

**Submitted by:** Muhammad Raffey

**Submitted to:** Sir Faizan

**Sap ID:** 70153209

**Department:** CS

**Section:** 3E

**Assignment no:** 2

# **Password Strength Checker**

.MODEL SMALL

.STACK 100H

.DATA

prompt db "Enter password (max 20 chars): $"

strongMsg db 13,10,'Password is Strong$'

upperMsg db 13,10,'Add an uppercase letter.$'

lowerMsg db 13,10,'Add a lowercase letter.$'

digitMsg db 13,10,'Add a digit.$'

.CODE

MAIN PROC

MOV AX, @DATA

MOV DS, AX

LEA DX, prompt

MOV AH, 09h

INT 21h

MOV BL, 0

XOR CX, CX

ReadLoop:

MOV AH, 01h

INT 21h

CMP AL, 13

JE DoneReading

CMP AL, 'A'

JL CheckLower

CMP AL, 'Z'

JG CheckLower

OR BL, 00000001b

JMP Next

CheckLower:

CMP AL, 'a'

JL CheckDigit

CMP AL, 'z'

JG CheckDigit

OR BL, 00000010b

JMP Next

CheckDigit:

CMP AL, '0'

JL Next

CMP AL, '9'

JG Next

OR BL, 00000100b

Next:

INC CX

CMP CX, 20

JL ReadLoop

DoneReading:

CMP BL, 7

JE AllGood

TEST BL, 00000001b

JNZ CheckLowerFlag

LEA DX, upperMsg

MOV AH, 09h

INT 21h

CheckLowerFlag:

TEST BL, 00000010b

JNZ CheckDigitFlag

LEA DX, lowerMsg

MOV AH, 09h

INT 21h

CheckDigitFlag:

TEST BL, 00000100b

JNZ EndProgram

LEA DX, digitMsg

MOV AH, 09h

INT 21h

JMP EndProgram

AllGood:

LEA DX, strongMsg

MOV AH, 09h

INT 21h

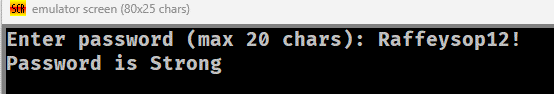
EndProgram:

MOV AH, 4Ch

INT 21h

MAIN ENDP

END MAIN



# **Word Count in a Sentence**

.MODEL SMALL

.STACK 100H

.DATA

msgPrompt db 'Enter a sentence (up to 50 chars): $'

msgResult db 0Dh,0Ah,'Number of words: $'

newLine db 0Dh,0Ah,'$'

userInput db 50 dup (?)

wordCount db 0

.CODE

MAIN PROC

mov ax, @DATA

mov ds, ax

lea dx, msgPrompt

mov ah, 09h

int 21h

mov cx, 50

lea bx, userInput

read\_loop:

mov ah, 01h

int 21h

cmp al, 0Dh

je end\_read

mov [bx], al

inc bx

loop read\_loop

end\_read:

mov al, 0

mov [bx], al

mov wordCount, 0

lea bx, userInput

mov ah, 0

count\_loop:

mov al, [bx]

cmp al, 0

je done\_count

cmp al, ' '

je delimiter

cmp al, ','

je delimiter

cmp al, '.'

je delimiter

cmp ah, 0

jne skip

mov ah, 1

inc wordCount

skip:

jmp move\_next

delimiter:

mov ah, 0

move\_next:

inc bx

jmp count\_loop

done\_count:

lea dx, msgResult

mov ah, 09h

int 21h

mov ax, 0

mov al, wordCount

mov bl, 10

mov cx, 0

dec\_loop:

cmp al, 0

je print\_digits

div bl

mov dl, ah

mov ah, 0

add dl, '0'

push dx

inc cx

jmp dec\_loop

print\_digits:

cmp cx, 0

jne print\_loop

mov dl, '0'

mov ah, 02h

int 21h

jmp done\_print

print\_loop:

pop dx

mov ah, 02h

int 21h

loop print\_loop

done\_print:

lea dx, newLine

mov ah, 09h

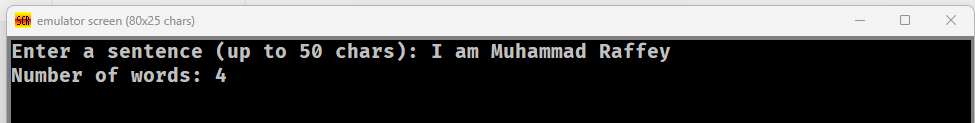
int 21h

mov ah, 4Ch

int 21h

MAIN ENDP

END MAIN



# **Vowel and Consonant Counter**

.MODEL SMALL

.STACK 100H

.DATA

msgPrompt db 'Enter a phrase (up to 30 chars): $'

msgVowels db 0Dh,0Ah, 'Number of vowels: $'

msgCons db 0Dh,0Ah, 'Number of consonants: $'

newLine db 0Dh,0Ah, '$'

userInput db 30 dup (?)

vowCount db 0

consCount db 0

.CODE

MAIN PROC

mov ax, @DATA

mov ds, ax

lea dx, msgPrompt

mov ah, 09h

int 21h

mov cx, 30

lea bx, userInput

read\_loop:

mov ah, 01h

int 21h

cmp al, 0Dh

je end\_read

mov [bx], al

inc bx

loop read\_loop

end\_read:

mov al, 0

mov [bx], al

mov vowCount, 0

mov consCount, 0

lea bx, userInput

count\_loop:

mov al, [bx]

cmp al, 0

je done\_count

cmp al, 'a'

jb skip\_case\_convert

cmp al, 'z'

ja skip\_case\_convert

sub al, 20h

skip\_case\_convert:

cmp al, 'A'

jb next\_char

cmp al, 'Z'

ja next\_char

cmp al, 'A'

je is\_vowel

cmp al, 'E'

je is\_vowel

cmp al, 'I'

je is\_vowel

cmp al, 'O'

je is\_vowel

cmp al, 'U'

je is\_vowel

inc consCount

jmp next\_char

is\_vowel:

inc vowCount

next\_char:

inc bx

jmp count\_loop

done\_count:

lea dx, msgVowels

mov ah, 09h

int 21h

mov ax, 0

mov al, vowCount

mov bl, 10

mov cx, 0

vow\_loop:

cmp al, 0

je vow\_print\_digits

div bl

mov dl, ah

mov ah, 0

add dl, '0'

push dx

inc cx

jmp vow\_loop

vow\_print\_digits:

cmp cx, 0

jne vow\_pop

mov dl, '0'

mov ah, 02h

int 21h

jmp vow\_done

vow\_pop:

pop dx

mov ah, 02h

int 21h

loop vow\_pop

vow\_done:

lea dx, msgCons

mov ah, 09h

int 21h

mov ax, 0

mov al, consCount

mov bl, 10

mov cx, 0

cons\_loop:

cmp al, 0

je cons\_print\_digits

div bl

mov dl, ah

mov ah, 0

add dl, '0'

push dx

inc cx

jmp cons\_loop

cons\_print\_digits:

cmp cx, 0

jne cons\_pop

mov dl, '0'

mov ah, 02h

int 21h

jmp cons\_done

cons\_pop:

pop dx

mov ah, 02h

int 21h

loop cons\_pop

cons\_done:

lea dx, newLine

mov ah, 09h

int 21h

mov ah, 4Ch

int 21h

MAIN ENDP

END MAIN

