**Task 01:**

;code for reversing an array  
.model small  
.stack 100h  
.data  
arr1 db 0, 1, 2, 3,4  
arr2 db 5 dup(?)  
.code  
reverse proc  
push bp  
mov bp,sp  
add bp,4  
mov si,[bp]  
add si,sizeof arr1  
sub si,1  
mov di,offset arr2  
mov cx,5  
l1:  
mov al,[si]  
mov [di],al  
add di,1  
sub si,1  
loop l1  
pop bp  
ret 2  
reverse endp  
     
main proc  
mov ax, @data  
mov ds, ax  
mov si,OFFSET arr1  
push si  
call reverse  
mov si,OFFSET arr2  
mov cx,5  
l2:  
mov dl,[si]  
add dl,30h  
mov ah,02h  
int 21h  
add si,1  
loop l2  
  
mov ah,4ch  
int 21h  
  
main endp  
end main

**Task 02:**  
  
  
;display the sum of squares of numbers using the procedure  
  
  
.model small  
  
.stack 100h  
  
.data  
msg db "The sum of sqaue is : $"  
var1 dw 1  
var2 dw 2  
var3 dw 1  
  
result dw ?  
  
.code  
  
square proc  
mov ax,var1  
mov bx,var1  
mul bx  
mov cx,ax  
  
mov ax,var2  
mov bx,var2  
mul bx  
  
add cx,ax  
mov ax,cx  
mov bl,10  ; to get correct answer  
div bl  
mov cl,ah  
mov ch,al  
mov ax,var3  
mov bx,var3  
mul bx  
AAM  
add cx,ax  
AAA  
  
  
  
PUSH cx  
POP cx  
  
  
  
ret  
  
square endp  
  
main proc  
  
mov ax, @data  
  
mov ds, ax  
  
mov dx,OFFSET msg  
mov ah,09  
int 21h  
  
  
  
  
call square  
  
mov result,cx  
  
;display the result  
 ; Display the result using AAM  
    mov dl, ch       ; Move the tens place to DL  
    add dl, '0'      ; Convert to ASCII  
    mov ah, 02h      ; Print character function  
    int 21h  
  
    mov dl, cl       ; Move the ones place to DL  
    add dl, '0'      ; Convert to ASCII  
    mov ah, 02h      ; Print character function  
    int 21h  
  
mov ah,4ch  
  
int 21h  
  
main endp  
  
end main

**Task 03:**

.model small

.stack 100h

.data

arr db "thisisalowercasestring$"

.code

main proc

mov ax, @data

mov ds, ax

mov si, offset arr

mov cx, lengthof arr

mov dx, offset arr

mov ah, 09

int 21h

mov dl, 0aH

mov ah, 02

int 21h

mov si, offset arr

mov cx, lengthof arr

L1:

mov al, [si]

sub al, 32

mov [si], al

inc si

loop L1

; adding string terminator symbol

dec si

mov [si], word ptr 36

mov dx, offset arr

mov ah, 09

int 21h

mov ah, 4ch

int 21h

main endp

end main