

COAL Assignment 03

Rayyan Aamir | 24K-0687 | BCS-3F

Question 01:

Code:

```
; Rayyan Aamir | 24K-0687 | BCS-3F
```

```
; Assignment 03 | Question 01
```

```
INCLUDE Irvine32.inc
```

```
.data
```

```
dividend WORD 0D4A4h
```

```
divisor WORD 0Ah
```

```
msg BYTE "The quotient is ",0
```

```
.code
```

```
main PROC
```

```
    movzx eax, dividend
```

```
    movzx ebx, divisor
```

```
    call RecursiveFunction
```

```
    mov edx, OFFSET msg
```

```
    call WriteString
```

```
    call WriteInt
```

```
    call CRLF
```

```
    exit
```

```
main ENDP
```

```
RecursiveFunction PROC
```

```
    cmp eax, 5h
```

```
    jbe return
```

```
    push ebx
```

```
    xor edx, edx
```

```
    div ebx
```

```
    call RecursiveFunction
```

```
    pop ebx
```

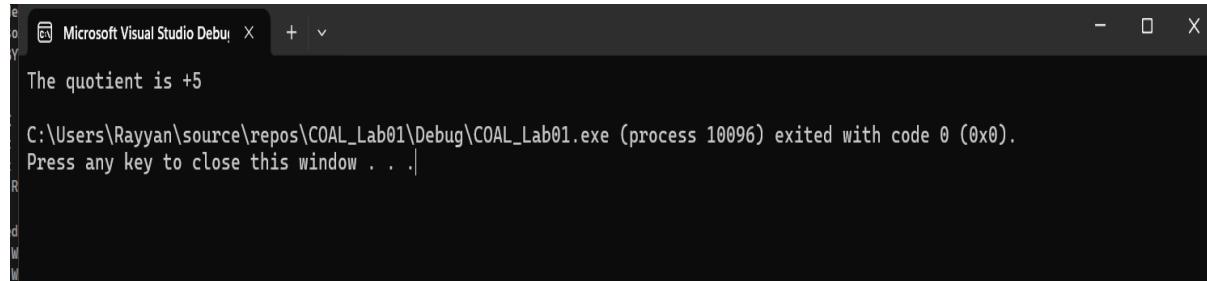
```
return:
```

```
    ret
```

```
RecursiveFunction ENDP
```

```
END main
```

Output:



Question 02:

Code:

```
; Rayyan Aamir | 24K-0687 | BCS-3F  
; Assignment 03 | Question 02
```

```
INCLUDE Irvine32.inc
```

```
.data
```

```
str1 BYTE 50 DUP(?)  
msg1 BYTE "Enter a string: ",0  
msg2 BYTE "Vowel Count",0  
countVowel BYTE 5 DUP(0)  
msg3 BYTE "a or A = ",0  
msg4 BYTE "e or E = ",0  
msg5 BYTE "i or I = ",0  
msg6 BYTE "o or O = ",0  
msg7 BYTE "u or U = ",0
```

```
.code
```

```
main PROC
```

```
    mov edx, OFFSET msg1  
    call WriteString  
    mov edx, OFFSET str1  
    mov ecx, SIZEOF str1  
    call ReadString
```

```
    mov esi, OFFSET str1
```

check_loop:

mov al, [esi]

cmp al, 0

je display

cmp al, 'a'

je inc_A

cmp al, 'A'

je inc_A

cmp al, 'e'

je inc_E

cmp al, 'E'

je inc_E

cmp al, 'i'

je inc_I

cmp al, 'I'

je inc_I

cmp al, 'o'

je inc_O

cmp al, 'O'

je inc_O

cmp al, 'u'

je inc_U

cmp al, 'U'

je inc_U

```
    jmp next
```

```
inc_A:
```

```
    inc BYTE PTR [countVowel + 0]
```

```
    jmp next
```

```
inc_E:
```

```
    inc BYTE PTR [countVowel + 1]
```

```
    jmp next
```

```
inc_I:
```

```
    inc BYTE PTR [countVowel + 2]
```

```
    jmp next
```

```
inc_O:
```

```
    inc BYTE PTR [countVowel + 3]
```

```
    jmp next
```

```
inc_U:
```

```
    inc BYTE PTR [countVowel + 4]
```

```
next:
```

```
    inc esi
```

```
    jmp check_loop
```

```
display:
```

```
    call CRLF
```

```
    mov edx, OFFSET msg2
```

```
    call WriteString
```

call CRLF

mov edx, OFFSET msg3

call WriteString

movzx eax, BYTE PTR [countVowel + 0]

call WriteDec

call CRLF

mov edx, OFFSET msg4

call WriteString

movzx eax, BYTE PTR [countVowel + 1]

call WriteDec

call CRLF

mov edx, OFFSET msg5

call WriteString

movzx eax, BYTE PTR [countVowel + 2]

call WriteDec

call CRLF

mov edx, OFFSET msg6

call WriteString

movzx eax, BYTE PTR [countVowel + 3]

call WriteDec

call CRLF

mov edx, OFFSET msg7

call WriteString

movzx eax, BYTE PTR [countVowel + 4]

```
call WriteDec
```

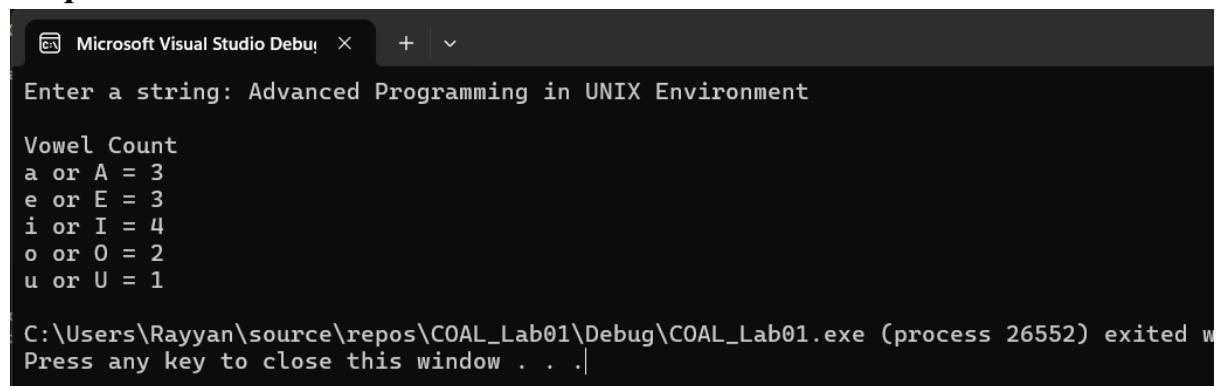
```
call CRLF
```

```
exit
```

```
main ENDP
```

```
END main
```

Output:



A screenshot of a Microsoft Visual Studio Debug window. The title bar says "Microsoft Visual Studio Debug". The window contains the following text:

```
Enter a string: Advanced Programming in UNIX Environment
Vowel Count
a or A = 3
e or E = 3
i or I = 4
o or O = 2
u or U = 1
C:\Users\Rayyan\source\repos\COAL_Lab01\Debug\COAL_Lab01.exe (process 26552) exited w
Press any key to close this window . . .
```

Question 03:

Code:

```
; Rayyan Aamir | 24K-0687 | BCS-3F
```

```
; Assignment 03 | Question 03
```

```
INCLUDE Irvine32.inc
```

```
.data
```

```
str1 BYTE #####FAST", 0
```

```
.code
```

```
STR_TRIMM PROC, AddrStr : PTR BYTE, char2trim : BYTE
```

```
    mov esi, AddrStr
```

```
    mov edi, esi
```

```
mov bl, char2trim
```

```
skip_loop:
```

```
    mov al, [esi]
```

```
    cmp al, 0
```

```
    je string_end
```

```
    cmp al, bl
```

```
    jne copy_rest_char
```

```
    inc esi
```

```
    jmp skip_loop
```

```
copy_rest_char:
```

```
    mov al, [esi]
```

```
    mov [edi], al
```

```
    cmp al, 0
```

```
    je return
```

```
    inc esi
```

```
    inc edi
```

```
    jmp copy_rest_char
```

```
string_end:
```

```
    mov BYTE PTR [edi], 0
```

```
return:
```

```
    ret
```

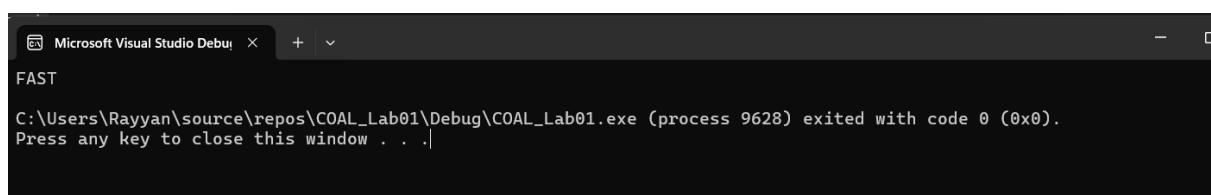
```
STR_TRIMM ENDP
```

```
main PROC
```

```
    INVOKE STR_TRIMM, ADDR str1, '#'
    mov edx, OFFSET str1
    call WriteString
    call CRLF

    exit
main ENDP
END main
```

Output:



Question 04:

Code:

```
; Rayyan Aamir | 24K-0687 | BCS-3F
; Assignment 03 | Question 04
```

```
INCLUDE Irvine32.inc
```

```
.data
```

```
arr1 DWORD 7,3,5,5,5,5,6,9
```

```
s1 DWORD 9
```

```
arr2 DWORD 5,5,5,5,9,8,7
```

```
s2 DWORD 8
```

```
arr3 DWORD 9,8,7,5,5,5,5,5
```

```
s3 DWORD 8
```

```
arr4 DWORD 9,8,7,5,5,5,5
```

```
s4 DWORD 7
```

```
.code
```

```
FindFive PROC, ptrArray : PTR DWORD, arrSize : DWORD
```

```
    push ebx
```

```
    push ecx
```

```
    push esi
```

```
    push edi
```

```
    mov esi, ptrArray
```

```
    mov ecx, arrSize
```

```
    sub ecx, 4
```

```
    jle not_found
```

```
L1:
```

```
    mov edi, 0
```

```
    mov ebx, 0
```

```
check_consecutive:
```

```
    cmp edi, 5
```

```
    je found
```

```
    mov eax, [esi + ebx * 4]
```

```
    cmp eax, 5
```

```
    jne next
```

```
    inc edi
```

```
    inc ebx
```

```
    jmp check_consecutive
```

next:

add esi, 4

loop L1

not_found:

mov eax, 0

jmp return

found:

mov eax, 1

return:

pop edi

pop esi

pop ecx

pop ebx

ret

FindFive ENDP

main PROC

INVOKE FindFive, ADDR arr1, s1

call WriteDec

call CRLF

INVOKE FindFive, ADDR arr2, s2

call WriteDec

call CRLF

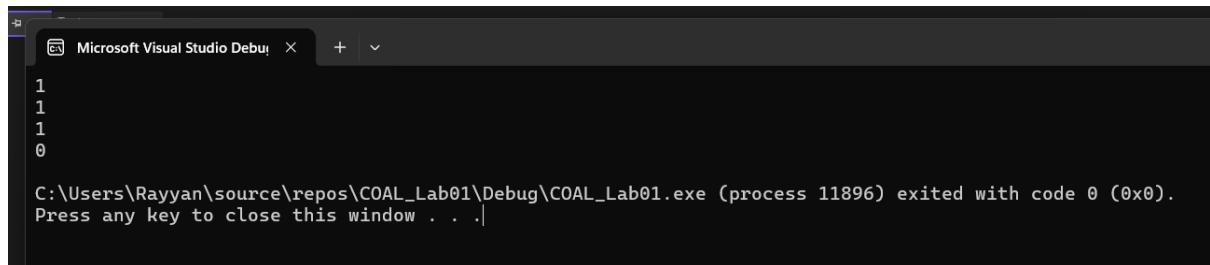
```
   (INVOKE FindFive, ADDR arr3, s3
    call WriteDec
    call CRLF

   (INVOKE FindFive, ADDR arr4, s3
    call WriteDec
    call CRLF

exit
main ENDP

END main
```

Output:



A screenshot of the Microsoft Visual Studio Debug window. The window title is "Microsoft Visual Studio Debug". The output pane displays the following text:
1
1
1
0

C:\Users\Rayyan\source\repos\COAL_Lab01\Debug\COAL_Lab01.exe (process 11896) exited with code 0 (0x0).
Press any key to close this window . . .