**Lab 13 (Portal 13)**

**Aamir Sohail**

**18F-0215**

# TASK 13.1

include Irvine32.inc

.data

str1 byte 10 dup(?)

str2 byte "Enter your string: ",0

vowel\_arr byte 10 dup(?)

consonant\_arr byte 10 dup(?)

str3 byte"----------------",0

str4 byte "VOWELS ARE: ",0

str5 byte "CONSONANTS ARE : hllwrld ",0

.code

main proc

mov edx,offset str2

call writestring

call STR\_IN

call SEARCH

call DIS\_STR

exit

main endp

STR\_IN proc

mov edx,offset str1

mov ecx,10

call readstring

ret

STR\_IN endp

SEARCH proc

mov ebx,0

mov ecx,lengthof str1

mov esi,0

l1:

mov bl,[str1+esi]

push bx

inc esi

loop l1

mov esi,0

mov edi,0

mov edx,0

mov ecx,lengthof str1

l2:

pop dx

cmp dx,'A'

jz vowel

cmp dx,'a'

jz vowel

cmp dx,'E'

jz vowel

cmp dx,'e'

jz vowel

cmp dx,'I'

jz vowel

cmp dx,'i'

jz vowel

cmp dx,'O'

jz vowel

cmp dx,'o'

jz vowel

cmp dx,'U'

jz vowel

cmp dx,'u'

jz vowel

jmp consonant

vowel:

mov [vowel\_arr+esi],dl

inc esi

jmp ok

consonant:

mov [consonant\_arr+edi],dl

inc edi

ok:

loop l2

ret

SEARCH endp

DIS\_STR proc

mov edx,offset str3

call writestring

call crlf

mov edx,offset str4

call writestring

mov edx,offset vowel\_arr

call writestring

call crlf

mov edx,offset str5

call writestring

ret

DIS\_STR endp

end

# TASK 13.2

include Irvine32.inc

.data

first BYTE "Enter Number 1: ",0

second BYTE "Enter Number 2: ",0

prime BYTE "Numbers are relatively prime! ",0

not\_prime BYTE "Numbers are not relative prime! ",0

gcd\_found BYTE "GCD IS: ",0

gcd word 0

.code

main proc

call DEC\_IN

call GCD\_AB

call DEC\_OUT

call crlf

call crlf

cmp gcd,1

je prime\_

mov edx, offset not\_prime

call writestring

jmp exit\_here

prime\_:

mov edx, offset prime

call writestring

exit\_here:

exit

main endp

GCD\_AB proc

mov eax,0

cmp bx,dx

je found

mov ax,bx ;greater value is now in ax

jmp find

find:

div dl ;divide ax by dl

cmp ah,0 ;compare remainder with 0

je found1

mov dh,ah

jmp found2

found2:

mov eax,0

mov al,dl

div dh

cmp ah,0 ;compare remainder with 0

je found\_final

mov dl,dh

mov dh,ah

jmp found2

found\_final:

mov eax,0

mov al,dh

mov gcd ,ax

jmp exithere

jmp exithere

found:

mov gcd,bx

jmp exithere

found1:

mov gcd ,dx

jmp exithere

exithere:

ret

GCD\_AB endp

DEC\_IN proc

mov eax,0

mov edx,offset first

call writestring

call readdec

call crlf

mov bx,ax

mov edx,offset second

call writestring

mov eax,0

call readdec

call crlf

mov edx,0

mov dx,ax

cmp bx,dx

jg ok

xchg bx,dx

ok:

call crlf

ret

DEC\_IN endp

DEC\_OUT proc

mov edx,offset gcd\_found

call writestring

mov eax,0

mov ax,gcd

call writedec

ret

DEC\_OUT endp

end