1. 在没有溢出的情况下都是成立的

int quot(int x, int y)

*//@requires* y!=0;

*//@ensures* \result == x / y;

{

  int a = x;

  int b = y;

  int s\_a = 1;

  int s\_b = 1;

  int c;

*if* (a < 0) {

    s\_a = -1;

    a = -a;

  }

*if* (b < 0) {

    s\_b = -1;

    b = -b;

  }

*for* (c=0; a>=b; c++)

*//@loop\_invariant* a>=0 && b>0 && c\*b+a==s\_a\*x;

    {

      a = a - b;

    }

*if* (s\_a \* s\_b < 0 && a != 0)

    {

      c = s\_a\*s\_b\*c - 1;

    }

*else*

      c = s\_a\*s\_b\*c;

*// @assert c\*y+a == s\_a\*x;*

*return* c;

}

int rem(int x,int y)

*//@requires* y>0;

*//@ensures* \result==x % y;

{

    int tmp\_x = x;

*while*(tmp\_x >= y){

        tmp\_x -= y;

    }

*return* tmp\_x;

}

bool t(int x,int y)

{

*return* quot(x,y)\*y + rem(x,y) == x;

}

二、整数转化成16进制

string int2hex(int x)

{

    int cur = 2;

    char[] res = alloc\_array(char,100)  ;

    res[0] = '0';

    res[1] = 'x';

*while*(x !=0){

      int index = x % 16;

      x /= 16;

*if*( index < 10){

        res[cur] = char\_chr(char\_ord('0')+index);

        cur++ ;

      }*else*{

        res[cur] = char\_chr(char\_ord('a')+index-10);

        cur++;

      }

    }

    int start = 2;

    int end = cur-1;

*while*(start < end){

        char tmp = res[start];

        res[start] = res[end];

        res[end] = tmp;

        start++;

        end--;

    }

    res[cur] = '\0';

*return* string\_from\_chararray(res);

}

三、逻辑右移

int lsr(int x,int l ){

    int mark = 0x7fffffff;

*for*(int i=0;i<l;i++){

        x /= 2;

        x = x & mark;

    }

*return* 0;

}

lsr(x,l) 当x>0时，其补码就是本生，lsr(x.l)相当于算数右移l位

当 x<0时，用补码表示时，lsr(x,l)相当于将其补码看成源码，执行算数右移。