Assignment 3 Date:_____

QUESTION 1

```
include irvine32.inc

.data
    dividend dword 0D4A4h
    divisor dword 0Ah.

.code
    main proc
        push divisor
        push dividend
        call divide

        exit
    main endp

    divide proc,
        -dividend: dword
        -divisor: dword

        mov eax, -dividend
        call writehex
        call crlf
```

```asm
    cmp eax, 5h
    jle done

    mov edx, 0
    div _divisor

    push _divisor
    push eax
    call divide

    done:
        ret
divide endp
end main
```

QUESTION 2

```asm
include irvine32.inc

.data
arr dword 2, 5, 7, 8, 3, 10, 19, 1
m1 byte "Enter number to find.", 0
m2 byte "Found", 0
m3 byte "Not found", 0
```

```
.code

find proto,
    val: dword,
    i: dword,
    len: dword


.code
    main proc
        mov edx, offset m1
        call writestring
        call readdec
        invoke find, eax, 0, sizeof arr


        exit
    main endp

    find proc,
        val: dword,
        i: dword,
        len: dword


        mov esi, i
        cmp esi, len
        jl con
```

```
        mov edx, offset m3
        call writestring
        call crlf
        jmp done

con:

        mov eax, arr[esi]
        cmp eax, val
        jne not_found
        mov edx, offset m2
        call writestring
        call crlf
        jmp done

not_found:

        add esi, type arr
        invoke find, val, esi, len


done:
        ret
find endp
end main
```
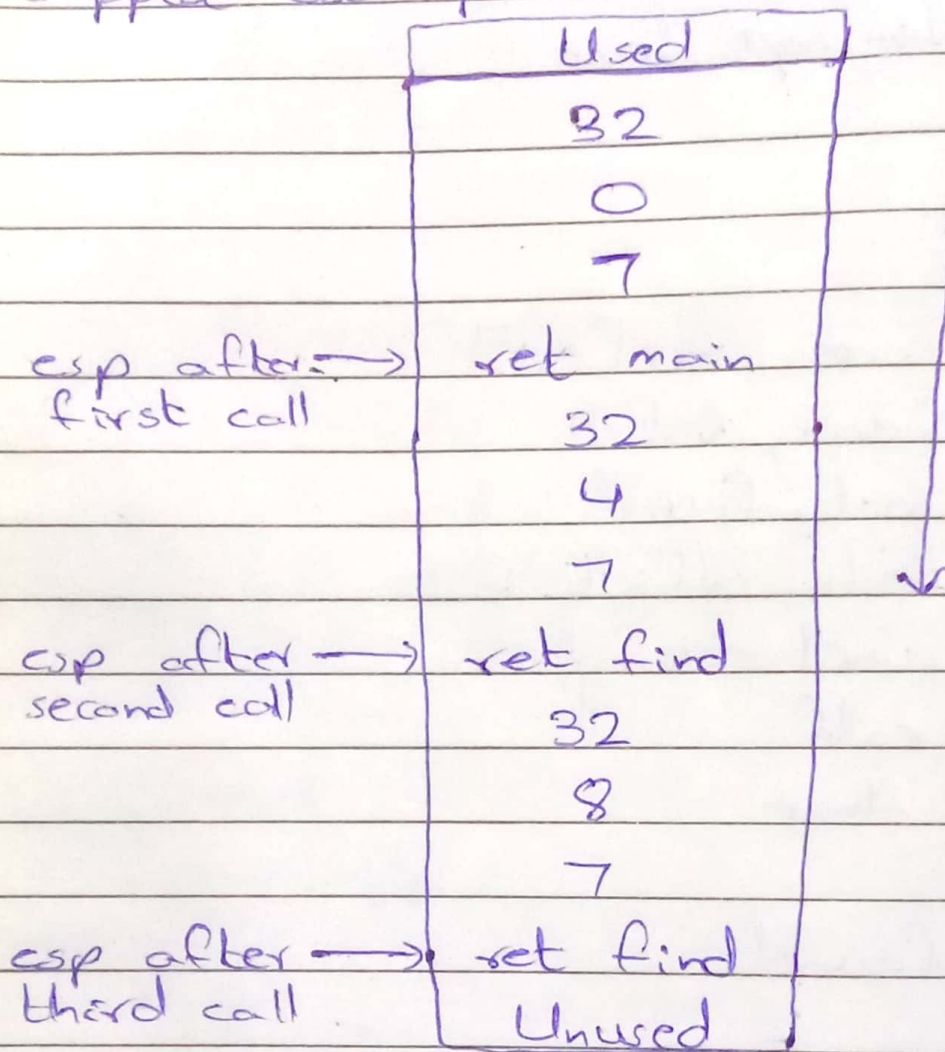
Date:_____

Stack:

Suppose user provide value '7'.

Used
32
0
7

esp after: → ret main
first call

| 32 |
| 4 |
| 7 |

esp after → ret find
second call

| 32 |
| 8 |
| 7 |

esp after → ret find
third call

| Unused |

QUESTION 3

include irvine32.inc

.data
    src byte "This is the source string", 0
    dest byte 100 dup(0)

```
m1 byte "Source: ", 0
m2 byte "Destination: ", 0
lenOfDest byte 0


.code

main proc
    mov edx, offset m1
    call writestring
    mov edx, offset src
    call writestring
    call crlf


    mov ecx, length of src
    mov esi, 0


    L1:
        mov al, src [esi]
        mov edi, offset dest
        push ecx
        movzx ecx, lengthOfDest
        repne scasb
        je next
        mov [edi], al
        inc lenOfDest
```

```
        next:

            pop ecx
            add esi, type src
            loop L4

        mov edx, offset m2
        call writestring
        mov edx, offset dest
        call writestring a.
        call crlf


            exit
        main endp
    end main
```

QUESTION 4

```
include irvine32.inc


.data
    str1 byte "Advanced Programming in
                Unix Environment", 0
    m byte "Vowel Count:", 0
    m1 byte "a or A = ", 0
```

```
m2  byte  " e or E = ", 0
m3  byte  " i or I = ", 0
m4  byte  " o or O = ", 0
m5  byte  " u or U = ", 0
cA  byte  0
cE  byte  0
cI  byte  0
cO  byte  0
cU  byte  0


print proto,
    s: ptr byte,
    cnt: byte


.code

main proc
    mov edx, offset m
    call writestring
    call crlf

    mov ecx, lengthofall str1
    mov esi, 0
```

```asm
L1:
    mov al, str1 [esi]
    or al, 00100000b
    cmp al, 'a'
    je a_case
    cmp al, 'e'
    je e_case
    cmp al, 'i'
    je i_case
    cmp al, 'o'
    je o_case
    cmp al, 'u'
    je u_case
    jmp next

a_case:
    inc cA
    jmp next

e_case:
    inc cE
    jmp next

i_case:
    inc cI
    jmp next
```

```
            o-case:
                inc c0
                jmp next


            u-case:
                inc cu


        next:


            add esi, type str1
        loop L1


    invoke print, addr m1, cA
    invoke print, addr m2, cE
    invoke print, addr m3, cI
    invoke print, addr m4, c0
    invoke print, addr m5, cu


    exit
main endp


print proc,
    s: ptr byte
    cnt: byte
```

```
        mov edx, s
        call writestring
        movzx eax, cnt
        call writedec
        call crlf


        ret
    print endp
end main
```