BYTE 5
var3 BYTE 3
msg1 BYTE "Yes, this is Armstrong.",0
msg2 BYTE "No, this is not a Armstrong.",0
Sum DWORD 0
.code
main PROC
movzx eax, var1
movzx ebx, var2
movzx edx, var3
mov ecx, 24
l1:
    add ebx, 5
Loop l1
mov ecx, 8
l2:
    add edx, 3
Loop l2
add sum, eax
add sum, ebx
add sum, edx
cmp sum, 153
je equal
mov edx, OFFSET msg2
call WriteString
call crlf
jmp not_equal
equal:
mov edx, OFFSET msg1
call WriteString
call crlf
not_equal:
exit
main ENDP
END main
```

Screenshot:



Task 3:

a)

Registers
EAX = 005B4000

Registers
EAX = 000005DE

b)

Registers
EAX = 005B3FFE

Registers
EAX = 05DE0000

c)

Registers
EAX = 005B4007

Registers
EAX = 00009400

d)

Registers
EAX = 005B4004

Registers
EAX = 000000EA

e)

Registers
EAX = 005B4010

Registers
EAX = 00000289

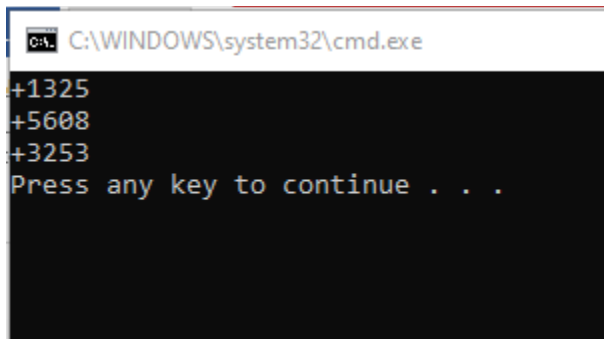
Task 4:

Code:

```
INCLUDE Irvine32.inc
.data
Array1 DB 11,22,33
Array2 DW 135,195,210
Array3 DD 1182,5394,3013
Res1 DWORD ?
Res2 DWORD ?
Res3 DWORD ?
.code
main PROC
mov eax, 0
mov ebx, 0
mov edx, 0
call Addition_Arrays
mov eax, Res1
call WriteInt
call crlf
mov eax, Res2
call WriteInt
call crlf
mov eax, Res3
call WriteInt
call crlf
exit
main ENDP
Addition_Arrays PROC
    mov edi, 0
    mov al, Array1[edi]
    mov ebx, DWORD PTR Array2[edi]
    mov edx, Array3[edi]
    DEC al
    DEC bx
    DEC edx
    add al, bl
    add ax, dx
    mov Res1, eax
    mov eax, 0
    mov ebx, 0
    mov edx, 0
    mov edi, 1
    mov al, Array1[edi*TYPE Array1]
    mov ebx, DWORD PTR Array2[edi*TYPE Array2]
    mov edx, Array3[edi*TYPE Array3]
    DEC al
    DEC bx
    DEC edx
    add al, bl
    add ax, dx
    mov Res2, eax
    mov eax, 0
    mov ebx, 0
    mov edx, 0
    mov edi, 2
```

```
    mov al, Array1[edi*TYPE Array1]
    mov ebx, DWORD PTR Array2[edi*TYPE Array2]
    mov edx, Array3[edi*TYPE Array3]
    DEC al
    DEC bx
    DEC edx
    add al, bl
    add ax, dx
    mov Res3, eax
    RET
Addition_Arrays ENDP
END main
```

Screenshot:



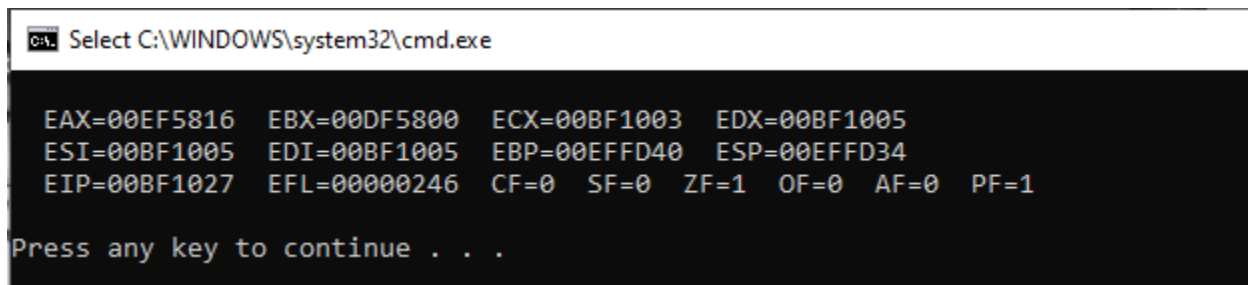
NOTE: Result showing is Addition according to instruction, column wise.

Task 5:

(a):

AX:	5816	AL:	16	AH:	58
CX:	1003	CL:	03	CH:	10
DX:	1005	DL:	05	DH:	10

Screenshot:



A screenshot of a Windows command prompt window. The title bar shows 'C:\WINDOWS\system32\cmd.exe'. The command prompt displays the following register values:

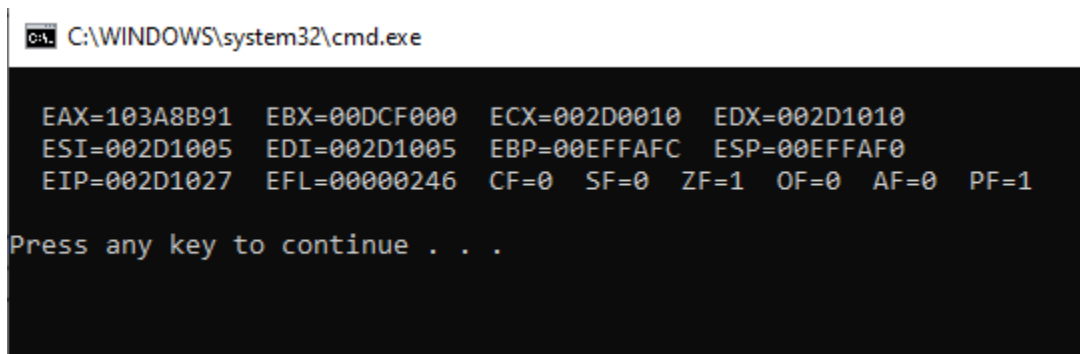
```
EAX=00EF5816  EBX=00DF5800  ECX=00BF1003  EDX=00BF1005  
ESI=00BF1005  EDI=00BF1005  EBP=00EFFF40  ESP=00EFFF34  
EIP=00BF1027  EFL=00000246  CF=0   SF=0   ZF=1   OF=0   AF=0   PF=1
```

Below the register values, it says 'Press any key to continue . . .'

(b):

AX:	8B91	AL:	91	AH:	8B
CX:	0010	CL:	10	CH:	00
DX:	1010	DL:	10	DH:	10

Screenshot:



A screenshot of a Windows command prompt window. The title bar shows 'C:\WINDOWS\system32\cmd.exe'. The command prompt displays the following register values:

```
EAX=103A8B91  EBX=00DCF000  ECX=002D0010  EDX=002D1010  
ESI=002D1005  EDI=002D1005  EBP=00EFFAFC  ESP=00EFFAF0  
EIP=002D1027  EFL=00000246  CF=0   SF=0   ZF=1   OF=0   AF=0   PF=1
```

Below the register values, it says 'Press any key to continue . . .'

Task 6:

Code:

```
INCLUDE Irvine32.inc
.data
one DWORD ?,?,?,?,?,?,?,?
      DWORD ?,?,?,?,?,?,?,?
.code
main PROC
mov eax, 1
mov esi, 0
mov ecx, 4
l1:
    PUSH ecx
    mov ecx, 4
    l2:
        mov one[esi*TYPE one], eax
        INC esi
        INC eax
    Loop l2
    POP ecx
Loop l1
mov esi, 0
mov ecx, 2
l3:
    PUSH ecx
    mov ecx, LENGTHOF one
    l4:
        mov eax, one[esi*TYPE one]
        INC esi
        call WriteInt
    Loop l4
    call crlf
    POP ecx
Loop l3
exit
main ENDP
END main
```

Screenshot:

