**Abdullah Qadri**

**K17-3922**

**Section: E**

**Coal Lab 8**

**Q1:**

INCLUDE Irvine32.inc

.data

var1 DWORD ?

var2 DWORD ?

var4 DWORD ?

msg1 BYTE "Enter first number: ",0

msg2 BYTE "Enter second number: ",0

msg3 BYTE "Value in Var4 is: ",0

.code

main proc

mov ebx, 0

mov ecx, 0

mov edx, 0

mov eax, 0

mov edx,offset msg1

call WriteString

call ReadDec

mov var1, eax

mov eax, 0

mov edx,offset msg2

call WriteString

call ReadDec

mov var2, eax

mov eax, 0

mov eax, var1

mov ebx, 5

mul ebx

sub var2, 3

div var2

mov edx,offset msg3

call WriteString

call WriteDec

mov var4, eax

call crlf

call WaitMSg

exit

main ENDP

END main

Q2:

INCLUDE Irvine32.inc

.data

val1 SDWORD 15

val2 SDWORD 10

val3 SDWORD 5

var1 SDWORD -2

var2 SDWORD -12

var3 SDWORD 5

var4 SDWORD ?

msg1 BYTE "Value in val1 is: ",0

msg2 BYTE "Value in var4 is: ",0

.code

main proc

mov eax, 0

mov ebx, 0

mov ecx, 0

mov edx, 0

;for value of val1

mov eax, val2

mov ebx, val3

idiv ebx

mov ecx, eax

mov eax, val1

mov ebx, val2

idiv ebx

mov eax, edx

mov eax, 0

mov eax, ecx

imul edx

mov val1,eax

mov edx,offset msg1

call WriteString

call WriteDec

mov val1, eax

call crlf

;for value of var4

mov eax, var1

mov ebx, -5

imul ebx

mov ecx, eax

neg var2

mov eax, var2

mov ebx, var3

idiv ebx

mov ebx, 0

mov ebx, edx

mov eax, 0

mov eax, ecx

idiv ebx

mov edx,offset msg2

call WriteString

call WriteDec

mov var4, eax

call crlf

call WaitMSg

exit

main ENDP

END main

Q3:

INCLUDE Irvine32.inc

.data

val1 DWORD ?

msg1 BYTE "Enter value: ",0

msg2 BYTE "Binary Multiplication is: ",0

.code

main proc

mov eax, 0

mov ebx, 0

mov ecx, 0

mov edx, 0

mov edx,offset msg1

call WriteString

call ReadDec

mov val1, eax

mov ebx, eax

mov ecx, eax

shl eax, 4

shl ebx, 2

add eax, ebx

add eax, ecx

mov edx,offset msg2

call WriteString

call WriteDec

call crlf

call WaitMSg

exit

main ENDP

END main

Q4:

INCLUDE Irvine32.inc

.data

var1 sword -128

.code

main proc

mov eax, 0

mov ebx, 0

mov ecx, 0

mov edx, 0

mov ax,var1

call writebin

shl eax, 16

sar eax, 16

call crlf

call writebin

call crlf

call WaitMSg

exit

main ENDP

END main

Q7:

INCLUDE Irvine32.inc

.data

key1 BYTE -2, 4, 1, 0, -3, 5, 2, -4, -4, 6

keySize = $ - key1

plainText1 BYTE "This plaintext message will be encrypted.",0

.code

main PROC

mov edx, OFFSET plainText1

call WriteString

call Crlf

mov esi, OFFSET plainText1

L1:

mov edi, OFFSET plainText1

mov ecx, keySize

call Encrypt

JNZ L1

mov ecx, keySize

call Encrypt

JNZ L1

mov edx, OFFSET plainText1

call WriteString

call Crlf

call WaitMsg

exit

main ENDP

Encrypt PROC

L1:

push ecx ;check for a null byte

cmp BYTE PTR[esi], 0

JE L4

mov cl,[edi]

cmp cl,0

JGE L2

neg cl

rol BYTE PTR[esi], cl

jmp L3

L2:

ror BYTE PTR[esi], cl

L3:

inc esi

inc edi

pop ecx

Loop L1

or eax, 1

jmp L5

L4:

pop ecx

L5:

ret

Encrypt ENDP

END main

Q5:

INCLUDE Irvine32.inc

.data

sss sword -128

ii = 0

.code

main PROC

mov ecx,12

L:

push ecx

mov eax,20

call RandomRange

inc eax

mov esi,eax

mov eax,ecx

call RandomRange

inc eax

mov ecx,eax

call gcd

call writeInt

call crlf

pop ecx

Loop L

call WaitMsg

exit

main ENDP

ABS proc uses esi ecx edx edi ebx

mov ecx,8

mov ebx,0ffffffffh

mov esi,0

mov edi,eax

L:

shrd ebx,esi,4

mov edx,0

shl eax,4

shld edx,eax,4

cmp edx,0fh

loopz L

mov eax,edi

and eax,ebx

ret

ABS endp

GCD proc

mov eax,esi

call abs

mov esi,eax

mov eax,ecx

call ABS

mov ecx,eax

L:

mov edx,0

mov eax,esi

div ecx

mov esi,ecx

mov ecx,edx

inc ecx

loop L

mov eax,esi

ret

GCD endp

END main

Q6:

include irvine32.inc

.data

.code

main proc

mov eax , 1

mov edx , 7

mov ecx , 0

mov ebx , 2

call extended\_sbb

mov esi, eax

mov eax, edx

call WriteDec

mov eax, esi

call crlf

call WriteDec

call crlf

call WaitMsg

exit

main endp

extended\_sbb proc

sub eax, ebx

sbb edx, 0

sub edx, ecx

ret

extended\_sbb endp

end main