Lab: 4

Fahid Imran

Roll No: 23i-0061

COAL

Instructor

Mr. Sulaman Saboor

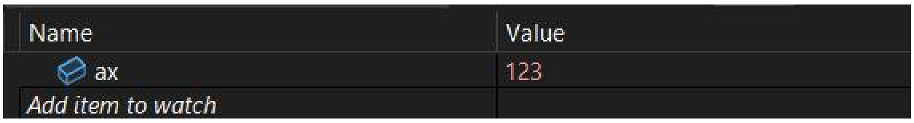
Fast NUCES Islamabad Campus

Task1:

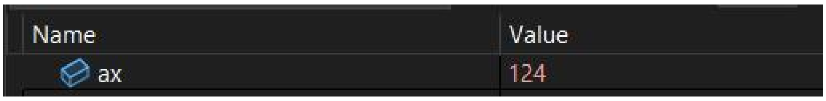
Code:

```
.386
.model flat, stdcall
.stack 4098

.code
main PROC
mov ax , 123
inc ax
inc ax
inc ax
inc ax
dec ax
dec ax
dec ax
dec ax
main endp
end main
```

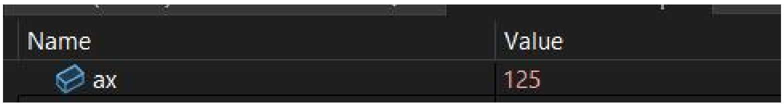
Output:



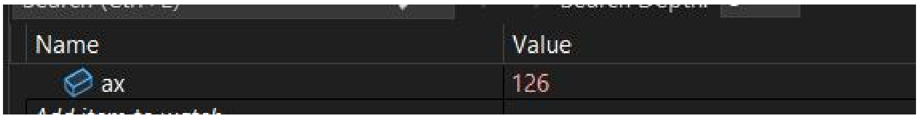
Name	Value
ax	123
Add item to watch	



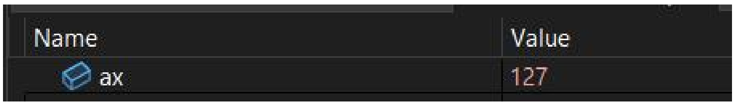
Name	Value
ax	124



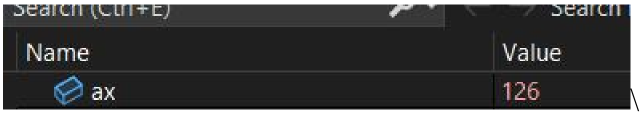
Name	Value
ax	125



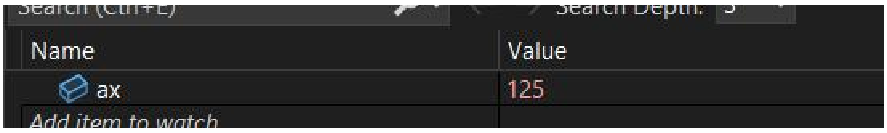
Name	Value
ax	126



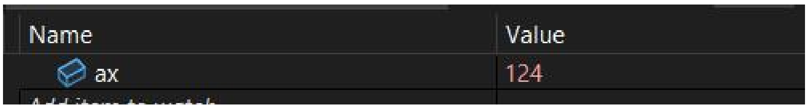
Name	Value
ax	127



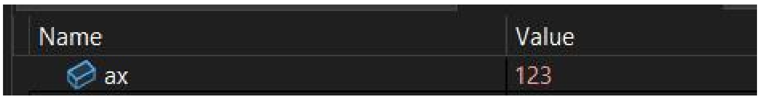
Name	Value
ax	126



```
Search (Ctrl+E)                      Search Depth: 3
Name                                 Value
    ax                               125
Add item to watch
```



```
Name                                 Value
    ax                               124
```



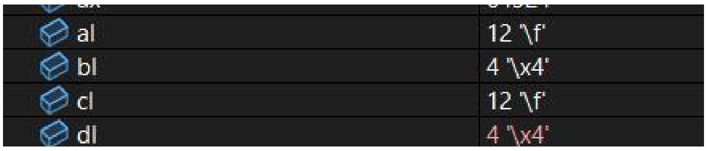
```
Name                                 Value
    ax                               123
```

Task2:

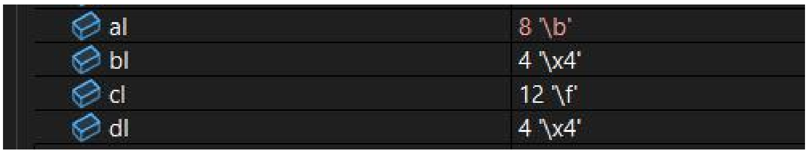
Code:

```
    .386
    .model flat, stdcall
    .stack 4098

    .code
    main PROC
    mov al , 12
    mov bl , 4
    mov cl , al
    mov dl , bl

    sub al, bl
    mov dl, al
    mov al, cl
    sub bl, al
    mov al, dl
    add al, bl

    main endp
end main
```

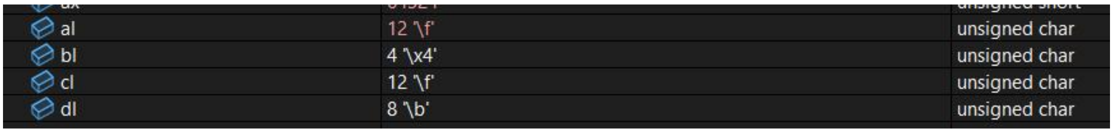
Output:



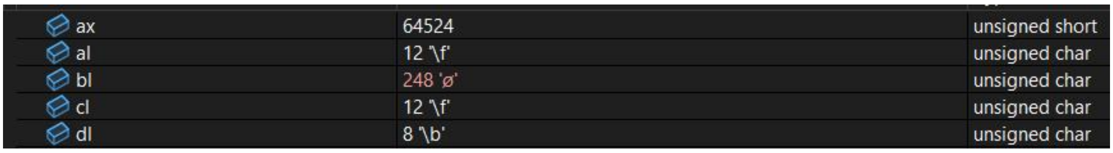
```
  al                                 12 '\f'
  bl                                 4 '\x4'
  cl                                 12 '\f'
  dl                                 4 '\x4'
```



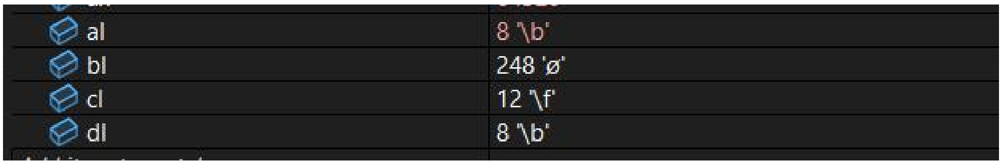
```
    al                               8 '\b'
    bl                               4 '\x4'
    cl                               12 '\f'
    dl                               4 '\x4'
```



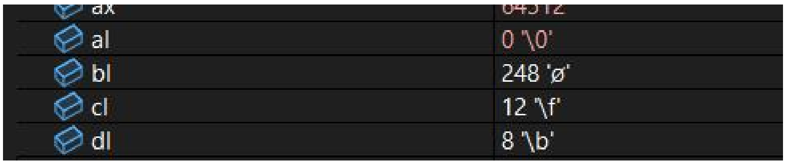
al	12 '\f'	unsigned char
bl	4 '\x4'	unsigned char
cl	12 '\f'	unsigned char
dl	8 '\b'	unsigned char



ax	64524	unsigned short
al	12 '\f'	unsigned char
bl	248 'ø'	unsigned char
cl	12 '\f'	unsigned char
dl	8 '\b'	unsigned char



al	8 '\b'
bl	248 'ø'
cl	12 '\f'
dl	8 '\b'



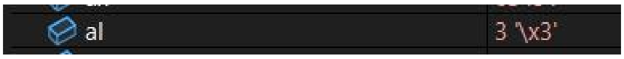
ax	64512
al	0 '\0'
bl	248 'ø'
cl	12 '\f'
dl	8 '\b'

Task3:

Code:

```
    .386
    .model flat, stdcall
    .stack 4098

    .code
    main PROC
    mov al , 3

    main endp
end main
```

Output:



al	3 '\x3'

Task4:

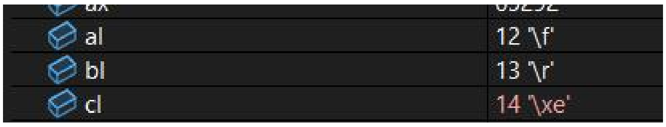
Code:

```
    .386
    .model flat, stdcall
    .stack 4098
    .data
```

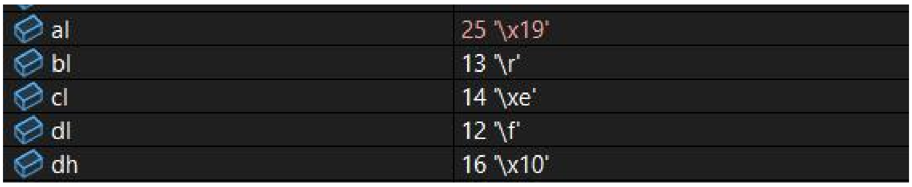
```
num1 word ?
num2 word ?
num3 word ?

.code
main PROC
mov al , 12
mov bl , 13
mov cl , 14

mov dl , al
add al , bl
mov dh , al
mov al , dl
mov dl , cl
add cl , dh
mov dh , cl
mov cl , dl

mov dl , al
sub al , bl
mov dh , al
mov al , dl
mov dl , cl
sub cl , dh
mov dh , cl
mov cl , dl

mov ax, 14
mov bx, 15
mov cx, 16
mov edx , 0

mov num1, ax
mov num2, bx
mov num3, cx

div bx
mov edx, 0
div cx
```

```
    mov ax, num1
    mov bx, num2
    mov cx, num3

    mul bx
    mul cx

    main endp
end main
```

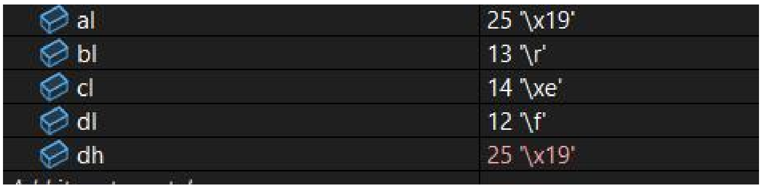
Output:



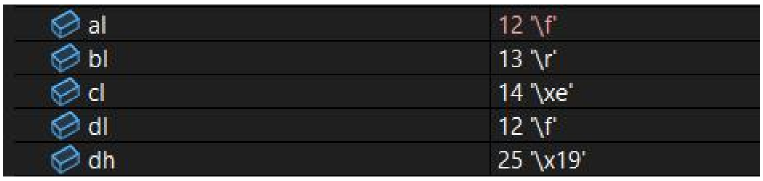
```
al    12 '\f'
bl    13 '\r'
cl    14 '\xe'
```



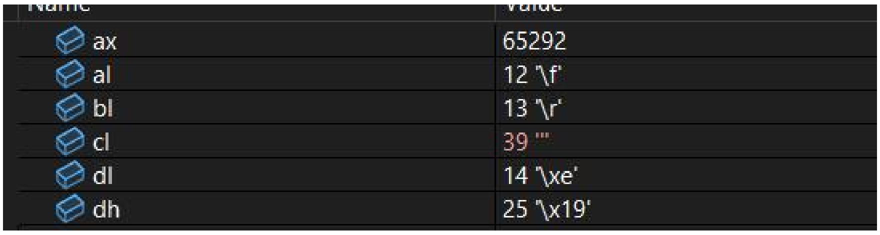
```
al    25 '\x19'
bl    13 '\r'
cl    14 '\xe'
dl    12 '\f'
dh    16 '\x10'
```



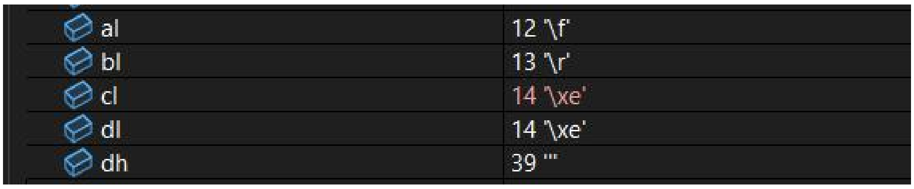
```
al    25 '\x19'
bl    13 '\r'
cl    14 '\xe'
dl    12 '\f'
dh    25 '\x19'
```



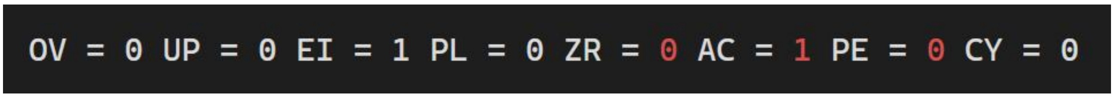
```
al    12 '\f'
bl    13 '\r'
cl    14 '\xe'
dl    12 '\f'
dh    25 '\x19'
```



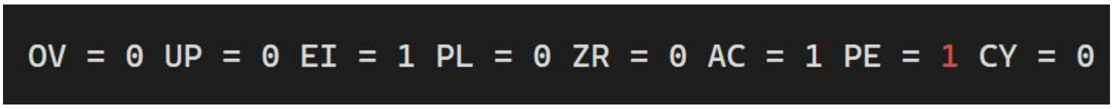
```
ax    65292
al    12 '\f'
bl    13 '\r'
cl    39 '''
dl    14 '\xe'
dh    25 '\x19'
```



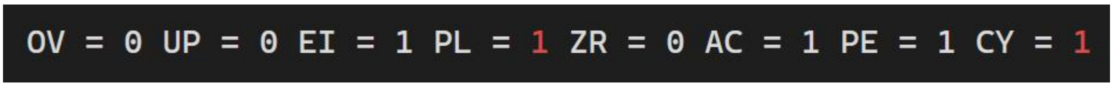
```
al    12 '\f'
bl    13 '\r'
cl    14 '\xe'
dl    14 '\xe'
dh    39 '''
```



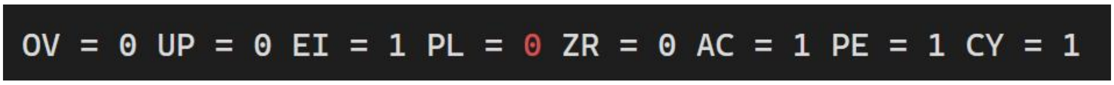
```
OV = 0 UP = 0 EI = 1 PL = 0 ZR = 0 AC = 1 PE = 0 CY = 0
```



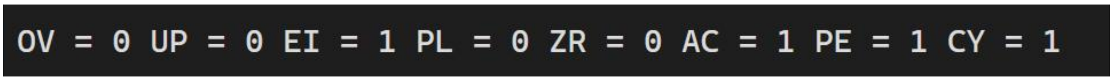
```
OV = 0 UP = 0 EI = 1 PL = 0 ZR = 0 AC = 1 PE = 1 CY = 0
```



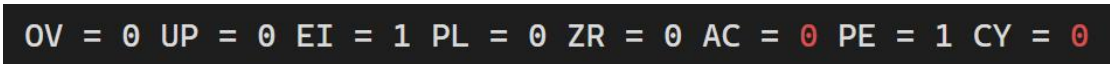
```
OV = 0 UP = 0 EI = 1 PL = 1 ZR = 0 AC = 1 PE = 1 CY = 1
```



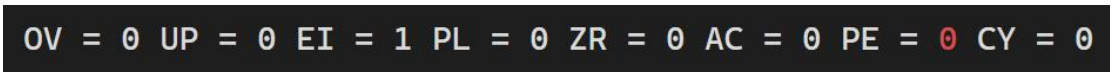
```
OV = 0 UP = 0 EI = 1 PL = 0 ZR = 0 AC = 1 PE = 1 CY = 1
```



```
OV = 0 UP = 0 EI = 1 PL = 0 ZR = 0 AC = 1 PE = 1 CY = 1
```



```
OV = 0 UP = 0 EI = 1 PL = 0 ZR = 0 AC = 0 PE = 1 CY = 0
```



```
OV = 0 UP = 0 EI = 1 PL = 0 ZR = 0 AC = 0 PE = 0 CY = 0
```

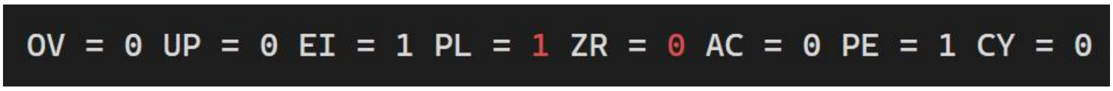
Task5:

Code:

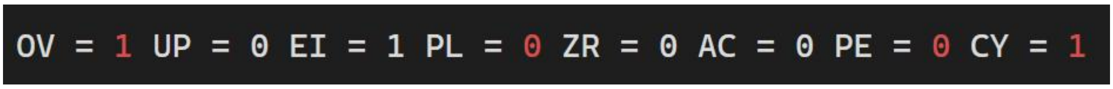
```
.386
.model flat, stdcall
.stack 4098
.data
num1 sbyte ?
num2 sbyte ?
num3 byte ?
num4 byte ?
.code
main PROC

mov num3, 01010101b
mov num4, 10101010b
mov al , num3
mov ah , num4
add al , ah

mov num1, 11010101b
mov num2, 10101010b
mov al , num1
mov ah , num2
add al , ah

mov num3, 01010101b
mov num4, 10101010b
mov al , num3
mov ah , num4
```

```
    sub al , ah

    mov num1, 11010101b
    mov num2, 10101010b
    mov al , num1
    mov ah , num2
    sub al , ah

    main endp
end main
```

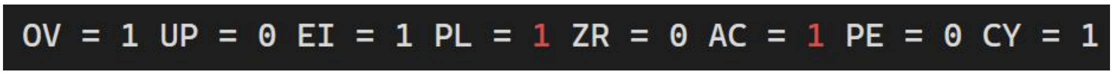
Output:



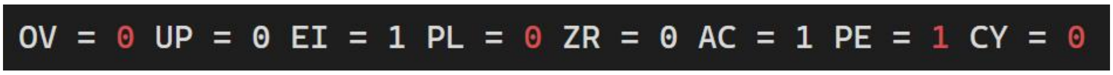
```
OV = 0 UP = 0 EI = 1 PL = 1 ZR = 0 AC = 0 PE = 1 CY = 0
```



```
OV = 1 UP = 0 EI = 1 PL = 0 ZR = 0 AC = 0 PE = 0 CY = 1
```



```
OV = 1 UP = 0 EI = 1 PL = 1 ZR = 0 AC = 1 PE = 0 CY = 1
```



```
OV = 0 UP = 0 EI = 1 PL = 0 ZR = 0 AC = 1 PE = 1 CY = 0
```

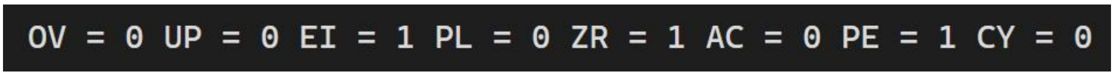
Task6:

Code:

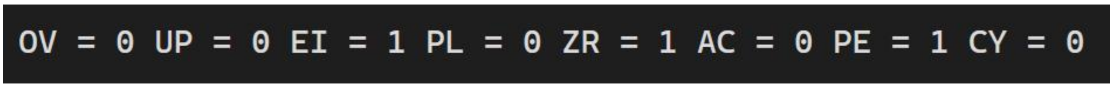
```
.386
.model flat, stdcall
.stack 4098
.data
num1 sword ?
num2 sword ?
num3 word ?
num4 word ?
.code
main PROC

mov num3, 0101010101010101b
mov num4, 1010101010101010b
mov ax , num3
mov bx , num4
mov edx , 0
```

```
    div bx

    mov num1, 1101010111010101b
    mov num2, 1010101010101010b
    mov ax , num1
    mov bx , num2
    mov edx, 0
    div bx

    mov num3, 0101010101010101b
    mov num4, 1010101010101010b
    mov ax , num3
    mov bx , num4
    mul bx

    mov num1, 1101010111010101b
    mov num2, 1010101010101010b
    mov ax , num1
    mov bx , num2
    mul bx

    main endp
end main
```

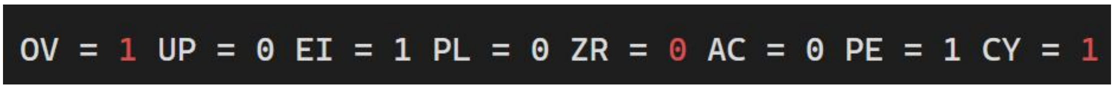
Output:



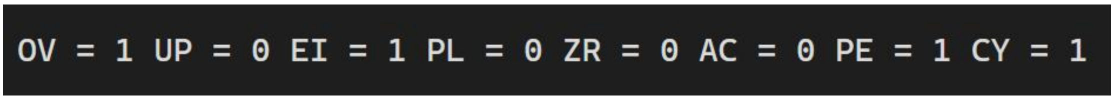
```
OV = 0 UP = 0 EI = 1 PL = 0 ZR = 1 AC = 0 PE = 1 CY = 0
```



```
OV = 0 UP = 0 EI = 1 PL = 0 ZR = 1 AC = 0 PE = 1 CY = 0
```



```
OV = 1 UP = 0 EI = 1 PL = 0 ZR = 0 AC = 0 PE = 1 CY = 1
```



```
OV = 1 UP = 0 EI = 1 PL = 0 ZR = 0 AC = 0 PE = 1 CY = 1
```

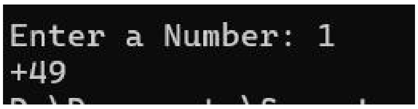
Task7:

Code:

```
include irvine32.inc
.386
.model flat,stdcall
```

```
.stack 4096

.data
a dword ?
b dword ?
result dword ?
remainder dword ?
var1 BYTE "Enter a Number: ", 0
var2 BYTE "Result ", 0


.code
main PROC

    mov edx, offset var1
    call writestring
    call readint
    add eax, 48

    call writeint




    main endp
end main
```

Output:



```
Enter a Number: 1
+49
```

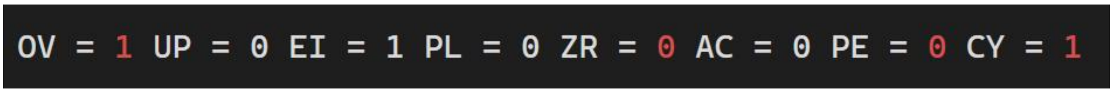
Task8:

Code:

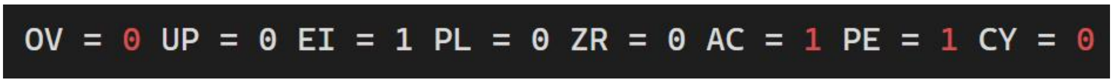
```
.386
.model flat, stdcall
.stack 4098
.data
num1 sbyte ?
```

```
    num2 sbyte ?
    .code
    main PROC

    mov num1, 11010101b
    mov num2, 10101010b
    mov al , num1
    mov ah , num2
    add al , ah

    mov num1, 11010101b
    mov num2, 10101010b
    mov al , num1
    mov ah , num2
    sub al , ah

    main endp
end main
```

Output:



```
OV = 1 UP = 0 EI = 1 PL = 0 ZR = 0 AC = 0 PE = 0 CY = 1
```



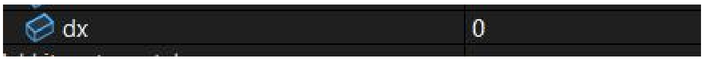
```
OV = 0 UP = 0 EI = 1 PL = 0 ZR = 0 AC = 1 PE = 1 CY = 0
```

Task9:

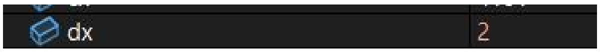
Code:

```
    .386
    .model flat, stdcall
    .stack 4098

    .code
    main PROC

    mov ax , 24
    mov bx , 8
    mov dx , 0
    div bx
    mov ax , 26
    mov bx , 8
    mov dx , 0
    div bx

    main endp
end main
```

Output:



```
dx      0
```



```
dx      2
```

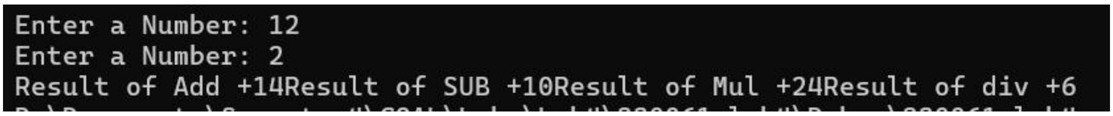
Task10:

Code:

```
include irvine32.inc
.386
.model flat,stdcall
.stack 4096

.data
a dword ?
b dword ?
result dword ?
remainder dword ?
var1 BYTE "Enter a Number: ", 0
var2 BYTE "Result of Add ", 0
var3 BYTE "Result of SUB ", 0
var4 BYTE "Result of Mul ", 0
var5 BYTE "Result of div ", 0
var6 BYTE "Remainder of div ", 0

.code
main PROC

    mov edx, offset var1
    call writestring
    call readint
    mov a, eax
    mov edx, offset var1
    call writestring
    call readint
    mov b , eax

    mov eax, a
    mov ebx, b
    add eax, ebx
    mov result, eax

    mov edx, offset var2
    call writestring
    mov edx, offset result
    call writeint

    mov eax, a
    mov ebx, b
    sub eax, ebx
    mov result, eax

    mov edx, offset var3
    call writestring
    mov edx, offset result
    call writeint

    mov eax, a
    mov ebx, b
    mul ebx
    mov result, eax

    mov edx, offset var4
    call writestring
    mov edx, offset result
    call writeint

    mov eax, a
    mov ebx, b
    mov edx, 0
    div ebx
    mov result, eax
    mov remainder, edx
```

```
        mov edx, offset var5
        call writestring
        mov edx, offset result
        call writeint



       main endp
end main
```

Output:



```
Enter a Number: 12
Enter a Number: 2
Result of Add +14Result of SUB +10Result of Mul +24Result of div +6
```