Assignment # 1

CS-240 COAL

Tentative Submission date (22- 04-2024)

(Submission is expected to be in hard as well as soft)

Name: **Umair Ahmad**

Reg# **NUM-BSCS-2022-36**

1. **What will be the value in EDX after each of the lines marked (a) and (b) execute?**

```
.data
one WORD 8002h
two WORD  4321h
.code
mov edx,21348041h
movsx edx,one ; (a)
movsx edx,two ; (b)
```

Output:

```
creating a PE executable

  EAX=0018FFCC  EBX=7FFDE000  ECX=00000000  EDX=00004321
  ESI=772E9191  EDI=00401000  EBP=0018FF94  ESP=0018FF8C
  EIP=00401018  EFL=00000246  CF=0  SF=0  ZF=1  OF=0  AF=0  PF=1

PS C:\Users\Umair_rana\Desktop\Assignment>
```

2. **What will be the value in EAX after the following lines execute?**

```
mov eax,1002FFFFh
inc ax
```

Output:

```
searching libraries
creating a PE executable

  EAX=10020000  EBX=7FFDE000  ECX=00000000  EDX=00004321
  ESI=772E9191  EDI=00401000  EBP=0018FF94  ESP=0018FF8C
  EIP=0040101F  EFL=00000256  CF=0  SF=0  ZF=1  OF=0  AF=1  PF=1

PS C:\Users\Umair_rana\Desktop\Assignment>
```

3. **What will be the value in EAX after the following lines execute?**

```
mov eax,30020000h
dec ax
```

Output:

```
  EAX=3002FFFF  EBX=7FFDE000  ECX=00000000  EDX=00401000
  ESI=772E9191  EDI=00401000  EBP=0018FF94  ESP=0018FF8C
  EIP=0040100C  EFL=00000296  CF=0  SF=1  ZF=0  OF=0  AF=1  PF=1

PS C:\Users\Umair_rana\Desktop\Assignment>
```

4. **What will be the value in EAX after the following lines execute?**

```
mov eax,1002FFFFh
neg ax
```

Output:

```
searching libraries
creating a PE executable

  EAX=10020001  EBX=7FFDE000  ECX=00000000  EDX=00401000
  ESI=772E9191  EDI=00401000  EBP=0018FF94  ESP=0018FF8C
  EIP=0040100D  EFL=00000213  CF=1  SF=0  ZF=0  OF=0  AF=1  PF=0

PS C:\Users\Umair_rana\Desktop\Assignment>
```

5. **What will be the value of the Parity flag after the following lines execute?**

```
mov al,1
add al,3
```

Output:

```
creating a PE executable

  EAX=0018FF04  EBX=7FFDE000  ECX=00000000  EDX=00401000
  ESI=772E9191  EDI=00401000  EBP=0018FF94  ESP=0018FF8C
  EIP=00401009  EFL=00000202  CF=0  SF=0  ZF=0  OF=0  AF=0  PF=0

PS C:\Users\Umair_rana\Desktop\Assignment>
```

6. **What will be the value of EAX and the Sign flag after the following lines execute?**

```
mov eax,5
sub eax,6
```

Output:

```
creating a PE executable

  EAX=FFFFFFFF  EBX=7FFDE000  ECX=00000000  EDX=00401000
  ESI=772E9191  EDI=00401000  EBP=0018FF94  ESP=0018FF8C
  EIP=0040100D  EFL=00000297  CF=1  SF=1  ZF=0  OF=0  AF=1  PF=1

PS C:\Users\Umair_rana\Desktop\Assignment>
```

7. **In the following code, the value in AL is intended to be a signed byte. Explain how the Overflow flag helps, or does not help you, to determine whether the final value in AL falls within a valid signed range.**

```
mov al,-1
add al,130
```

ANS:

In the provided assembly code, the intention is to perform arithmetic operations on a signed byte stored in the AL register. The AL register is an 8-bit register, which can hold values from -128 to 127 in two's complement form.

The mov al, -1 instruction sets AL to -1, which is represented in two's complement as 0xFF. The add al, 130 instruction then adds 130 to AL, which in two's complement form is 0x82.

The Overflow flag (OF) is set when a signed arithmetic operation results in a value that is too large to be represented in the destination register. In this case, adding 130 to -1 (0xFF) in a signed context would result in 129, which is outside the valid signed range for an 8-bit value.

After the addition, the OF flag will be set because the result does not fit within a signed byte. This indicates that the final value in AL (0x82) is outside the valid signed range, and it helps us determine that overflow has occurred in the signed arithmetic operation

8. **What value will RAX contain after the following instruction executes?**

```
mov rax,44445555h
```

ANS:

In the provided assembly code, the error is that the rax register is being used without including the necessary 64-bit register prefix (qword ptr). The correct syntax for moving a 32-bit value (44445555h) into the rax register is

Output:

```
Portions Copyright (c) 1992-2002 Sybase, Inc. All Rights Reserved.
Source code is available under the Sybase Open Watcom Public License.

c:\Users\Umair_rana\Desktop\Assignment\Task8.temp.asm(7) : Error A2102: Symbol not defined : rax
c:\Users\Umair_rana\Desktop\Assignment\Task8.temp.asm: 12 lines, 2 passes, 15 ms, 0 warnings, 1 errors
Error A2106: Cannot open file: ";" [ENOENT]

C:\Users\Umair_rana\Desktop\Assignment>
```

9. **What value will RAX contain after the following instructions execute?**

```
.data
dwordVal DWORD 84326732h

.code
mov rax,0FFFFFFFF00000000h  mov
rax,dwordVal
```

ANS:

The error in the provided assembly code is related to the instruction mov rax, 0FFFFFFFF00000000h. This error occurs because the constant value 0FFFFFFFF00000000h is too large to fit into a 64-bit register (rax) in a single instruction. The mov instruction expects a 64-bit immediate value, but the given value exceeds the range of a signed 64-bit integer.

Output:

```
Source code is available under the Sybase Open Watcom Public License.

c:\Users\Umair_rana\Desktop\Assignment\Task9.temp.asm(7) : Error A2235: Constant value too large: FFFFFFFF00000000h
c:\Users\Umair_rana\Desktop\Assignment\Task9.temp.asm(8) : Error A2049: Invalid instruction operands
c:\Users\Umair_rana\Desktop\Assignment\Task9.temp.asm: 13 lines, 1 passes, 16 ms, 0 warnings, 2 errors
Error A2106: Cannot open file: ";" [ENOENT]

C:\Users\Umair_rana\Desktop\Assignment>
```

10. **What value will EAX contain after the following instructions execute?**

```
.data
dVal DWORD 12345678h
.code mov ax,3 mov
WORD PTR dVal+2, ax
mov eax,dVal
```

Output:

```
creating a PE executable

  EAX=00035678  EBX=7FFDE000  ECX=00000000  EDX=00401000
  ESI=77909191  EDI=00401000  EBP=0018FF94  ESP=0018FF8C
  EIP=00401014  EFL=00000246  CF=0  SF=0  ZF=1  OF=0  AF=0  PF=1

PS C:\Users\Umair_rana\Desktop\Assignment>
```

11. **What will EAX contain after the following instructions execute?**

```
.data
dVal DWORD ?
.code
mov dVal,12345678h
mov ax,WORD PTR dVal+2
add ax,3
mov WORD PTR
dVal,ax
mov eax,dVal
```

Output:

```
  EAX=12341237  EBX=7FFDE000  ECX=00000000  EDX=00401000
  ESI=772E9191  EDI=00401000  EBP=0018FF94  ESP=0018FF8C
  EIP=00401024  EFL=00000202  CF=0  SF=0  ZF=0  OF=0  AF=0  PF=0

PS C:\Users\Umair_rana\Desktop\Assignment>
PS C:\Users\Umair_rana\Desktop\Assignment>
```

s