```c
#include "Proj_3E_header_file.h"

int main (void){
long Num_1;
char digits[8];
int counter = 0;

setup_HW;
User_prompt;

String_to_PC("Enter positive number & terminate with Return key.\r\n");
Num_1 = Num_from_KBD_Local(digits);                                    //Acquires data from keyboard

do{
Num_to_PC(10,++counter); Char_to_PC('\t');Num_to_PC(10,Num_1); newline();   //Converts numbers strings and sends them to the pc.
I2C_Tx_long(Num_1);                                                   //Sends number to the display
waitforkeypress();
Num_1 = (Num_1 / 2) *3;} while (Num_1 < 66666666);                     //Do some arithmetic

Num_1 = (Num_1 / 3) *2;

do{Num_1 = (Num_1 / 3) *2;                                             //Do the arithmetic in reverse
Num_to_PC(10,--counter); Char_to_PC('\t');Num_to_PC(10,Num_1); newline();
I2C_Tx_long(Num_1);
waitforkeypress();}while (counter-1);}


/*****************************************************************************************/
long Num_from_KBD_Local(char digits[]){                                //Local version, ignores negative number
unsigned char keypress;                                                //Resources version does not.
for(int n = 0; n<=7; n++) digits[n] = 0;                               //Clear array

do
{keypress =  waitforkeypress();}                                       //wait for first keypress
while (!(decimal_digit(keypress)));                                    //Ignore illegal characters
digits[0] = keypress;                                                  //Save first one to array
I2C_Tx_8_byte_array(digits);                                           //send array to display
while ((keypress = wait_for_return_key()) != '\r'){                    //Continue until return key is pressed

if (decimal_digit (keypress))                                          //Ignore illegal characters
{for(int n = 7; n>=1; n--)digits[n] = digits[n-1];                     //Shifts display left for each keypress
digits[0] = keypress;                                                  //Places new digit in array[0]
I2C_Tx_8_byte_array(digits);}}                                         //Updates the display
newline();
return I2C_displayToNum();}                                            //Mini-OS converts display to a long number and sends it to the UNO.
```