Q1.

Include Irvine32.inc

Include Macros.inc

.data

arr SDWORD 30, -40, 20, 65, 80, 45

j SDWORD ?

k SDWORD ?

.code

MAIN PROC

mov ecx,LENGTHOF arr

mov ebx,0

mov j, 20

mov k, 50

push OFFSET arr

push SIZEOF arr

push j

push k

call arraySum

call printResults

mov j, 35

mov k, 90

push OFFSET arr

push SIZEOF arr

push j

push k

call arraySum

call printResults

exit

MAIN ENDP

printResults PROC

mWrite "The sum of all the elements in the range: "

call WriteInt

call crlf

ret

printResults ENDP

arraySum PROC uses EBX ECX EDX ESI

local first:SDWORD,last:SDWORD,sizeArray:DWORD

mov esi,[ebp + 20]

mov eax,[ebp + 16]

mov sizeArray,eax

mov eax,[ebp + 12]

mov first,eax

mov eax,[ebp + 8]

mov last,eax

mov eax,0

mov edx,0

mov ecx,sizeArray

sumInRange:

mov ebx,[esi + edx \* 4]

cmp ebx,first

jge checkIfWithinRange

jmp continueLoop

checkIfWithinRange:

cmp ebx,last

jle addIt

jmp continueLoop

addIt:

add eax,ebx

continueLoop:

;inc ecx

sub ecx,4

inc edx

loop sumInRange

ret 16

arraySum ENDP

END MAIN

Q2.

TITLE 20K-0208-Q2

Include Irvine32.inc

Include Macros.inc

.data

arr DWORD 60, 4, 17, 45, 7

.code

MAIN PROC

mov ecx,LENGTHOF arr

mov ebx,0

call selectionSort

mov ecx,LENGTHOF arr

mov ebx,0

printSortedArray:

mov eax,[arr + ebx \* 4]

call WriteDec

mWrite " "

inc ebx

loop printSortedArray

exit

MAIN ENDP

SWAP PROC

push ebp

mov ebp,esp

mov edx,[ebp + 8]

push edx

mov eax,[arr + edx \* 4]

mov edx,[ebp + 12]

xchg eax,[arr + edx \* 4]

pop edx

mov [arr + edx \* 4],eax

pop ebp

ret 8

SWAP ENDP

selectionSort PROC

LOCAL largest:DWORD,i:DWORD,j:DWORD

mov ecx,LENGTHOF arr

mov largest,0

dec ecx

mov i,ecx

mov j,ecx

outerLoop:

mov ebx,i

mov largest,ebx

push ecx

mov edx,i

mov j,edx

innerLoop:

dec j

mov edx,j

mov eax,[arr + edx \* 4]

mov edx,largest

mov ebx,[arr + edx \* 4]

cmp eax,ebx

jg markNewMax

jmp continueLoop

markNewMax:

mov edx,j

mov largest,edx

continueLoop:

loop innerLoop

push i

push largest

call SWAP

pop ecx

dec i

loop outerLoop

ret

selectionSort ENDP

END MAIN

Q3.

TITLE 20K-0208-Q3

Include Irvine32.inc

Include Macros.inc

.data

arr BYTE 10 DUP(?)

.code

MAIN PROC

mov ecx,LENGTHOF arr

mov ebx,0

getInput:

mWrite "Enter value # "

mov eax,ebx

inc eax

call WriteDec

mWrite " : "

call ReadInt

mov [arr + ebx],al

inc ebx

loop getInput

mov esi,OFFSET arr

mov ebx,LENGTHOF arr

call bubbleSort

mov ebx,0

mov ecx,LENGTHOF arr

printArray:

mov al,[arr + ebx]

call WriteDec

inc ebx

mWrite " "

loop printArray

exit

MAIN ENDP

bubbleSort PROC

mov edi,esi

mov ecx,ebx

dec ecx

mov ebx,0

mov eax,0

outerLoop:

push ecx

mov esi,edi

innerLoop:

mov al,[esi]

mov bl,[esi + 1]

cmp al,bl

jg swapElements

continueLoop:

inc esi

loop innerLoop

pop ecx

loop outerLoop

jmp endProgram

swapElements:

mov al,[esi]

mov bl,[esi + 1]

xchg al,bl

mov [esi],al

mov [esi + 1],bl

jmp continueLoop

endProgram:

ret 8

bubbleSort ENDP

END MAIN

Q4.

Include Irvine32.inc

Include Macros.inc

.data

N DWORD ?

.code

MAIN PROC

mWrite "Enter the Number: "

call ReadInt

mov N,eax

call factorial

mWrite "factorial = "

call WriteDec

exit

MAIN ENDP

factorial PROC

mov eax,1

cmp N,0

jle endProgram

mov ecx,N

calculate:

;mov edx,0

mul ecx

loop calculate

endProgram:

ret

factorial ENDP

END MAIN

Q5.

Include Irvine32.inc

Include Macros.inc

.data

character BYTE ?

binaryCode DWORD 00000000b

oneCount DWORD 0

.code

MAIN PROC

mov eax,0

mov edx,0

mWrite " TYPE A CHARACTER : "

call ReadChar

call WriteChar

mov character,al

mov ah,0

movzx ebx,al

mov binaryCode,ebx

mov eax,binaryCode

call countOnes

call crlf

mWrite " THE ASCII CODE OF "

mov al,character

call WriteChar

mWrite " IN BINARY IS: "

mov eax,binaryCode

mov ebx,1

call WriteBinB

call crlf

mWrite " THE NUMBER OF 1 BITS IS: "

mov eax,oneCount

call WriteDec

exit

MAIN ENDP

countOnes PROC

mov eax,binaryCode

mov ebx,2

loopForCount:

mov edx,0

div ebx

cmp edx,1

jz incrementCount

cmp eax,0

jz endLoop

jmp continueLoop

incrementCount:

inc oneCount

continueLoop:

loop loopForCount

endLoop:

ret

countOnes ENDP

END MAIN

Q6.

Include Irvine32.inc

Include Macros.inc

countMatches PROTO,

Parr1:PTR SDWORD,

Parr2:PTR SDWORD,

lengthArr:DWORD

.data

arr1 SDWORD 1,2,3,4,5,6

arr2 SDWORD 2,1,3,4,6,5

.code

MAIN PROC

INVOKE countMatches,ADDR arr1,ADDR arr2,LENGTHOF arr1

mWrite "The number of common elements is: "

call WriteDec

exit

MAIN ENDP

countMatches PROC uses EBX ECX EDX ESI EDI,

Parr1:PTR SDWORD,

Parr2:PTR SDWORD,

lengthArr:DWORD

mov esi,Parr1

mov edi,Parr2

mov ebx,0

mov eax,0

mov ecx,lengthArr

dec ecx

compareElements:

mov edx,[esi + ebx \* 4]

cmp edx,[edi + ebx \* 4]

jz incrementCount

jmp continueLoop

incrementCount:

inc eax

continueLoop:

inc ebx

loop compareElements

ret

countMatches ENDP

END MAIN

Q7.