

```

; This program has following features.
; It accepts text of small letters and spaces from keyboard and capitalizes
; the first alphabetical character of every word and removes spaces between
; them. Text can be edited using BACKSPACE and DELETE keys.The input string
; should have maximum of 70 characters.We can exit any time.

```

```

mystack segment stack
    db 256 dup(?)
mystack ends

```

```

mydata segment 'data'
    adjust db 20h                                ;To capitalize first letter.
    prompt1 db 13,10,"Enter the text of small letters having$"
    prompt2 db 13,10,"max. 70 characters.Press ENTER when finished: ",13,10,"$"
    prompt3 db 13,10,13,"Required text is here: ",13,10,13,10,"$"
    prompt4 db 13,10,13,"You want to exit(Enter 1) or continue(Enter 0): $"
    stg1 db 72 dup(?)                            ;The input string.
    stg2 db 70 dup(?)                            ;The processed string.
mydata ends

```

```

mycode segment 'code'
    assume cs:mycode,ds:mydata,ss:mystack

```

```

start proc
    mov ax,seg adjust
    mov ds,ax

```

```

ask:

```

```

    mov dx,offset prompt1                        ;Displayes prompt1 and prompt2 using the
    call display                                  ;subroutine dusplay.
    mov dx,offset prompt2
    call display

```

```

    mov dx,offset stg1                            ;In first byte we move the maximum number
    mov stg1,70                                    ;of characters we wantand offset of stg1
    mov ah,0ah                                      ;is moved into dx to call DOS function
    int 21h                                         ;call 0ah to read the string of characters.

```

```

    mov bx,offset stg1                            ;Move offset of stg1 in bx.
    mov si,offset stg2                            ;Move offset of stg2 in si.

```

```

    inc bx                                         ;The second byte of the stg1 contain the
    xor cx,cx                                       ;number of characters actually typed till
    mov cl,[bx]                                     ;ENTER is pressed which is moved to cx.
    inc bx

```

```

again:

```

```

    mov al,[bx]                                    ;This takes a byte from stg1 and
    cmp al,20h                                      ;if character is space it sets value of
    je inner                                        ;adjust to 20h to capitalize next letter.
    sub al,adjust                                    ;Then it takes next byte without moving
    mov adjust,0                                    ;space into processed string(stg2).If it
    mov [si],al                                     ;is not space it sets value of adjust to
    inc bx                                         ;0h so next byte should remain same.It

```

```

        inc si                ;keeps on moving processed characters
        jmp pos              ;into stg2.

inner:
    mov adjust,20h
    inc bx

pos:
    loop again
    mov dl,0dh                ;This appends the end of processed string
    mov [si+1],dl            ;with a carriage return(0dh),line feed
    mov dl,0ah                ;(0ah) and $ so it can be displayed
    mov [si+2],dl            ;by DOS.
    mov dl,$
    mov [si+3],dl

    mov dx,offset prompt3    ;This displays the prompt3,the processd
    call display              ;string and prompt4 using the subroutine.
    mov dx,offset stg2
    call display
    mov dx,offset prompt4
    call display

    mov ah,01h                ;We read the input with echo and if it is
    int 21h                    ;zero (ASCII code 30H) we jump to ask to
    cmp al,30h                ;run the program again.
    je ask

    mov ah,4ch                ;Terminates the process.
    int 21h

    ret

start endp

display proc near            ;This defines the subroutine which is DOS
    mov ah,09h                ;function call 09H.
    int 21h
    ret
display endp

mycode ends
end start

```