```
/* * * * * * * * * * * * * * * * * * * * * * * * *
 * Author: Matthew Casiro                        *
 * Date: Jan 26 2016                             *
 * Purpose: Ensure user input is a valid integer *
 * * * * * * * * * * * * * * * * * * * * * * * * */
#include <stdio.h>
#include <stdbool.h>
#define MAX_CNT 9
#define NEG_ONE -1

enum Status_Code {
   DONE, MAKE_NEGATIVE, IGNORE_ZERO, ADD_DIGIT, RESET };

// Evaluate character and return appropriate result code
int checkInput(int input, bool isFirst, bool onlyNegative, bool
allowLeadingZero, int *counter) {
   if ((input == '\n' || input == EOF) && !onlyNegative) {
      return DONE;
   }
   if (*counter < MAX_CNT) {
      if (isFirst && input == '-') {return MAKE_NEGATIVE;}
      if (allowLeadingZero && input == '0') {return IGNORE_ZERO;}
      if (input >= '0' && input <= '9') {
         *counter = *counter + 1;
         return ADD_DIGIT;
      }
   } return RESET;
}

int getInt() {
   int input = 0, output = 0;
   bool doneChecks, isNegative = false, doneInput = false;
   bool isFirst = true, allowLeadingZero = true, onlyNegative = false;
   unsigned counter = 0;
   enum Status_Code result;

   while (!doneInput) {
      // Presume valid until proven otherwise
      doneChecks = false;
      while (!doneChecks) {
         // Pull next character in line and evaluate
         input = getc(stdin);
         result = checkInput(input, isFirst, onlyNegative,
allowLeadingZero, &counter);
         // React appropriately to current input character
         switch (result) {
            case DONE :
               // Skip to output
               doneChecks = true;
               doneInput = true;
               break;
            case MAKE_NEGATIVE :
               isFirst = false;
               isNegative = true;
               onlyNegative = true;
               break;
            case IGNORE_ZERO :
               // If leading zero, do nothing
               break;
            case ADD_DIGIT :
               // Shift output by 10^1, add valid input value
               output = (output * 10) + (input - '0');
               isFirst = false;
               onlyNegative = false;
               allowLeadingZero = false;
               break;
            case RESET :
               // Reset input buffer
               while (input != '\n' && input != EOF) {
                  input = getc(stdin);
               }
               // Reset variables
               output = counter = 0;
               isFirst = allowLeadingZero = true;
               isNegative = onlyNegative = false;
               doneChecks = true;
               doneInput = false;
               printf("ERR: Enter an integer between +/-
999,999,999:\n");
               break;
            default :
               printf("\n!!!~FATAL ERROR~!!!\n");
         }
      }
   }
   // Check for negative value and adjust
   if (isNegative) output *= NEG_ONE;

   return output;
}
```