```
int Bitwalker Poke(unsigned int Startposition, unsigned int Length,
                      uint8 t *Bitstream, unsigned int BitstreamSizeInBytes,
                      uint64 t Value)
    int
         retres;
    uint64 t MaxValue;
    int i;
    /*@ assert
        rte: unsigned overflow:
          0 <= (unsigned int)(Startposition+Length)-(unsigned int)1;</pre>
    */
0
   /*@ assert rte: unsigned overflow: 0 <= Startposition+Length; */</pre>
    /*@ assert rte: unsigned overflow: Startposition+Length <= 4294967295; */
    if (((Startposition + Length) - (unsigned int)1) >> 3 >= BitstreamSizeInBytes) {
        retres = -1:
      goto return label;
0
    /*@ assert
        rte: unsigned overflow:
          0 <=
          (unsigned long)((unsigned long long)0x01<<Length)-(unsigned long long)1;</pre>
    */
0
   /*@ assert rte: shift: 0 <= Length && Length < 64; */
    MaxValue = ((unsigned long long)0x01 << Length) - (unsigned long long)1;
    if (MaxValue < Value) {</pre>
        retres = -2;
      goto return label;
0
    /*@ assert
        rte: unsigned overflow:
          0 <= (unsigned int)(Startposition+Length)-(unsigned int)1;</pre>
/*@ assert rte: unsigned overflow: 0 <= Startposition+Length; */</pre>
    /*@ assert rte: unsigned overflow: Startposition+Length <= 4294967295; */
    i = (int)((Startposition + Length) - (unsigned int)1);
    while (i >= (int)Startposition) {
      if ((Value & (unsigned long long)0\times01) == (unsigned long long)0) {
        /*@ assert rte: mem access: \valid(Bitstream+(int)(i>>3)); */
0
        /*@ assert rte: shift: 0 <= i; */
0
        /*@ assert rte: mem access: \valid read(Bitstream+(int)(i>>3)); */
0
        /*@ assert rte: index bound: 0 <= (int)(i&0x07); */</pre>
0
        /*@ assert rte: index bound: (int)(i&0x07) < 8; */</pre>
        *(Bitstream + (i >> 3)) = (unsigned char)((int)*(Bitstream + (i >> 3)) & ~ ((int)BitwalkerBitMaskTable[
                                                    i & 0x071));
      else {
        /*@ assert rte: mem access: \valid(Bitstream+(int)(i>>3)); */
0
        /*@ assert rte: shift: 0 <= i; */
0
        /*@ assert rte: mem access: \valid read(Bitstream+(int)(i>>3)); */
0
        /*@ assert rte: index bound: 0 <= (int)(i&0x07); */</pre>
0
        /*@ assert rte: index bound: (int)(i&0x07) < 8; */
        *(Bitstream + (i >> 3)) = (unsigned char)((int)*(Bitstream + (i >> 3)) | (int)BitwalkerBitMaskTable[
                                                    i & 0x07]);
      Value >>= 1;
/*@ assert rte: signed overflow: -2147483648 <= i-1; */
      retres = 0;
    return label: return retres;
```