

```

1  /* KMP.c */
2  #include <stdio.h>
3  #include <stdlib.h>
4
5  void buildNext(char *pattern, int *next, int len)
6  {
7      next[0] = -1;
8      int i = 0, t = -1;
9      while (i < len - 1)
10     {
11         (0 > t || pattern[i] == pattern[t]) ? (next[++ i] = ++ t) : (t = next[t]);
12     }
13 }
14
15 void buildNextImprove(char *pattern, int *next, int len)
16 {
17     next[0] = -1;
18     int i = 0, t = -1;
19     while (i < len - 1)
20     {
21         if (0 > t || pattern[i] == pattern[t]) //匹配
22         {
23             i ++;
24             t ++;
25             //引入“负面”信息
26             if (pattern[i] != pattern[t])
27             {
28                 next[i] = t;
29             }
30             else
31             {
32                 next[i] = next[t];
33             }
34         }
35         else //失配
36         {
37             t = next[t];
38         }
39     }
40 }
41
42 static int match(char *pattern, int lenP, char *text, int lenT)
43 {
44     int *next = (int *)malloc(sizeof(int) * lenP);
45     buildNextImprove(pattern, next, lenP);
46     int i = 0, j = 0;
47     while (i < lenT && j < lenP)
48     {
49         if (0 > j || text[i] == pattern[j])
50         {
51             i ++;
52             j ++;
53         }
54         else
55         {
56             //失配在第0个位置时, next[0]=-1
57             j = next[j];
58         }
59     }
60     free(next);
61     return i - j;
62 }
63
64 int main()
65 {
66     int n, m;
67     scanf("%d", &n);
68     char *A = (char *)malloc(n + 1);
69     scanf("%s", A);
70     A[n] = '\0';
71     scanf("%d", &m);
72     char *B = (char *)malloc(m + 1);
73     scanf("%s", B);

```

```
74     B[m] = '\0';
75     int res = match(A, n, B, m);
76     if (res <= m - n)
77     {
78         printf("match success\n");
79     }
80     else
81     {
82         printf("match fail\n");
83     }
84     free(A);
85     free(B);
86     return 0;
87 }
```