b) lemma: for an arbitrary positive integer n, we write n in decimal system, meanly, if n has k digits, n = m[k]…m[2]m[1],when the algorithm DivisibleBy11(n) performs the while loop the i th time just finished the 6th line, (notice: for every positive integer, DivisibleBy11(n) performs the while loop at least 1 time) if i is odd, , (m[i]… m[2]m[1] rem 11) = s. if i is even, (m[i]… m[2]m[1] rem 11) = (11-s).

proof in induction:

let P(i) = ” when the algorithm DivisibleBy11(n) performs the while loop the i th time just finished the 6th line, (notice: for every positive integer, DivisibleBy11(n) performs the while loop at least 1 time) if i is odd, (m[i]… m[2]m[1] rem 11) = s. if i is even, (m[i]… m[2]m[1] rem 11) = (11-s).”

for an arbitrary positive integer n, assume n has q digits in decimal system.

Base case:

When i = 1, s = (n rem 10) = m[1], k = (n div 10) = m[i]… m[3]m[2].

(m[1] rem 11) = m[1] = s

p(1) is true.

When i = 2, when executed the line number 5 and before the execution of line number 6, s = ((n div 10) rem 10) – (n rem 10) = (m[i]… m[3]m[2] rem 10) – m[1] = m[2] – m[1], k = ((n div 10) div 10))= m[i]… m[4]m[3].

When executing line number 6, if s <0 previously, then (m[2] < m[1]) AND (s = 11 - m[2] + m[1]),

If s>0 previously, then (m[2] > m[1]) AND (s = m[2] - m[1]),

Constructor case:

Assume for an arbitrary i q, p(i) is true. When the algorithm DivisibleBy11(n) performs the while loop the i th time just finished the 6th line, we have k and s,

Assume i is odd.

k = m[q]…m[i+1], s = (m[i]… m[2]m[1] rem 11)

i+1 is even,

if the i+1 th while loop does not exist, then p(i) is true for every iq.

else, for the i+1 th while loop, before the execution of line number 6,

s’ = (k rem 10) -s = m[i+1]-s

k’ = k rem 10 = m[q]…m[i+2],