noopt.asm:

q1:

; 20 : strcpy(str, "A st");

; the string "A st" , including the null terminator, was previously

; stored in the variable $SG4294967291

; \_str$ is the offset of the beggining of str array (str array in the c program) from ebp

mov ecx, DWORD PTR $SG4294967291

; copy the first 4 bytes of the variable $SG429496729

; which means it copied "A st" to ecx, without the null terminator

mov DWORD PTR \_str$[ebp], ecx

; copy the content of ecx to the beggining of str array

; which means copy "A st" to the beggining of str array

mov dl, BYTE PTR $SG4294967291+4

; copy the null terminator after "A st" to the lowest byte of edx

mov BYTE PTR \_str$[ebp+4], dl

; copy the null terminator from dl to the fifth byte of str arry

; which means put a null terminator in str after "A st"

opt.asm:

; Same as noopt.asm, except, it uses only eax without using ecx and edx registers.

ex1.s:

/NO\_APP

; notice that 1953701953 = 0x74732041

; in addition, in ASCII representation, 0x41 = 'A', 0x20 = ' ', 0x73 = 's', 0x74 = 't'

; that means that the constans 1953701953 represents the 4 chars - "A st"

lea eax, [ebp-208]

; calculates ebp-208 and store it in eax

mov DWORD PTR [eax], 1953701953

; copy "A st" (as previously explained) to the address stored in eax

; which means copy "A st" to the beggining of str array

mov BYTE PTR [eax+4], 0

; put null terminator at the fifth byte of str array => put null terminator after "A st"

q2: