23K-0004
BAI-4A

COAL Assignment # 03

Date:

Q1.

```
.data
    dividend DWORD 004A4h
    divisor  DWORD 0Ah
    str1 BYTE "Original dividend", 0
    str2 BYTE "Divisor", 0
    str3 BYTE "Result after recursive division", 0

.code
main PROC
    mov edx, OFFSET str1
    call WriteString
    mov eax, dividend
    call Write Hex

    mov edx, OFFSET str2
    call WriteString
    mov eax, divisor
    call WriteHex

    mov ecx, dividend
    mov ebx, divisor
    call RecussiveDivision

    mov edx, OFFSET str3
    call Writestring
    call WriteHex
    exit
```

```
RecursiveDivision PROC
    cmp eax, 5h
    jle done
    push ebx
    mov edx, 0
    div ebx
    call RecursiveDivision
    pop ebx
done:
    cmp eax, 5h
    jne skip
    mov ecx, 3
skip:
    ret
RecursiveDivision ENDP
END main
```

23K-0004

BAI-4A

Date: _____

Q2.

```
.data
    arr DWORD 10,23,45,66,72,89,100,150,200,300
    arrsize DWORD LENGTHOF arr
    prompt BYTE "Enter a number to search",0
    found BYTE "Value found at index: ",0
    notfound BYTE "Value not found",0

.code
main PROC
    mov edx, OFFSET prompt
    call WriteString
    call ReadInt
    push eax

    mov eax,0
    push eax
    pop ecx

    pop eax
    push eax
    mov esi, OFFSET arr
    push arrsize
    push esi

    INVOKE search, esi, arrsize, eax,ecx
    cmp eax, -1
    je Notfound
    mov edx, OFFSET found
    call WriteString
    call WriteInt
    jmp EndProg

Notfound:
    mov edx, OFFSET notfound
    call WriteString
EndProg:
    exit
main ENDP

search PROC USES ecx edx esi edi,
    ptrArr: PTR DWORD,
    size: DWORD,
    target, DWORD,
    index: DWORD

    mov eax, index
    cmp eax, size
    jae not_found

    mov esi, ptrArr
    mov ecx, index
    mov ebx, 4
    mul ebx
    add esi, ebx
    mov eax, [esi]
```

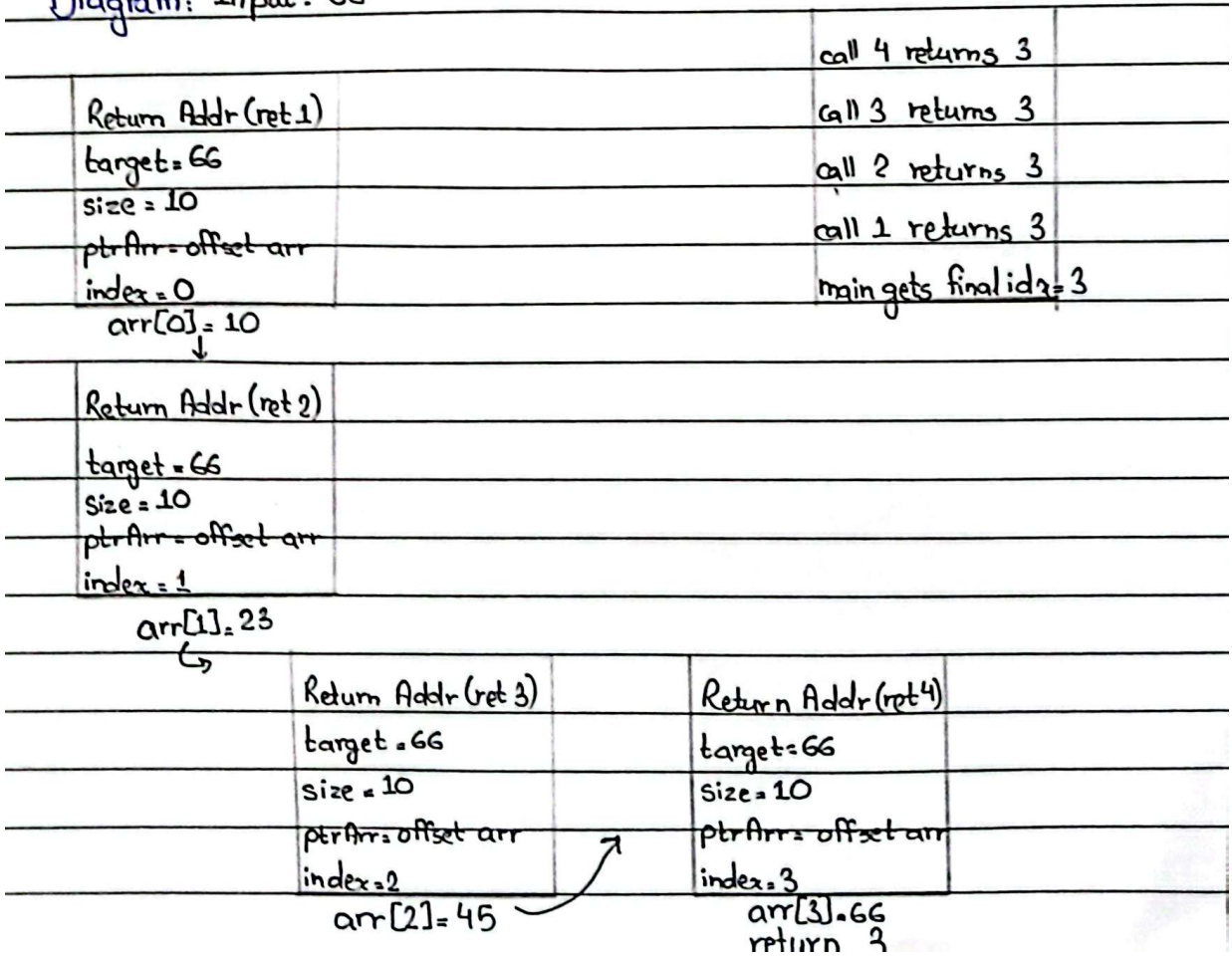
23K-0004

BAI-4A Date: _____

```
cmp eax, target
je found

mov eax, index
inc eax
INVOKE search, ptrArr, size, target, eax
ret
found:
    mov eax, index
    ret
not_found:
    mov eax, -1
    ret
search ENDP
END main
```

Diagram: Input: 66



23K-0004
BAI-4A

Date: _____

Q3

```
.data
sourcestr BYTE "This is the source string",0
targetstr BYTE 100 DUP(0)
length DWORD 0
.code
main PROC
    mov esi, OFFSET sourcestr
    mov edi, OFFSET targetstr
    mov ecx,0
nextchar:
    mov al, [esi]
    cmp al,0
    je done

    push esi
    push ecx
    mov ebx, 0
    mov edx, OFFSET target str
checkDup:
    cmp ebx, ecx
    jge notfound
    mov bl, [edx+ebx]
    cmp bl, al
    je skip
    inc ebx
    jump checkDup
notfound:
    mov [edi], al
    inc edi
    inc ecx

skip:
    pop ecx
    pop esi
    inc esi
    jump nextchar
done:
    mov edx, OFFSET sourcestr
    call WriteString
    mov edx, OFFSET targetstr
    call Writestring

    exit
main ENDP
END main
```

23K-0004
BAI-4A

Date: _____

Q4.

```
.data
    prompt BYTE "Enter a string",0
    inputstr BYTE 100 DUP(0)
    inputsize = SIZEOF inputstr
    countA DWORD 0
    countE DWORD 0
    countI DWORD 0
    countO DWORD 0
    countU DWORD 0
    msgA BYTE "a or A = ",0
    msgE BYTE "e or E = ",0
    msgI BYTE "i or I = ",0
    msgO BYTE "o or O = ",0
    msgU BYTE "u or U = ",0
.code
main PROC
    mov edx, OFFSET prompt
    call WriteString
    mov edx, OFFSET inputstr
    mov ecx, inputsize
    call ReadString
    mov esi, OFFSET inputstr
nextchar:
    mov al, [esi]
    cmp al, 0
    je showResult

    call ToLower

    cmp al, 'a'
    je incA
    cmp al, 'e'
    je incE
    cmp al, 'i'
    je incI
    cmp al, 'o'
    je incO
    cmp al, 'u'
    je incU
    jmp skip
incA:
    inc countA
    jmp skip
incE:
    inc countE
    jmp skip
incI:
    inc countI
    jmp skip
incO:
    inc countO
    jmp skip
incU:
    inc countU
    jmp skip

skip:
    inc esi
    jmp nextchar
```

23K-0004
BAI-4A

Date:

```
showResult:
    mov edx, OFFSET msgA
    call WriteString
    mov eax, countA
    call WriteDec
    call Crlf

    mov edx, OFFSET msgE
    call WriteString
    mov eax, countE
    call WriteDec
    call Crlf

    mov edx, OFFSET msgI
    call WriteString
    mov eax, countI
    call WriteDec
    call Crlf

    mov edx, OFFSET msgO
    call WriteString
    mov eax, countO
    call WriteDec
    call Crlf

    mov edx, OFFSET msgU
    call WriteString
    mov eax, countU
    call WriteDec
    call Crlf
  exit
  main ENDP
END main
```