LAB 7

Q1 a)

include irvine32.inc

.data

var dword 15

x dword ?

msgt byte "Yes your condition is true: ",0

msgf byte "No condition is false: ",0

.code

main proc

mov eax,0

mov ebx,0

mov esi,0

mov edi,0

mov ecx,6

mov edx,4

cmp var,ecx

jb next

mov eax,1

mov x,eax

mov edx,offset msgf

call writestring

call writedec

call crlf

exit

next:

cmp ecx,edx

jae thn

mov eax,1

mov x,eax

mov edx,offset msgf

call writestring

call writedec

call crlf

exit

thn:

mov eax,0

mov x,eax

mov edx,offset msgt

call writestring

call writedec

call crlf

exit

main endp

end main 

Q1 b)

include irvine32.inc

.data

var dword 0

x dword 2

str1 byte "Hello",0

str2 byte "World",0

.code

main proc

mov eax,0

mov ebx,0

mov esi,0

mov edi,0

mov ecx,10

mov edx,0

mov ebx,2

top:

mov edx,0

mov eax,var

div ebx

cmp var,ecx

ja next

cmp edx,0

mov edx,offset str1

call writestring

call crlf

inc var

jmp top

next1:

mov edx,offset str2

call writestring

call crlf

inc var

jmp top

next:

call crlf

exit

main endp

end main



Q3

include irvine32.inc

.data

intArr SWORD 0, 0, 0, 0, 1, 20, 35, -12, 66, 4, 0

c1 dword 0

c2 dword 0

str1 byte "Total non zero elements are: ",0

str2 byte "Total zero elements are : ",0

.code

main proc

mov eax,0

mov ebx,0

mov ecx,0

mov edx,0

mov ecx,lengthof intArr

mov esi,0

l1:

mov ax,intArr[esi]

cmp ax,0

je next

inc c1

top:

add esi,2

loop l1

mov edx,offset str1

call writestring

mov eax,c1

call writedec

call crlf

mov edx,offset str2

call writestring

mov eax,c2

call writedec

call crlf

call crlf

exit

next:

inc c2

jmp top

main endp

end main



Q4)

include irvine32.inc

.data

intArr byte 4 dup(?)

c1 dword 0

c2 dword 0

first byte "Enter the first element : ",0

second byte "Enter the second element : ",0

third byte "Enter the third element : ",0

fourth byte "Enter the fourth element : ",0

str1 byte "Yes all four elements are equal: ",0

str2 byte "Not all four elements are equal: ",0

v1 dword ?

v2 dword ?

v3 dword ?

v4 dword ?

.code

main proc

mov eax,0

mov ebx,0

mov ecx,0

mov edx,0

mov edx,offset first

call writestring

call readdec

mov v1,eax

mov edx,offset second

call writestring

call readdec

mov ebx,eax

mov edx,offset third

call writestring

call readdec

mov v3,eax

mov edx,offset fourth

call writestring

call readdec

cmp v1,ebx

je n1

mov edx,offset str2

call writestring

call crlf

exit

n1:

cmp v3,ebx

je n2

mov edx,offset str2

call writestring

call crlf

exit

n2:

cmp v3,eax

je print

mov edx,offset str2

call writestring

call crlf

exit

print:

mov edx,offset str1

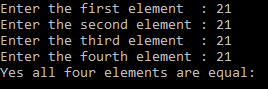
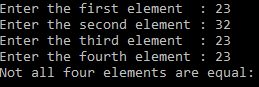
call writestring

call crlf

exit

main endp

end main

Q5)

include irvine32.inc

.data

array dword 10, 4, 7, 14, 299, 156, 3, 19, 29, 300, 20

search byte "Enter the element that you want to search : ",0

str1 byte "Yes element found. ",0

str2 byte "No element does not found. ",0

.code

main proc

mov eax,0

mov ebx,0

mov ecx,0

mov edx,0

mov edx,offset search

call writestring

call readdec

mov ebx,eax

mov ecx,lengthof array

mov esi,0

l1:

mov eax,array[esi]

cmp eax,ebx

je ex

add esi,4

loop l1

mov edx,offset str2

call writestring

call crlf

exit

ex:

mov edx,offset str1

call writestring

call crlf

exit

main endp

end main



Q6)

include irvine32.inc

.data

counter dword 0

v dword ?

list dword 10 dup(?)

print byte "Total swap counter value is: ",0

insert byte "Enter the element: ",0

total byte "How many elements you want to enter: ",0

.code

main proc

mov eax,0

mov ebx,0

mov ecx,0

mov edx,0

mov esi,offset list

mov edi,offset list

add edi,4

mov edx,offset total

call writestring

call readdec

mov v,eax

mov ecx,eax

call crlf

mov ebp,esi

l1:

mov edx, offset insert

call writestring

call readdec

mov [ebp],eax

add ebp,4

loop l1

mov ecx,v

dec ecx

l2:

mov eax,[esi]

mov ebx,[edi]

cmp eax,ebx

jbe ex

mov [esi],ebx

mov [edi],eax

inc counter

top:

add esi,4

add edi,4

loop l2

call crlf

mov edx,offset print

call writestring

mov eax,counter

call writedec

call crlf

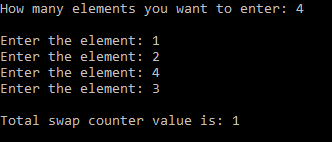
exit

ex:

jmp top

main endp

end main



LAB 8

Q1)

include irvine32.inc

.data

var1 dword ?

var2 dword ?

var4 dword ?

temp dword ?

temp1 dword ?

x dword 5

y dword 3

v1 byte "Enter the value of var1: ",0

v2 byte "Enter the value of var2: ",0

result byte "The value in var4 is : ",0

.code

main proc

mov eax,0

mov ebx,0

mov edx,offset v1

call writestring

call readdec

mov var1,eax

mov edx,offset v2

call writestring

call readdec

mov var2,eax

mov eax,var1

mul x

mov temp,eax

mov eax,var2

sub eax,3

mov temp1,eax

mov eax,temp

div temp1

mov var4,eax

mov edx,offset result

call writestring

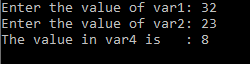
call writedec

call crlf

exit

main endp

end main



Q2)

include irvine32.inc

.data

val1 dword ?

val2 dword ?

val3 dword ?

val4 dword ?

temp dword 5

temp1 dword ?

x dword ?

y dword ?

z dword ?

v2 byte "Enter the value of var2: ",0

v3 byte "Enter the value of var3: ",0

res byte "The value in val1 is : ",0

v4 byte "The value in val4 is : ",0

.code

main proc

mov eax,0

mov ebx,0

mov ecx,0

mov edx,0

mov edx,offset v2

call writestring

call readdec

mov val2,eax

mov edx,offset v3

call writestring

call readdec

mov val3,eax

mov edx,0

mov eax,val2

div val3

mov val1,eax

mov x,eax

mov edx,0

mov eax,val1

div val2

mov y,edx

mov eax,x

mul y

mov edx,offset res

call writestring

call writedec

mov eax,val1

mul temp

mov z,eax

mov eax,val2

mov edx,0

div val3

mov temp1,edx

mov eax,z

mov edx,0

div temp1

mov edx,offset v4

call crlf

call writestring

mov val4,eax

call writedec

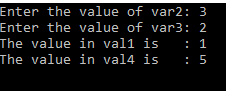
call crlf

caLL crlf

exit

main endp

end main



Q3)

include irvine32.inc

.data

v1 byte ?

v2 byte ?

v3 byte ?

value byte "The value is: ",0

.code

main proc

mov eax,0

mov al,2

shl al,3

mov v1,al

mov al,2

shl al,1

mov v2,al

mov al,v1

add al,v2

mov v3,al

mov al,1

shl al,0

add al,v3

mov edx,offset value

call writestring

call writedec

call crlf

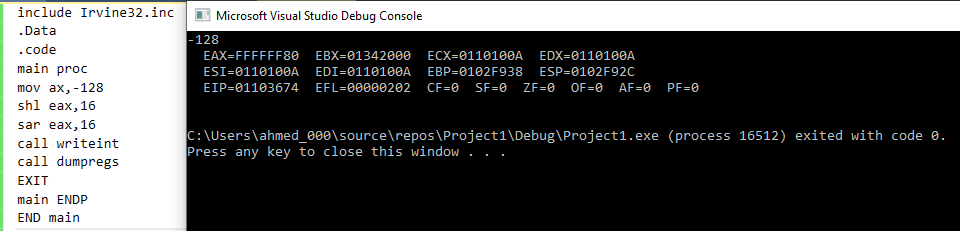
exit

main endp

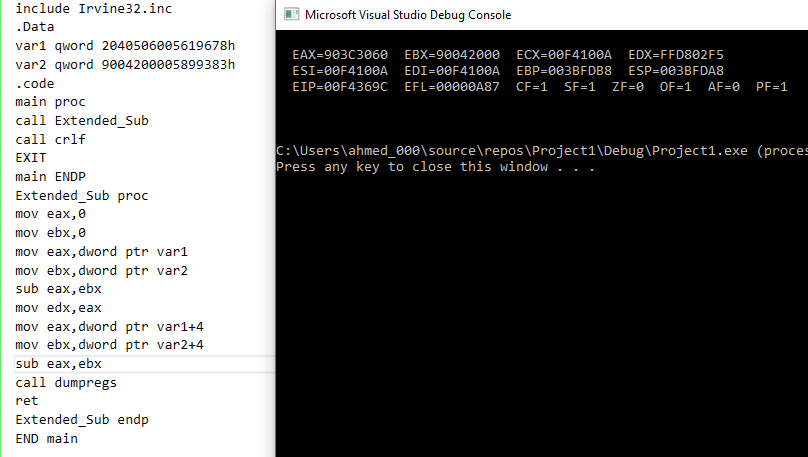
end main



Q4)



Q5)



Q6)

include irvine32.inc

.data

v1 byte "Enter the marks of student: ",0

v2 byte ?

ran byte "Sorry....! Marks are out of range 0->100."

result byte "You got this grade: ",0

.code

main proc

mov edx,0

mov eax,0

mov edx,offset v1

call writestring

call readdec

mov v1,al

cmp al,100

ja exi

cmp v1,90

jb n1

mov edx,offset result

call writestring

mov al,65

call writechar

call crlf

call crlf

exit

n1:

cmp al,80

jb n2

mov edx,offset result

call writestring

mov al,66

call writechar

call crlf

call crlf

exit

n2:

cmp al,70

jb n3

mov edx,offset result

call writestring

mov al,67

call writechar

call crlf

call crlf

exit

n3:

cmp al,60

jb n4

mov edx,offset result

call writestring

mov al,68

call writechar

call crlf

call crlf

exit

n4:

cmp al,0

jb exi

mov edx,offset result

call writestring

mov al,70

call writechar

call crlf

call crlf

exit

exi:

mov edx,offset ran

call writestring

call crlf

call crlf

exit

main endp

end main

end main



Q7)

INCLUDE IRVINE32.inc

.DATA

key BYTE -2,4,1,0,-3,5,2,-4,-4,6

msg BYTE "This plaintext message will be encrypted",0

.CODE

main PROC

mov EAX,0

mov EBX,0

mov ECX,0

mov ESI,0

START:

cmp key[ESI],0

jge L1

mov CL,key[ESI]

rol msg[ESI],CL

inc ESI

cmp ESI,lengthof msg

jz l2

jnz START

L1:

mov CL,key[ESI]

ror msg[ESI],CL

inc ESI

cmp ESI,lengthof msg

jz l2

jnz START

l2:

mov EDX,offset msg

call writestring

call crlf

EXIT

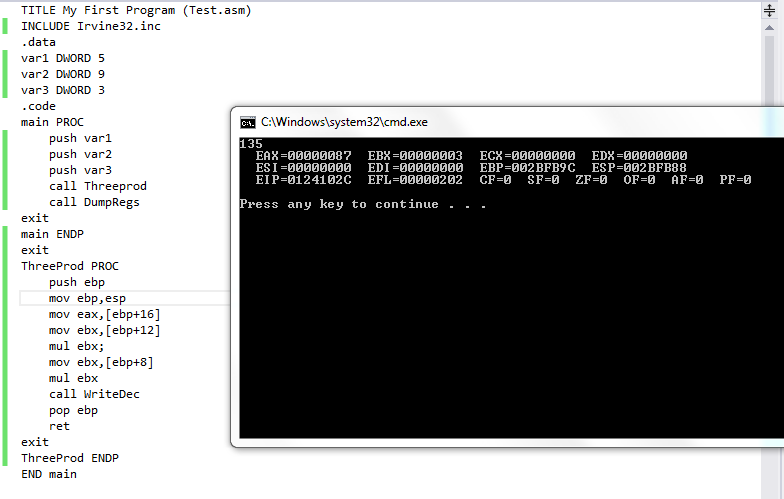
MAIN ENDP

END MAIN



LAB 9

Q1)



Q2)

TITLE My First Program (Test.asm)

INCLUDE Irvine32.inc

.data

array byte 1,2,3,4,100,6,7,8,9

textMax byte "Minimum value: ", 0

textMin byte "Minimum value: ", 0

.code

main PROC

push offset array

call FindMinMax

call DumpRegs

exit

main ENDP

FindMinMax PROC

Enter 0, 1

mov eax, 0

mov ecx, lengthof array

mov esi, 0

mov al, array[esi]

inc esi

findMax:

mov bl, array[esi]

cmp al, bl

jna saveMax

returnMax:

inc esi

loop findMax

jmp endMax

saveMax:

mov al, bl ;greater is going

jmp returnMax

endMax:

mov edx, offset textMin

call writeString

call writeInt

call crlf

mov eax, 0

mov ecx, lengthof array

mov esi, 0

mov al, array[esi]

inc esi

findMin:

mov bl, array[esi]

cmp al, bl

ja saveMin

returnMin:

inc esi

loop findMin

jmp endMin

saveMin:

mov al, bl ;lower is preserved

jmp returnMax

endMin:

mov edx, offset textMax

call writeString

call writeInt

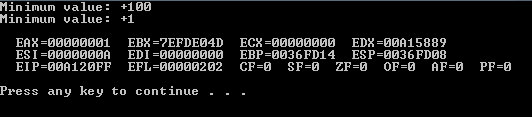
call crlf

leave

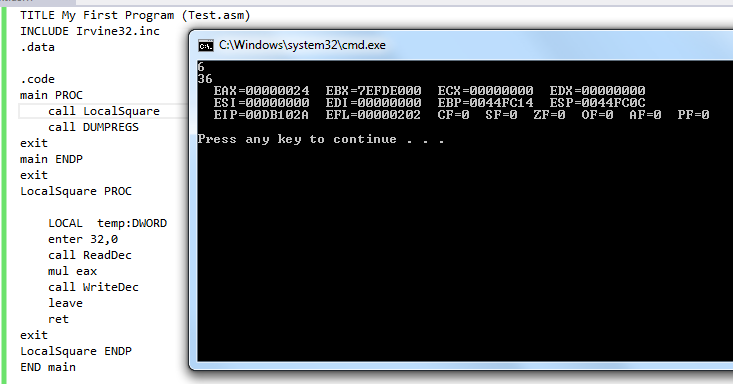
ret

FindMinMax ENDP

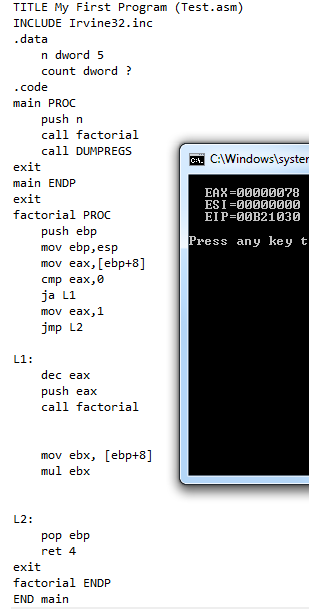
END main



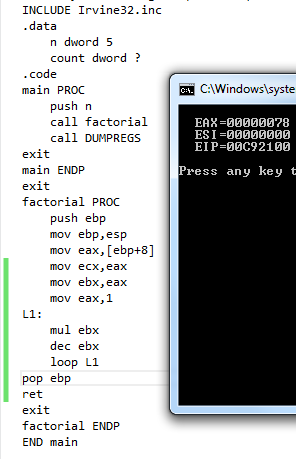
Q3)



Q4)



Q5)



Q6)

Include irvine32.inc

.data

var1 DWORD 4 DUP(?)

var2 byte "Largest is :",0

.code

main PROC

mov ecx,4

mov esi,0

L1:

call readInt

mov var1[esi\*type var1],eax

mov eax,var1[esi\*type var1]

inc esi

loop L1

call checkPrime

exit

main ENDP

checkPrime PROC

push ebp

mov ebp,esp

mov esi,0

mov ecx,4

L2:

mov ebx,ecx

mov ecx,var1[esi]

sub ecx,2

mov edi,2

L1:

mov edx,0

mov eax,var1[esi]

div edi

cmp edx,0

je exits

inc edi

loop L1

add esi,4

mov ecx,ebx

loop L2

call largestPrime

exits:

pop ebp

ret

checkPrime ENDP

largestPrime proc

mov esi,offset var1

mov eax,[esi]

add esi,4

mov ecx, 3

l1:

cmp eax,[esi]

jge exitss

mov eax,[esi]

exitss:

add esi,4

loop l1

mov edx,offset var2

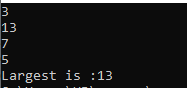
call writestring

call writeDec

ret

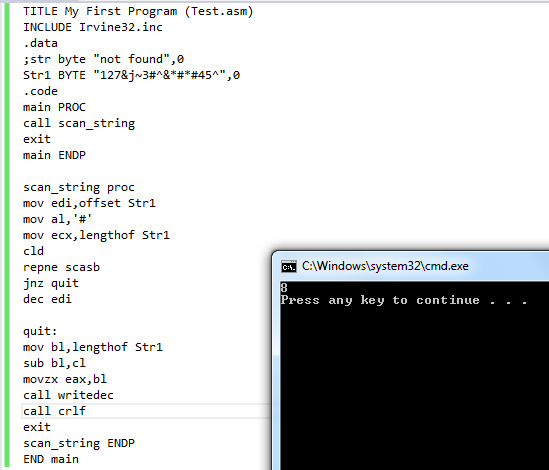
largestPrime endp

END main

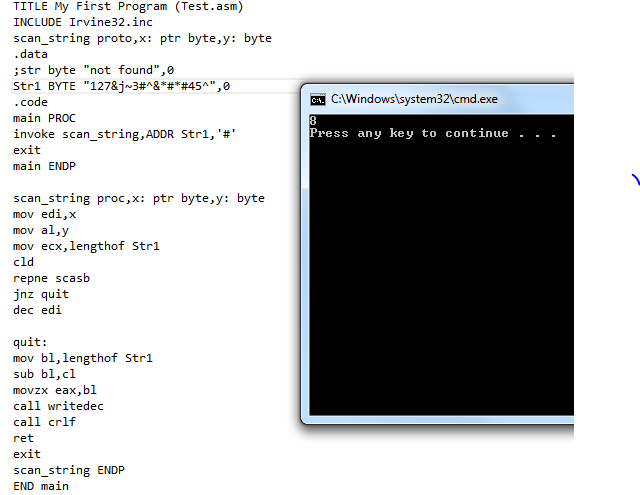


LAB 10

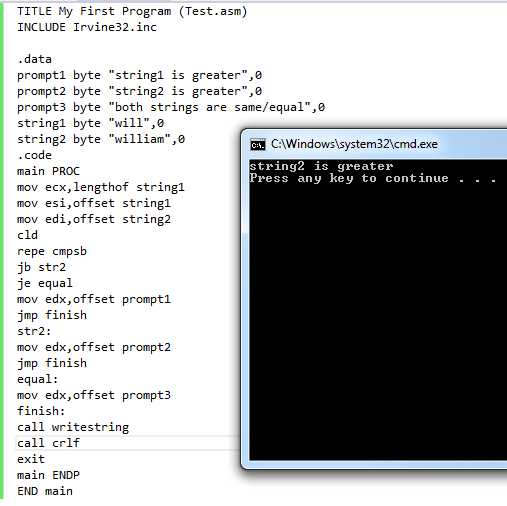
Q1)



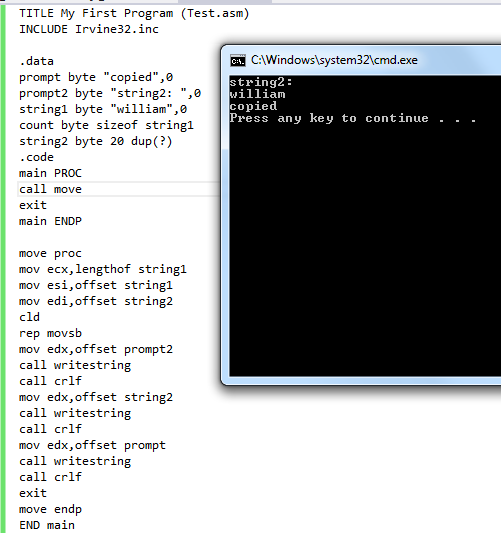
Q2)



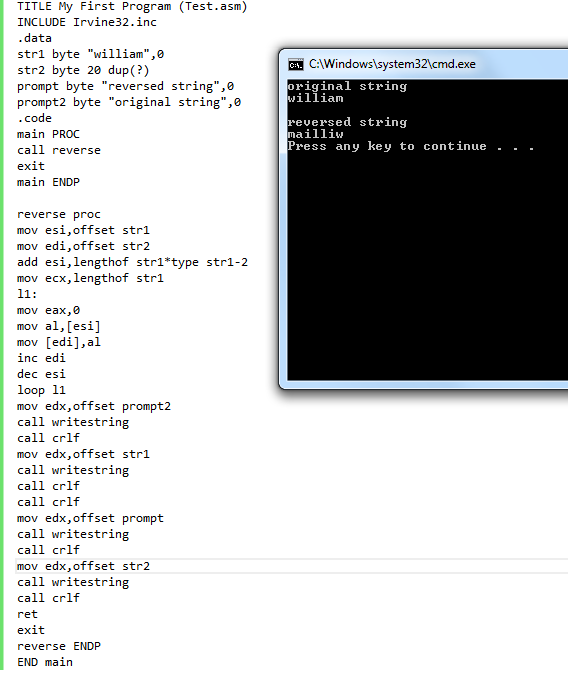
Q3)



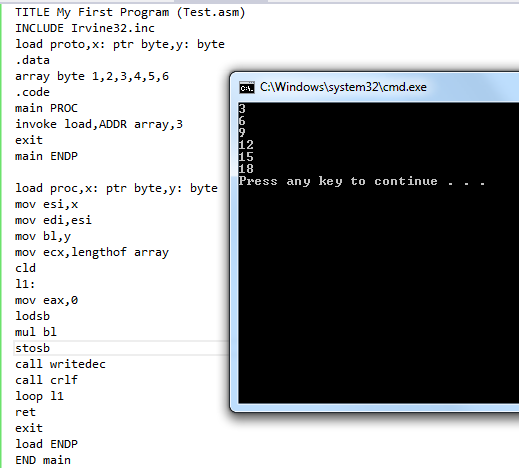
Q4)



Q5)



Q6)



Q7)

INCLUDE Irvine32.inc

.data

target BYTE " AABDCFBBC" , 0

freqTable DWORD 256 DUP( 0)

.code

main PROC

call Clrscr

call Get\_frequencies

call DisplayTable

exit

main ENDP

Get\_frequencies PROC,

pString:PTR BYTE,pTable:PTR DWORD

mov esi,offset target

mov edi,offset freqTable

cld

L1:mov eax,0; clear upper bits of EAX

lodsb

cmp al,0

je Exit\_proc

shl eax,2

inc DWORD PTR[edi+eax]

jmp L1

Exit\_proc:

ret

Get\_frequencies ENDP

DisplayTable PROC

.data

colonStr BYTE ": ",0

.code

call Crlf

mov ecx,LENGTHOF freqTable

mov esi,OFFSET freqTable

mov ebx,0; index counter

L1:

mov eax,[esi]

cmp eax,0

jna L2

mov eax, ebx

call WriteChar

mov edx, OFFSET colonStr

call WriteString

mov eax,[esi]

call WriteDec

call Crlf

L2:

add esi, TYPE freqTable

inc ebx

loop L1

call Crlf

ret

DisplayTable ENDP

END main

