void encrypt\_chars (int length, char EKey)

{ char temp\_char; // Character temporary store

for (int i = 0; i < length; i++){ // Encrypt characters one at a time

temp\_char = OChars [i]; // Get the next char from Original Chars array

\_\_asm {

push eax // Pushes the old values of eax onto stack to save it for later.

push ecx // Pushes the old values of ecx onto stack to save it for later.

movzx ecx,temp\_char // Copies one of the characters, typed and stored into OChars array, into ecx register but makes value 32 bits by adding zeros.

lea eax,EKey // Gets the memory address for the encryption key and moves it into the eax register.

push ecx //Pushes the new ecx onto the stack.

push eax //Pushes the new eax onto the stack.

call encrypt\_13 //This will call the encryption assembly code down below.

mov temp\_char,al //Copies the last 8 bits of the Ekey into temp\_char.

add esp, 8 //Adds eight to esp to point the stack pointer to the base of the stack. Eight is used as both eax and ecx contain 4 bits each.

pop ecx // this will pop ecx from the stack

pop eax // this will pop eax from the stack

}

EChars [i] = temp\_char; // Store encrypted char in the Encrypted Chars array

}

return;

\_\_asm {

encrypt\_13:

push ebp //Pushes the base pointer onto to the stack.

mov ebp, esp //copies the stack pointer onto the base pointer so they are at the same position to begin with.

mov eax, [ebp+8] //copies the value of eax on the stack, into the eax register

mov ecx, [ebp+12] //copies the value of ecx on the stack, into the ecx register

push ebx //push ebx onto the stack

push ecx //push char onto the stack

mov ebx, [eax] //moves Ekey from the stack onto the ebx register

xchg eax, ebx //exchanges value in ebx register with value in eax register

and eax, 0xFF //converts Ekey value and the number 255 in hexidecimal to binary and performs the boolean "and" opperation on both values

ror al, 1 //shifts last 8 bit values of Ekey register to right by 1. Number furthest to right moved to carry flag and copied to furthest bit on left

add eax, 0x01 //adds the number 1 in hexideecimal to the Ekey register

mov edx, eax //copies the Ekey from the eax register into the edx register

xchg eax, ebx // exchanges value in ebx register with value in eax register

mov[eax], edx //copies the value in edx register to the value in eax regiter

and edx, 0x0F //converts edx value and the number 15 in hexidecimal to binary and performs the boolean "and" opperation on both values

pop eax //removes the Ekey from the stack

add eax, edx // adds the value of edx with the Ekey.

xor eax, edx //converts edx value and the Ekey value to binary and performs the boolean "xor" opperation on both values

pop ebx // removes the ebx register from the stack

pop ebp //pops the base pointer off the stack

ret // returns back to the point of call and exits out of the assembly code

}

}