```
kmer_length = 8
max_mismatch = 3     // maximum allowed mismatched base-pair
N = max_primer_length + 6
S = read_seq[0:N]
Hash = {all kmers}     // A hash table contains all kmer sequence of primers

/* start to perform 'k-mers model' */
i = 0
while i < 12 do
    key = S[i: i+kmer_length]
    primer = Hash[key]
    mis_num = Mismatch_Check(primer, S)
    if mis_num < max_mismatch then
        return primer
    end if
    i = i + 1
end while

/* if 'k-mers model' fails, start to perfrom 'dynamic model' */
K_Num = N - kmer_length      // k-mer number of sequence S
K = {k_j; j = 1..K_Num}      // all potential kmer sequence of S
Hit = {NULL}                 // record hit number for each canditate primer

j = 0
while j < K_Num do
    primer = Hash[K_j]
    Hit[primer] = Hit[primer] +1
    j = j + 1
end while

target_primer = Max_Hit_Primer(Hit)   // get primer that has highest kmer hit
mis_num = Dynamic_Check(target_primer, S) // Needleman-Wunsch local alignment
if mis_num < max_mismatch then
    return target_primer
end if
```