Q1)

```
Include Irvine32.inc
.data
arr SDWORD 30,-40,20,65,80,45
j DWORD 20
k DWORD 50
.code
main PROC
mov edx,offset arr
mov ecx,lengthof arr
call sum_r
call writedec
call crlf
mov j,35
mov k,90
mov edx,offset arr
mov ecx,lengthof arr
call sum_r
call writedec
exit
main ENDP

sum_r PROC
push edx
push ecx
mov eax,0
mov ebx,0
L1:
mov ebx,[edx]
cmp ebx,j
jge L2
jmp L_last
L2:
cmp ebx,k
jne L3
jmp L_last
L3:
add eax,ebx
L_last: add edx,4
Loop L1
Pop edx
Pop ecx
Ret
sum_r ENDP
END main
```

Q2)

```
INCLUDE Irvine32.inc
selectionSort PROTO,offarr:DWORD,size1:dword
SWAP PROTO,var3:DWORD,var4:DWORD
.data
arr DWORD 60,4,17,45,7
var1 DWORD ?
var2 DWORD 0
min1 DWORD 0
```

```
.code
main PROC
INVOKE selectionSort,ADDR arr,lengthof arr
exit
main endp

selectionSort PROC,offarr:DWORD,size1:dword
mov esi,offarr
mov ecx,size1
mov ebx,0
dec ecx
_while:
	cmp ebx,ecx
	JE _endwhile
	mov min1,ebx
	mov var1,ebx
	inc var1
	push ecx
	mov ecx,var1
		_while2:
					cmp ecx,size1
					JE _endwhile2
						mov edi,[esi+min1]
						cmp esi,edi
						jl yes
						jmp no
						yes:mov min1,ecx
					mov var2,ebx
					no:INVOKE SWAP,min1,var2
					inc ecx
					jmp _while2
		_endwhile2:
	pop ecx
	inc ebx
	jmp _while
_endwhile:

ret
selectionSort END
SWAP PROC,var3:DWORD,var4:DWORD
mov eax,var3
mov ebx,var4
xchg eax,ebx
mov arr[edi*TYPE arr],eax
inc edi
mov arr[edi*TYPE arr],ebx
dec edi
mov eax,0
mov ebx,0
ret
SWAP endp
end main
```

Q3)

```
Include Irvine32.inc
```

```
.data
arr DWORD 10 DUP (?)
.code
main PROC
mov esi,offset lengthof arr
mov ecx,lengthof arr
L1:
call readint
mov [esi],eax
add esi,4
Loop L1
call bubblesort
mov ecx,lengthof arr
mov esi,offset arr
L2:
mov eax,[esi]
call writedec
call crlf
add esi,4
Loop L2
exit
main ENDP
bubblesort PROC
mov ecx, lengthof arr
lea esi,arr
L3:
push ecx
mov ecx, lengthof arr
L4:
mov ecx,[esi]
cmp [esi+4],eax
jg L5
xchg eax,[esi+4]
mov [esi],eax
L5:
add esi,4
loop L4
pop ecx
Loop L3
ret
bubblesort ENDP
END main
```

Q4)

```
Include Irvine32.inc
.data
prmpt1 BYTE "Enter Num: " ,0
prmpt2 BYTE "Result: ",0
.code
main PROC
mov edx,offset prmpt1
call writestring
call readint
push eax
call factorial
mov edx,offset prmpt2
call writestring
```

```
call writedec
exit
main ENDP
factorial PROC
push ebp
mov ebp,esp
mov eax,[ebp+8]
cmp eax,0
jg L1
mov eax,1
jmp Last
L1:
dec eax
push eax
call factorial
mov ebx,[ebp+8]
mul ebx
Last:
pop ebp
ret 4
factorial endp
end main
```

Q5)

```
INCLUDE Irvine32.inc
.data
prmpt1 BYTE "Type A Character: ",0
prmpt2 BYTE "The ASCII Code In Binary Is: ",0
prmpt3 BYTE "The Number Of 1 Bits Is\Are: ",0
.code
main proc
mov edx,OFFSET prmpt1
call writestring
mov eax,0
call readchar
call crlf
mov dl,00000000b
and ah,dl
mov edx,OFFSET prmpt2
call writestring
call writebin
call crlf
mov ebx,0
mov ecx,8
L1:
       shr al,1
       jc yes
       jmp no
       yes:inc ebx
       no:loop L1
mov eax,ebx
mov edx,OFFSET prmpt3
call writestring
call writedec
exit
main endp
```

```
end Main
```

Q6)

```
Include Irvine32.inc
countMatches PROTO, offarr1:PTR SDWORD,offarr2:PTR SDWORD,size1:DWORD
.data
arr1 SDWORD -1,2,-3,4
arr2 SDWORD -1,-2,3,4
prmpt BYTE "Count:  ",0
count DWORD 0
.code
main PROC
INVOKE countMatches,ADDR arr1,ADDR arr2,lengthof arr1
mov edx,offset prmpt
call writestring
call writedec
exit
main ENDP
countMatches PROC,offarr1:PTR SDWORD,offarr2:PTR SDWORD,size1:DWORD
mov eax,0
mov esi,offarr1
mov edi,offarr2
mov ecx,lengthof arr1
L1: mov ebx,[esi]
mov edx,[edi]
cmp ebx,edx
jz yes
jnz no
yes:inc count
no:add esi,4
add edi,4
Loop L1
mov eax,count
ret
countMatches ENDP
end main
```

Q7)

```
INCLUDE Irvine32.inc
.data
array1 TBYTE 6189700196426013744956 21,1000700196426013744956 21,1000700133331374419562 1
array2 TBYTE 6189600196426013744956 21,1000700186426013744956 21,1000700132331374419562 1

.code
main proc
call Extended_Sub
exit
main endp
Extended_Sub proc
mov ecx,3
lea edi,array1
lea esi,array2
```

```
L1:
	mov eax,[edi]
	sub eax,[esi]
	call writedec
	call crlf
	add esi,4
loop L1
ret
Extended_Sub endp
end Main
```

Q8)

```
INCLUDE Irvine32.inc
.data
array1 TBYTE 618970019642601374495621,100070019642601374495621,100070013333313744195621
array2 TBYTE 618970019642601374495621,100070019642601374495621,100070013333313744195621

.code
main proc
call Extended_Add
exit
main endp
Extended_Add proc
mov ecx,3
lea edi,array1
lea esi,array2
L1:
	mov eax,[edi]
	add eax,[esi]
	call writedec
	call crlf
	add esi,4
loop L1
ret
Extended_Add endp
end Main
```

Q9)

```
include Irvine32.inc
find_GCD PROTO,var1:DWORD,var2:DWORD
.data
prmpt BYTE "GCD: ",0
.code
main PROC
mov eax,0
INVOKE find_GCD,5d,20d
mov edx,offset prmpt
call writestring
call writedec
call crlf
INVOKE find_GCD,24d,18d
mov edx,offset prmpt
```

```
call writestring
call writedec
call crlf
INVOKE find_GCD,432d,226d
mov edx,offset prmpt
call writestring
call writedec
exit
main ENDP
find_GCD PROC,var1:DWORD,var2:DWORD
mov eax,var1
mov ebx,var2
cmp eax,0
je ret_b
cmp ebx,0
je ret_a
cmp eax,ebx
jg a_gr_b
mov eax,var1
mov ebx,var2
sub ebx,eax
mov var2,ebx
INVOKE find_GCD,var1,var2
ret
ret_b: mov edx,var2
ret
ret_a: mov edx,var1
ret
a_gr_b:  mov eax,var1
mov ebx, var2
sub eax,ebx
mov var1,eax
INVOKE find_GCD,var1,var2
ret
find_GCD endp
end main
```

Q10)

```
Include Irvine32.inc
countNearMatches PROTO,offarr1:PTR SDWORD,offarr2:PTR SDWORD,size1:DWORD,diff:DWORD
.data
arr1 SDWORD 9,8,7,6,5
arr2 SDWORD 1,2,3,4,5
diffin DWORD ?
count DWORD 0
prmpt BYTE "Enter Difference: " ,0
prmpt2 BYTE "Result: ",0
.code
main PROC
mov edx,offset prmpt
call writestring
call readint
mov diffin,eax
INVOKE CountNearMatches,ADDR arr1,ADDR arr2,lengthof arr1,diffin
mov edx,offset prmpt2
call writestring
```

```
call writedec
exit
main ENDP
CountNearMatches PROC offarr1:PTR SDWORD,offarr2:PTR SDWORD,size1:DWORD,diff:DWORD
mov ebx,0
mov edx,0
mov esi,offarr1
mov edi,offarr2
mov ecx,size1
L1:
mov ebx,[esi]
mov edx,[edi]
cmp ebx,edx
jg L2
mov eax,edx
sub eax,edx
cmp eax,diff
jg counter
jl skip
L2:
mov eax,ebx
sub eax,edx
cmp eax,diff
jg counter
jl skip
counter: inc count
skip: add esi,4
add edi,4
loop L1
mov eax,count
ret
CountNearMatches ENDP
END main
```