|  |  |
| --- | --- |
| .386  INCLUDE Irvine32.inc  .data  title\_msg db "==============================", 0  welcome\_msg db " PASSWORD STRENGTH CHECKER ", 0  line\_break db "==============================", 0  prompt db "Enter your password: ", 0  weak\_msg db "[Weak] Password is too simple! Enter a stronger one.", 0  medium\_msg db "[Medium] Your password is okay. Do you want to continue? (Y/N): ", 0  strong\_msg db "[Strong] Great! Your password is secure.", 0  confirm\_msg db "Confirm your password: ", 0  success\_msg db "Password matched! Welcome!", 0  fail\_msg db "Passwords do not match. Try again!", 0  entropy\_msg db "Entropy Score: ", 0  newline db 10, 13, 0  user\_input db 20 dup(?) ; Buffer for user input  confirmed\_password db 20 dup(?) ; Buffer for confirmation  pass\_length DWORD 0  has\_upper DWORD 0  has\_lower DWORD 0  has\_digit DWORD 0  has\_special DWORD 0  entropy\_score DWORD 0  user\_choice db ?  common\_passwords db "123456",0, "password",0, "qwerty",0, "abc123",0, "111111",0, "123123",0, "admin",0  common\_count DWORD 7  .code  main PROC  call Clrscr  mov edx, OFFSET title\_msg  call WriteString  call Crlf  mov edx, OFFSET welcome\_msg  call WriteString  call Crlf  mov edx, OFFSET line\_break  call WriteString  call Crlf  get\_password:  mov edx, OFFSET prompt  call WriteString  mov edx, OFFSET user\_input  mov ecx, 20  call ReadString  mov pass\_length, eax  mov has\_upper, 0  mov has\_lower, 0  mov has\_digit, 0  mov has\_special, 0  mov entropy\_score, 0  mov esi, OFFSET user\_input  xor ecx, ecx  check\_loop:  mov al, [esi]  cmp al, 0  je done\_check  cmp al, 'A'  jl check\_lower  cmp al, 'Z'  jg check\_lower  mov has\_upper, 1  add entropy\_score, 6  jmp next\_char  check\_lower:  cmp al, 'a'  jl check\_digit  cmp al, 'z'  jg check\_digit  mov has\_lower, 1  add entropy\_score, 5  jmp next\_char  check\_digit:  cmp al, '0'  jl check\_special  cmp al, '9'  jg check\_special  mov has\_digit, 1  add entropy\_score, 5  jmp next\_char  check\_special:  mov has\_special, 1  add entropy\_score, 10  next\_char:  inc esi  loop check\_loop  done\_check:  call check\_common\_passwords  cmp eax, 1  je handle\_weak    mov eax, pass\_length  add entropy\_score, eax  mov edx, OFFSET entropy\_msg  call WriteString  mov eax, entropy\_score  call WriteDec  call Crlf  cmp entropy\_score, 25  jl handle\_weak  cmp entropy\_score, 45  jl handle\_medium  jmp handle\_strong  handle\_medium:  call Crlf  mov edx, OFFSET medium\_msg  call WriteString  call ReadChar  mov user\_choice, al  call Crlf  cmp user\_choice, 'Y'  je confirm\_password  cmp user\_choice, 'y'  je confirm\_password  jmp get\_password  handle\_strong:  call Crlf  mov edx, OFFSET strong\_msg  call WriteString  call Crlf  jmp confirm\_password  handle\_weak:  call Crlf  mov edx, OFFSET weak\_msg  call WriteString  call Crlf  jmp get\_password  confirm\_password:  call Crlf  mov edx, OFFSET confirm\_msg  call WriteString  mov edx, OFFSET confirmed\_password  mov ecx, 20  call ReadString  mov esi, OFFSET user\_input  mov edi, OFFSET confirmed\_password  mov ecx, pass\_length  repe cmpsb  jne password\_mismatch  call Crlf  mov edx, OFFSET success\_msg  call WriteString  call Crlf  jmp done  password\_mismatch:  call Crlf  mov edx, OFFSET fail\_msg  call WriteString  call Crlf  jmp confirm\_password  done:  call Crlf  mov edx, OFFSET line\_break  call WriteString  call Crlf  exit  check\_common\_passwords PROC  mov esi, OFFSET common\_passwords  mov ecx, common\_count  common\_loop:  push esi  push ecx  mov edi, esi  mov esi, OFFSET user\_input  mov ecx, pass\_length  repe cmpsb  pop ecx  pop esi  je weak\_found  add esi, 7  loop common\_loop  mov eax, 0  ret  weak\_found:  mov eax, 1  ret  check\_common\_passwords ENDP  main ENDP  END main |  |