```
/**
 * strcpy.s
 * PARAMETERS: X0 (STRING TO BE COPIED)
 * OUTPUT    : X0 (POINTER TO COPIED STRING)
 * The purpose of this function is to copy a string dynamically using malloc.
 * It is up to the user to free the memory when done.
 * ALL REGISTERS PRESERVED EXCEPT X0
 */

    .text
    .global strcpy

strcpy:
        // storing X0-X19 registers, as malloc will not preserve most of these
        str X1,     [sp, -16]!
        stp X2, X3, [sp, -16]!
        stp X4, X5, [sp, -16]!
        stp X6, X7, [sp, -16]!
        stp X8, X9, [sp, -16]!
        stp X10, X11, [sp, -16]!
        stp X12, X13, [sp, -16]!
        stp X14, X15, [sp, -16]!
        stp X16, X17, [sp, -16]!
        stp X18, X19, [sp, -16]!
        stp X20, X21, [sp, -16]!
        str lr, [sp, -16]!
        mov x1, #10000  // setting a theoretical max string value, can be adjusted accordingly
        mov x19, x0     // storing a copy of x0 into x19
        bl  length      // calling length to fulfill malloc’s parameter of requested bytes
        add x0, x0, #1
        mov x21, x0     // move length into x21
        bl  malloc      // call malloc
        mov x20, #0     // setting variable to 0 for loop count
strcpyLoop:
        ldrb w17, [x19, x20]    // loading byte of given string into w17
        strb w17, [x0, x20]     // storing w17 into new string
        add x20, x20, #1        // incrementing
        cmp x20, x21            // comparing x19 to x20
        b.ge end                // if it’s greater than the length, goto end
        b strcpyLoop
end:
         // popping registers back from stack
        ldr lr, [sp], 16
        ldp X20, X21, [sp], 16
        ldp X18, X19, [sp], 16
        ldp X16, X17, [sp], 16
        ldp X14, X15, [sp], 16
        ldp X12, X13, [sp], 16
        ldp X10, X11, [sp], 16
        ldp X8, X9, [sp], 16
        ldp X6, X7, [sp], 16
        ldp X4, X5, [sp], 16
        ldp X2, X3, [sp], 16
        ldr X1,     [sp], 16
        ret lr
```