Task 1

INCLUDE Irvine32.inc

.data

str1 BYTE "127&j~3#^&\*#\*#45^",0

output BYTE " # Location: ",0

char BYTE '#'

.code

main PROC

mov edx, OFFSET output

call WriteString

call find

jz off

call WriteDec

off:

exit

main ENDP

find PROC uses esi edi ecx

mov esi,OFFSET str1

mov edi,esi

mov al,char

mov ecx,LengthOf str1

cld

repne scasb

jnz NotFound

dec edi

sub edi,OFFSET str1

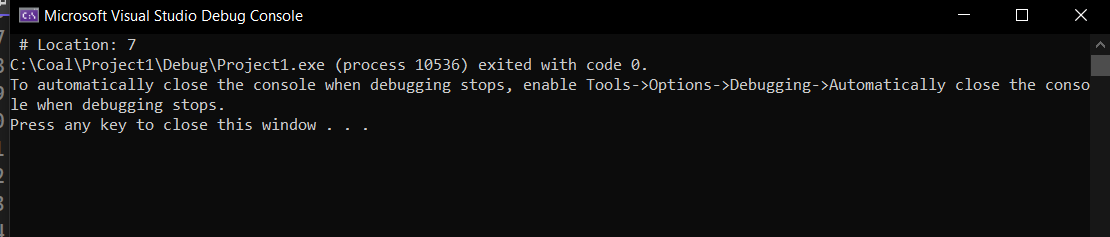
mov eax,edi

NotFound:

ret

find ENDP

END main



Task 5

INCLUDE Irvine32.inc

.data

arr dword 12 dup(?)

row dword 3

col dword 4

.code

main PROC

mov esi,0

mov ecx,row

li:

push ecx

mov ecx,col

l2:

mov arr[esi],0

add esi,4

loop l2

pop ecx

loop li

mov esi,0

mov ecx,row

l1:

push ecx

mov ecx, col

l3:

mov eax,arr[esi]

call writedec

add esi,4

loop l3

pop ecx

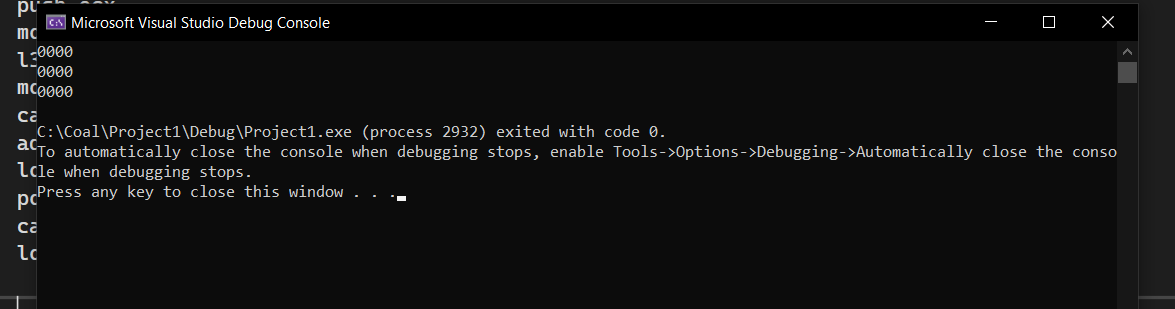
call crlf

loop l1

exit

main ENDP

END main



Task 3

INCLUDE Irvine32.inc

.data

str1 BYTE 'partham',0

str2 BYTE 'kumar',0

no BYTE 'Strings are equal',0

yes BYTE 'Strings are not equal',0

.code

main PROC

mov eax, OFFSET str1

mov ebx, OFFSET str2

call Compare

exit

main ENDP

Compare PROC USES eax ebx, string1:BYTE, string2:BYTE

INVOKE Str\_Compare, ADDR string1, ADDR string2

JA notEqual

JB notEqual

JE Equal

ret

notEqual:

mov edx, OFFSET yes

call WriteString

ret

Equal:

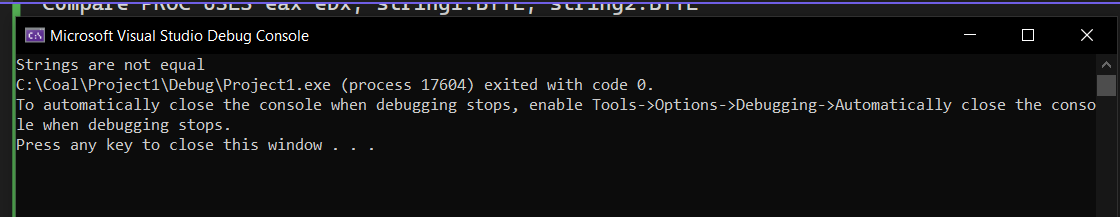
mov edx, OFFSET no

call WriteString

ret

Compare ENDP

END main



Task 4

INCLUDE Irvine32.inc

.data

array BYTE "partham", 0

.code

main PROC

mov esi, OFFSET array

mov ecx, lengthof array - 1

call Reverse

call Crlf

exit

main ENDP

Reverse PROC

mov esi, OFFSET array

reverseLoop:

mov al, [esi + ecx - 1]

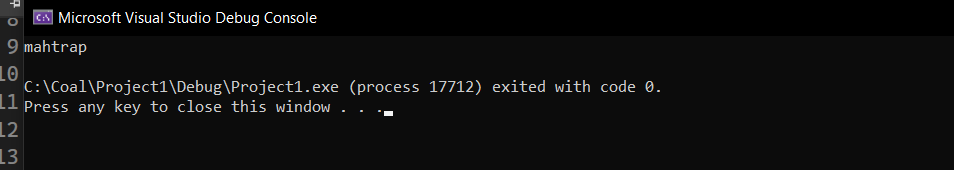
call WriteChar

loop reverseLoop

ret

Reverse ENDP

END main

****