```c
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include "processControlFcns.h"

// The following is a small part of a very sensitive
// program used to modify the temperature of a chemical process
// in a water cleaning plant. Sometimes the process is too warm
// and the process needs to be cooled. Two externally defined
// functions are used
//        int readTemperature()
//            returns the current temperature of the process
//        void lowerTemperature1Degree()
//            lowers the temperature one degree
// This program is accessible via a web interface, but it is protected
// by a secret password. To lower the temperature several degrees, we
// call lowerTemperature1Degree() through a loop. This function should
// never be called more than 9 times in a row because then the system
// breaks and the water is no longer cleaned.

int convertToNum(char chr) {
// convert character chr to a number using the underlying ASCII
// encoding. Each character is encoded using the ASCII-code, so a '0'
// is coded as 48, a '1' as code 49, an 'A' as code 65 etc.
// return -1 if an error occurrs

  int num = -1;

  if (chr>47 && chr<58) {
     // chr is a CHAR = printable letter where
     // code 48= numerical 0;code 57=numerical 9
     num = chr-48; // num is an INT containing the corresponding number
  }

  return num;
}

int main(void) {
   char secret[] = "pa33";
   char usrName[6];
   char usrPassword[9];
   char lowerTempChr[2];
   int  lowerTemp;

   // check if user is authorized to run program
   printf("Enter username:");
   gets( usrName );
   printf("Enter password:");
   gets( usrPassword );
   if ( strcmp( usrPassword, secret ) !=0 ) {
      printf("You are not allowed to run this program\n");
      exit(1);
   }

   printf("Current temperature of process is %d degrees\n", readTemperature());
   printf("Enter degrees to lower temperature (0—9):");
   gets(lowerTempChr);
   lowerTemp= convertToNum(lowerTempChr[0]); // only convert first char

   printf("The temperature will now be lowered one degree at a time, for a total of %d degrees.\n", lowerTemp
);

   while (lowerTemp != 0) {
     printf("Lowering temperature by one degree (%d)\n", lowerTemp);
     lowerTemperature1Degree();
     lowerTemp = lowerTemp -1;
   }

   printf("Current temperature of process is %d degrees\n", readTemperature());
   printf("Operation has now finished. Bye!\n");
   return 0;
}
```