

COAL Lab 10

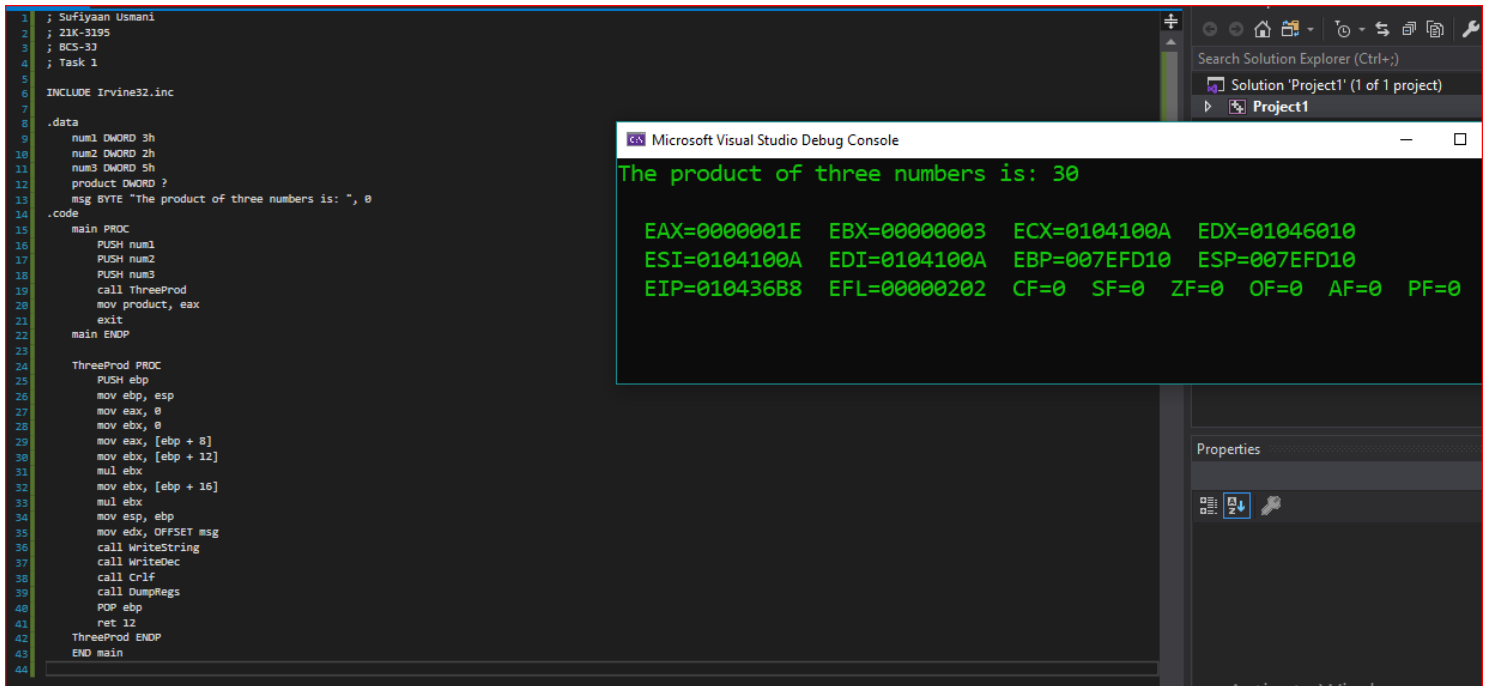
Name: Sufiyaan Usmani

Roll No: 21K-3195

Section: BCS-3J

Instructor: Sir Kashan Hussain

Task 1:



The screenshot displays the Visual Studio IDE with an assembly file open on the left and the Microsoft Visual Studio Debug Console on the right. The assembly code defines three numbers (1, 2, 3) and calculates their product (6). The debug console shows the output of the program, which is "The product of three numbers is: 30". The register dump shows the state of the CPU registers after the program execution.

```
1 ; Sufiyaan Usmani
2 ; 21K-3195
3 ; BCS-3J
4 ; Task 1
5
6 INCLUDE Irvine32.inc
7
8 .data
9     num1 DWORD 1h
10    num2 DWORD 2h
11    num3 DWORD 3h
12    product DWORD ?
13    msg BYTE "The product of three numbers is: ", 0
14
15 .code
16 main PROC
17     PUSH num1
18     PUSH num2
19     PUSH num3
20     call ThreeProd
21     mov product, eax
22     exit
23 main ENDP
24
25 ThreeProd PROC
26     PUSH ebp
27     mov ebp, esp
28     mov eax, 0
29     mov ebx, 0
30     mov eax, [ebp + 8]
31     mov ebx, [ebp + 12]
32     mul ebx
33     mov ebx, [ebp + 16]
34     mul ebx
35     mov esp, ebp
36     mov edx, OFFSET msg
37     call WriteString
38     call WriteDec
39     call CrLf
40     call DumpRegs
41     POP ebp
42     ret 12
43 ThreeProd ENDP
44 END main
```

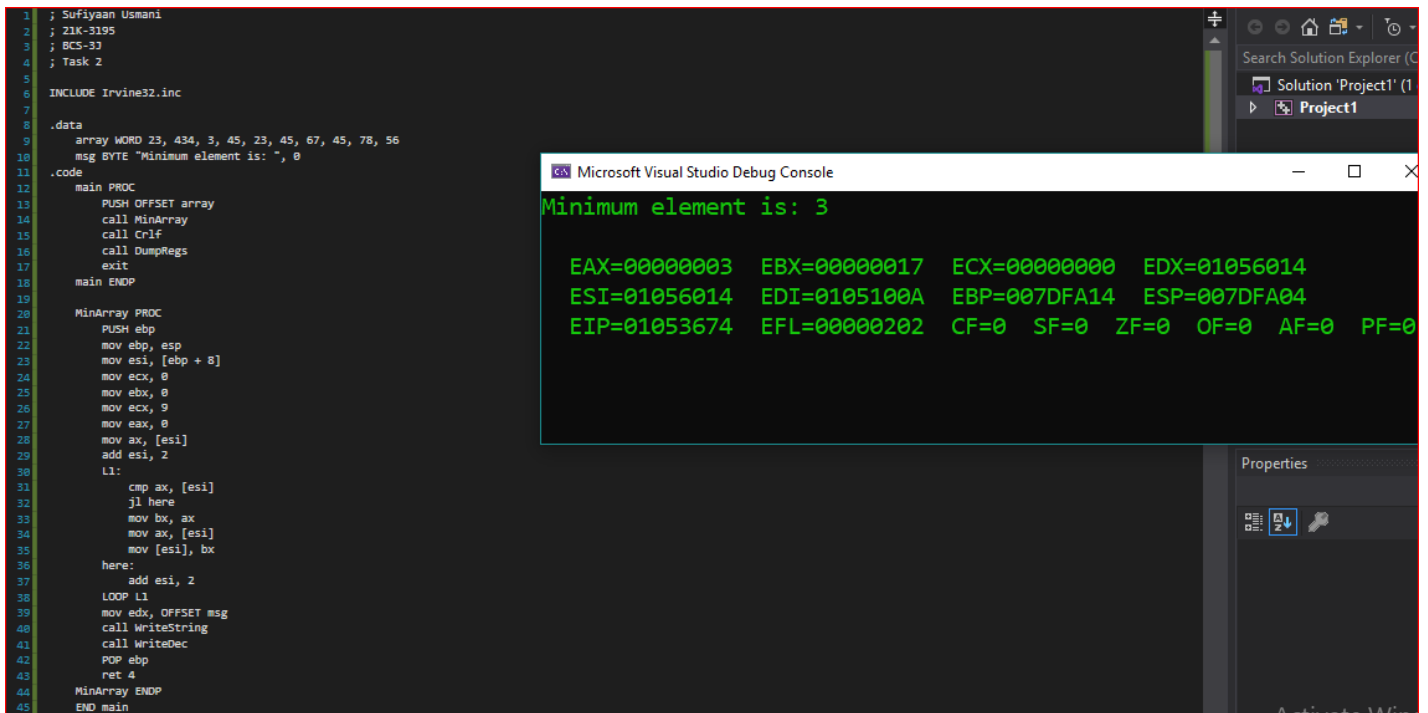
Microsoft Visual Studio Debug Console

The product of three numbers is: 30

EAX=0000001E EBX=00000003 ECX=0104100A EDX=01046010
ESI=0104100A EDI=0104100A EBP=007EFD10 ESP=007EFD10
EIP=010436B8 EFL=00000202 CF=0 SF=0 ZF=0 OF=0 AF=0 PF=0

Properties

Task 2:



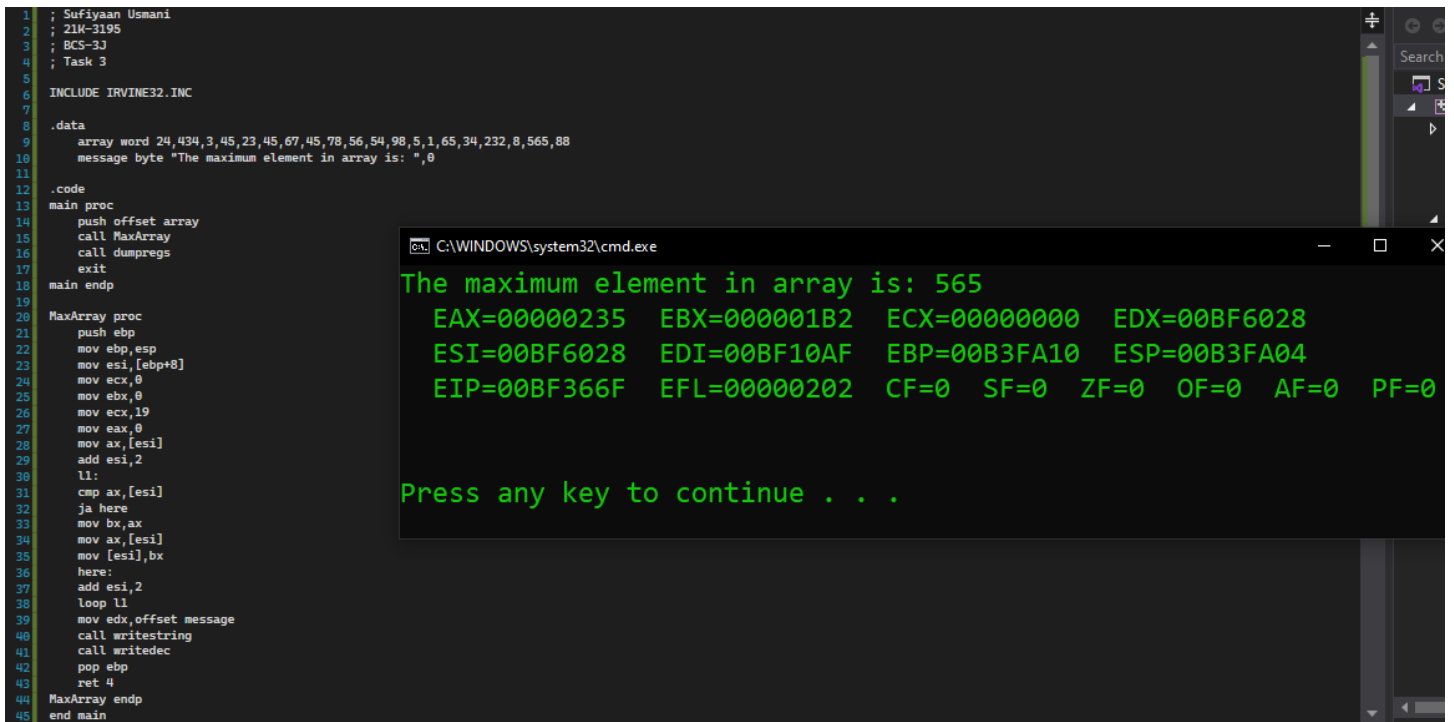
The screenshot shows the Visual Studio IDE with an assembly file open. The code defines a data segment with an array of 10 words and a message. The main procedure calls MinArray, which finds the minimum element in the array. The debug console shows the output: "Minimum element is: 3".

```
1 ; Sufiyaan Usmani
2 ; 21K-3195
3 ; BCS-3J
4 ; Task 2
5
6 INCLUDE Irvine32.inc
7
8 .data
9 array WORD 23, 434, 3, 45, 23, 45, 67, 45, 78, 56
10 msg BYTE "Minimum element is: ", 0
11
12 .code
13 main PROC
14     PUSH OFFSET array
15     call MinArray
16     call Crlf
17     call DumpRegs
18     exit
19 main ENDP
20
21 MinArray PROC
22     PUSH ebp
23     mov ebp, esp
24     mov esi, [ebp + 8]
25     mov ecx, 0
26     mov ebx, 0
27     mov ecx, 9
28     mov eax, [esi]
29     add esi, 2
30     L1:
31         cmp ax, [esi]
32         jl here
33         mov bx, ax
34         mov ax, [esi]
35         mov [esi], bx
36     here:
37         add esi, 2
38     LOOP L1
39     mov edx, OFFSET msg
40     call WriteString
41     call WriteDec
42     POP ebp
43     ret 4
44 MinArray ENDP
45 END main
```

Microsoft Visual Studio Debug Console

```
Minimum element is: 3
EAX=00000003 EBX=00000017 ECX=00000000 EDX=01056014
ESI=01056014 EDI=0105100A EBP=007DFA14 ESP=007DFA04
EIP=01053674 EFL=00000202 CF=0 SF=0 ZF=0 OF=0 AF=0 PF=0
```

Task 3:



The screenshot shows the Visual Studio IDE with an assembly file open. The code defines a data segment with an array of 10 words and a message. The main procedure calls MaxArray, which finds the maximum element in the array. The debug console shows the output: "The maximum element in array is: 565".

```
1 ; Sufiyaan Usmani
2 ; 21K-3195
3 ; BCS-3J
4 ; Task 3
5
6 INCLUDE Irvine32.inc
7
8 .data
9 array word 24, 434, 3, 45, 23, 45, 67, 45, 78, 56, 54, 98, 5, 1, 65, 34, 232, 8, 565, 88
10 message byte "The maximum element in array is: ", 0
11
12 .code
13 main proc
14     push offset array
15     call MaxArray
16     call dumpregs
17     exit
18 main endp
19
20 MaxArray proc
21     push ebp
22     mov ebp, esp
23     mov esi, [ebp+8]
24     mov ecx, 0
25     mov ebx, 0
26     mov ecx, 19
27     mov eax, 0
28     mov ax, [esi]
29     add esi, 2
30     L1:
31         cmp ax, [esi]
32         ja here
33         mov bx, ax
34         mov ax, [esi]
35         mov [esi], bx
36     here:
37         add esi, 2
38     loop L1
39     mov edx, offset message
40     call writestring
41     call writedec
42     pop ebp
43     ret 4
44 MaxArray endp
45 end main
```

C:\WINDOWS\system32\cmd.exe

```
The maximum element in array is: 565
EAX=00000235 EBX=000001B2 ECX=00000000 EDX=00BF6028
ESI=00BF6028 EDI=00BF10AF EBP=00B3FA10 ESP=00B3FA04
EIP=00BF366F EFL=00000202 CF=0 SF=0 ZF=0 OF=0 AF=0 PF=0
Press any key to continue . . .
```

Task 4:

```
1 ; Sufiyaan Usmani
2 ; 21K-3195
3 ; BCS-3J
4 ; Task 4
5
6 INCLUDE Irvine32.inc
7 .data
8     message byte "Enter an integer: ",0
9     message2 byte "The square of that number is: ",0
10 .code
11 main proc
12     call LocalSquare
13     call dumpregs
14     exit
15 main endp
16
17 LocalSquare PROC
18     Enter 1,0
19     mov eax,0
20     mov edx,offset message
21     call writestring
22     call readint
23     mov [ebp-4],eax
24     mov edx,offset message2
25     call writestring
26     mul eax
27     call writedec
28     leave
29     ret
30 LocalSquare endp
31 end main
```

C:\WINDOWS\system32\cmd.exe

Enter an integer: 3
The square of that number is: 9
EAX=00000009 EBX=00D35000 ECX=007310AF EDX=00000000
ESI=007310AF EDI=007310AF EBP=00F5FB08 ESP=00F5FAFC
EIP=0073366A EFL=00000202 CF=0 SF=0 ZF=0 OF=0 AF=0 PF=0
Press any key to continue . . .

Task 5:

```
1 ; Sufiyaan Usmani
2 ; 21K-3195
3 ; BCS-3J
4 ; Task 5
5
6 INCLUDE Irvine32.inc
7 .data
8     message byte "Enter the integer you want the factorial of: ",0
9     message2 byte "The factorial is: ",0
10 var dword ?
11 .code
12 main proc
13     mov edx,offset message
14     call writestring
15     mov eax,0
16     call readdec
17     mov var,eax
18     mov ebx,0
19     mov ebx,eax
20     dec ebx
21     call fact
22     mov edx,offset message2
23     call writestring
24     call writedec
25     call dumpregs
26     exit
27 main endp
28
29 fact proc
30     enter 0,0
31     cmp eax,0
32     je h1
33     cmp ebx,0
34     je here
35     mul ebx
36     dec ebx
37     call fact
38     jmp here
39 h1:
40     mov eax,1
41 here:
42     leave
43     ret
44 fact endp
45 end main
```

C:\WINDOWS\system32\cmd.exe

Enter the integer you want the factorial of: 5
The factorial is: 120
EAX=00000078 EBX=00000000 ECX=009E10AF EDX=009E602E
ESI=009E10AF EDI=009E10AF EBP=009BFAD0 ESP=009BFAC4
EIP=009E369A EFL=00000202 CF=0 SF=0 ZF=0 OF=0 AF=0 PF=0
Press any key to continue . . .

Lab Exercise:

```
1 ; Sufiyaan Usmani
2 ; 21K-3195
3 ; BCS-3J
4
5 INCLUDE Irvine32.inc
6
7 .data
8     x DWORD 5
9     y DWORD 6
10 .code
11     main PROC
12         mov eax, x
13         call WriteDec
14         mov al, " "
15         call WriteChar
16         mov eax, y
17         call WriteDec
18         call Crlf
19         PUSH x
20         PUSH OFFSET y
21         call change
22         mov eax, x
23         call WriteDec
24         mov al, " "
25         call WriteChar
26         mov eax, y
27         call WriteDec
28         call Crlf
29         exit
30     main ENDP
31
32     change PROC
33         PUSH ebp
34         mov ebp, esp
35         mov DWORD PTR [ebp + 12], 10
36         mov eax, DWORD PTR [ebp + 8]
37         mov DWORD PTR [eax], 14
38         mov eax, DWORD PTR [ebp + 12]
39         call WriteDec
40         mov al, " "
41         call WriteChar
42         mov eax, DWORD PTR [ebp + 8]
43         mov eax, DWORD PTR [eax]
44         call WriteDec
45         call Crlf
46         mov esp, ebp
47         POP ebp
48         ret 8
49     change ENDP
50 END main
```

C:\WINDOWS\system32\cmd.exe

```
5 6
10 14
5 14
Press any key to continue . . .
```