Algorithm

Functions used:

1)add = this function will take numbers in the form of string, one number will be of n-bits and the other will be of 1-bit and will return their sum in the form of string.

2)append_zero = this will take two arguments, one will be the number which needs to be appended with zero and the other will be the number of zeroes to be appended. The numerical value of the number will remain same after computation.

3)remove_zero = this will remove the additional zeroes present in the number at the start.

4)subtract = this will take two numbers in the form of strings and will return their difference.

5)multiply = this function will take numbers in the form of string, one number will be of n-bits and the other will be of 1-bit and will return their product in the form of string.

6)find_sqrt = this will find the floored integer square root of a number, input will be only upto 2 digit numbers.

7)compare = this will take two numbers as an argument and will return true if first number is strictly greater than the other one.

8)check_num = this will take two numbers and will return the smaller number among the two by appending the appropriate unit digit to it. Appropriate unit digit refers to the operation we make while long division step, i.e

(smaller number appended with appropriate_digit)*(appropriate_digit)<=larger number

9)help1 = this will return answer i.e floored integer square root along with remainder for case when number is 2 digit long.

10)help2 = this will return answer i.e floored integer square root along with remainder for case when number is 1 digit long.

11)help = this is the main function, this will take arguments (num,up,left,diff,start,last). Num will be the number given in testcase, up will contain the answer calculated upto digit start, left will contain the divisor in the long division method for the current value of start, diff will be the dividend, this function will return the tuple (up,diff) when value of start equals last.

12)isqrtld = this will call help1, help2, help for trivial cases(when input is less than 3 digit long) and return the desired answer otherwise for the first step of the long division method for finding the floored square root of first two/one digit it will call check_num, after that it will pass appropriate parameters to help() which will return answer for different cases(odd/even).

Proof of Correctness

Proof of correctness of major functions:

1)check_num(dividend,divisor,iterator):

- Iterator will start with i=1
- Let divisor be of type $a_n a_{n-1}....a_0$ and the desired answer d be such that $a_n a_{n-1}....a_0 d$ x d ≤ dividend

Invariant: for all j : j<i , $a_n a_{n-1}....a_0 j$ x j ≤ dividend

Recursion: if $a_n a_{n-1}....a_0 i$ x i ≤ dividend then check_num(dividend,divisor,i+1)

Else return i-1

Since invariant is true for any step of recursion therefore recursion is correct.

2)find_sqrt(num,iterator):

Invariant: for all j : j<i , $j^2 \leq num$

Recursion: if $i^2 \leq num$ then find_sqrt(num,i+1)

Else return i-1

Since invariant is true for any step therefore recursion is correct.

3)help(num,quotient,divisor,dividend,iterator,last):

Induction Hypothesis: for all i<iterator quotient is correct.

Base Case: for i=1, i=2 hypothesis is true since find_sqrt() function is correct.

Induction Step: Assuming quotient is correct for iterator=i,

For i=i+2 inorder to find the next digit of quotient check_num() function

will be called which will definitely give correct output since check_num() is correct

therefore quotient is true for i=i+2, hence by induction I can say that the function is correct.

4)isqrtld(num):

Since this function is called only once and the above mentioned functions are called in this function therefore this function is also correct.