**Q1)**

Include irvine32.inc

.data

Str1 BYTE "127&j~3#^&\*#\*#45^",0

msg BYTE "index of the first occurence: ",0

.code

main PROC

mov al,'#'

mov ecx,LENGTHOF Str1

mov edi,offset Str1

cld

repne scasb

sub edi,1

mov esi,offset Str1

sub edi,esi

mov eax,edi

mov edx,offset msg

call writeString

call writedec

call crlf

main ENDP

exit

END main

**Q2)**

Include irvine32.inc

.data

Str1 BYTE "127&j~3#^&\*#\*#45^",0

var BYTE ?

prompt BYTE "Enter a character to search:",0

msg BYTE "index of the first occurence: ",0

.code

main PROC

mov ecx,3

mov eax,0

mov edx,offset var

call readstring

movzx eax,var

push eax

call scan\_char

main ENDP

scan\_char PROC

push ebp

mov ebp,esp

mov al,BYTE PTR [ebp+8]

mov ecx,LENGTHOF Str1

mov edi,offset Str1

cld

repne scasb

sub edi,1

mov esi,offset Str1

sub edi,esi

mov eax,edi

mov edx,offset msg

call writeString

call writedec

call crlf

scan\_char ENDP

exit

END main

**Q3)**

Include irvine32.inc

.data

var1 BYTE "Nashit",0

var2 BYTE "Nashit",0

msg1 BYTE "Strings are equal",0

msg2 BYTE "Strings are not equal",0

.code

main PROC

mov eax,offset var1

mov ebx,offset var2

push eax

push ebx

call Is\_equal

jnz noteq

mov edx,offset msg1

call writestring

jmp quit

noteq:

mov edx,offset msg2

call writestring

quit:

call crlf

exit

main ENDP

Is\_Equal PROC

push ebp

mov ebp,esp

mov esi,[ebp+8]

mov edi,[ebp+12]

L1:

mov al,[esi]

mov dl,[edi]

cmp al,0

jne L2

cmp dl,0

jne L2

jmp L3

L2:

add edi,1

add esi,1

cmp al,dl

je L1

L3:

pop ebp

ret 8

Is\_equal endp

END Main

exit

END main

**Q4)**

Include irvine32.inc

.data

var1 BYTE "nashit",0

var2 BYTE ?

msg BYTE "Reversed String: ",0

.code

main PROC

mov ecx,LENGTHOF var1

mov eax,offset var1

push eax

mov edx,offset msg

call writestring

call crlf

call reverse\_string

main ENDP

exit

reverse\_string proc

push ebp

mov ebp,esp

mov esi,[ebp+8]

sub ecx,2

push ecx

L1:

inc esi

loop L1

pop ecx

inc ecx

mov edi,offset var2

L2:

mov al,[esi]

mov [edi],al

sub esi,1

add edi,1

LOOP L2

mov [edi] ,al

mov edx,offset var2

pop ebp

ret 4

reverse\_string ENDP

exit

END main

**Q5)**

Include irvine32.inc

.data

array DWORD 2,5,6,7,2,4

var DWORD 3

.code

main PROC

call multiply

mov ecx,lengthof array

mov esi,offset array

L2:

mov eax,[esi]

call writedec

call crlf

add esi,4

loop l2

exit

main endp

multiply proc

cld

mov esi,offset array

mov edi,esi

mov ecx,Lengthof array

L1:lodsd

mul var

stosd

loop L1

ret

multiply endp

END main

Q6)

Include irvine32.inc

.data

array DWORD 2,5,6,7,2,4

var DWORD 3

.code

main PROC

call multiply

mov ecx,lengthof array

mov esi,offset array

L2:

mov eax,[esi]

call writedec

call crlf

add esi,4

loop l2

exit

main endp

multiply proc

cld

mov esi,offset array

mov edi,esi

mov ecx,Lengthof array

L1:lodsd

mul var

stosd

loop L1

ret

multiply endp

END main