Problem 12.1

The implementation is in P1.cpp.

Problem 12.2

a) Rabin-Karp algorithm is used for searching a partial pattern in the given string, and print all occurrences of that pattern in the string. Unlike Naïve search string algorithm, Rabin-Karp algorithm compares the hash value of substrings of text and pattern first, and starts comparing character by character only when the hash values are equal. 1 shift to the right will be occurred in the text, and hash value of substring will be updated until both of them are equal. We always set a text with length=n and a pattern with length=m. Then, we have to choose a prime number q, d = Number of characters in the alphabet, h = d^(m-1).

hash(text[s+1 .. s+m]) = (d(hash(text[s .. s+m-1]) – text[s]*h) + text[s + m]) % q

hash(text[s .. s+m-1]) : Hash value at shift s.

hash(text[s+1 .. s+m]) : Hash value at next shift (or shift s+1)

```
RABIN-KARP-MATCHER(T, P, d, q)
 1  n = T.length
 2  m = P.length
 3  h = d^(m−1) mod q
 4  p = 0
 5  t_0 = 0
 6  for i = 1 to m                    // preprocessing
 7      p = (dp + P[i]) mod q
 8      t_0 = (dt_0 + T[i]) mod q
 9  for s = 0 to n − m                // matching
10      if p == t_s
11          if P[1..m] == T[s + 1..s + m]
12              print "Pattern occurs with shift" s
13      if s < n − m
14          t_{s+1} = (d(t_s − T[s + 1]h) + T[s + m + 1]) mod q
```

For instance, T=ABDCB, P=DC, q=11, n=5, m=2, d=256. So, h= (1*256)%11 = 3.

At for loop at line 6,

Iter 1: p = (256*0+68)%11 = 2

t = (256*0+65)%11 = 10

Iter 2: p = (256*0+67)%11 = 7

t = (256*10+68)%11 = 8

And we go to line 9, starting new loop to compare hash values. We get different hash values: p=7, t=8, so we skip 1st if statement. Since (s=0)<(n-m=3), So, we will do shifting in the text and calculate the hash value. If t<0, we always update t+=q to get a positive hash value. Now, t = (256*(8-65*3)+68)%11 = -9, and updated t=-9+11=2.

At shift 1, t=2,p=7 (p≠t), so we move 2nd if statement since (s=1)<(n-m=3): update t. Now, t = (256*(2-66*3)+67)%11= -4+11 = 7.
At shift 2, we found p=t=7 so we print line 12. Since (s=2)<(n-m=3), we update t = (256*(7-68*3)+66)%11= -8+11 = 3. At shift 3, t=3,p=7 (p≠t), so both if statements in the loop will be false. So, we found only one pattern match at shift 2, and the algorithm is finished successfully.

b) The implementation is in P2.cpp.

References

Mandal, K. P. (2018, November 21). Place K-knights such that they do not attack each other. Retrieved from https://www.geeksforgeeks.org/place-k-knights-such-that-they-do-not-attack-each-other/
(I used this website to get some inspirations for Problem 12.1)

Rathbhupendra. (2019, April 08). Rabin-Karp Algorithm for Pattern Searching. Retrieved from https://www.geeksforgeeks.org/rabin-karp-algorithm-for-pattern-searching/
(I used this website to get some inspirations for Problem 12.2)