.model small

.stack 100h

.data

InputStrMsg db "Input string:", 13, 10, '$'

InputSubstrMsg db 0dh, 0ah, "Enter the substring you want to delete:", 13, 10, '$'

InputNewSubstrMsg db 13, 10, "Enter new substring: $"

ResultMsg db 13, 10, "Result: $"

LengthErrorMsg db 13, 10, "Length(word) > Length(str)$"

string db 200 dup("$")

sbstrToRemove db 200 dup("$")

sbstrToInsert db 200 dup("$")

capacity EQU 200

flag dw 0

.code

main proc

mov ax, @data

mov ds, ax

mov es, ax

lea dx, InputStrMsg

call print

lea dx, string

call scan

lea dx, InputSubstrMsg

call print

lea dx, sbstrToRemove

call scan

xor cx, cx ; cx = 0

mov cl, string[1] ; cx = c high + c low

sub cl, sbstrToRemove[1]

js LengthError

lea dx, InputNewSubstrMsg

call print

lea dx, sbstrToInsert

call scan

mov ah, capacity

mov string[0], ah

mov sbstrToRemove[0], ah

mov sbstrToInsert[0], ah

xor cx, cx

mov cl, string[1]

sub cl, sbstrToRemove[1]

jb Return

inc cl

cld

lea si, string[2]

lea di, sbstrToRemove[2]

call ReplaceSubstring

jmp Return

LengthError:

mov dx, offset LengthErrorMsg

call print

Return:

lea dx, ResultMsg

call print

lea dx, string[2]

call print

mov ah, 4ch

int 21h

ret

main endp

; \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ReplaceSubstring proc

StrLoop:

mov flag, 1

push si

push di

push cx

mov bx, si

xor cx, cx

mov cl, sbstrToRemove[1]

repe cmpsb

je FOUND

jne NOT\_FOUND

FOUND:

call DeleteSubstring

mov ax, bx

call InsertSubstring

mov dl, sbstrToInsert[1]

add string[1], dl

mov flag, dx

NOT\_FOUND:

pop cx

pop di

pop si

add si, flag

Loop StrLoop

ret

ReplaceSubstring endp

DeleteSubstring proc

push si

push di

mov cl, string[1]

mov di, bx

repe movsb

pop di

pop si

ret

DeleteSubstring endp

; \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

InsertSubstring proc

lea cx, string[2]

add cl, string[1]

mov si, cx

dec si

mov bx, si

add bl, sbstrToInsert[1]

mov di, bx

mov dx, ax

sub cx, dx

std

repe movsb

lea si, sbstrToInsert[2]

mov di, ax

xor cx, cx

mov cl, sbstrToInsert[1]

cld

repe movsb

ret

InsertSubstring endp

; \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

scan proc

mov ah, 0Ah

int 21h

ret

scan endp

; \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

print proc

mov ah, 9

int 21h

ret

print endp

; \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

end main

================================================================================

.model small

.stack 100h

.data

InputStrMsg db "Input string:", 13, 10, '$'

InputSubstrMsg db 0dh, 0ah, "Enter the substring you want to delete:", 13, 10, '$'

;InputNewSubstrMsg db 13, 10, "Enter new substring: $"

ResultMsg db 13, 10, "Result: $"

LengthErrorMsg db 13, 10, "Length(word) > Length(str)$"

string db 200 dup("$")

sbstrToRemove db 200 dup("$")

sbstrToInsert db 200 dup("$")

capacity EQU 200

flag dw 0

.code

main proc

mov ax, @data

mov ds, ax

mov es, ax

lea dx, InputStrMsg

call print

lea dx, string

call scan

lea dx, InputSubstrMsg

call print

lea dx, sbstrToRemove

call scan

xor cx, cx ; cx = 0

mov cl, string[1] ; cx = c high + c low

sub cl, sbstrToRemove[1]

js LengthError

mov ah, capacity

mov string[0], ah

mov sbstrToRemove[0], ah

mov sbstrToInsert[0], ah

xor cx, cx

mov cl, string[1]

sub cl, sbstrToRemove[1]

jb Return

inc cl

cld

lea si, string[2]

lea di, sbstrToRemove[2]

call ExterminateSubstring

jmp Return

LengthError:

mov dx, offset LengthErrorMsg

call print

Return:

lea dx, ResultMsg

call print

lea dx, string[2]

call print

mov ah, 4ch

int 21h

ret

main endp

; \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ExterminateSubstring proc

StrLoop:

mov flag, 1

push si

push di

push cx

mov bx, si

xor cx, cx

mov cl, sbstrToRemove[1]

repe cmpsb

je FOUND

jne NOT\_FOUND

FOUND:

call DeleteSubstring

mov ax, bx

add string[1], dl

mov flag, dx

NOT\_FOUND:

pop cx

pop di

pop si

add si, flag

Loop StrLoop

ret

ExterminateSubstring endp

DeleteSubstring proc

push si

push di

mov cl, string[1]

mov di, bx

repe movsb

pop di

pop si

ret

DeleteSubstring endp

; \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

scan proc

mov ah, 0Ah

int 21h

ret

scan endp

; \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

print proc

mov ah, 9

int 21h

ret

print endp

; \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

end main