

## 测试作业

### 1 完成二叉查找

```
1  int binarySearch(int* arr, int key)
2  {
3      int left = 0;
4      int right = N-1;
5      while (left<=right)
6      {
7          int mid;
8          mid = (left + right) / 2;
9          if (arr[mid] == key)
10         {
11             return mid;
12         }
13         else if (arr[mid]>key)
14         {
15             left = mid + 1;
16         }
17         else
18         {
19             right = mid - 1;
20         }
21     }
22     return 0;
23 }
```

### 2 往一个字符串数组当中插入20个不同的字符串，然后输入一个字符串，使用下列哈希算法判断该字符串是否出现在数组当中，如果出现，它的下标是多少？

```
int hash(char *key) {
    int h = 0, g;
    while (*key) {
        h = (h << 4) + *key++;
        g = h & 0xf0000000;
        if (g)
            h ^= g >> 24;
        h &= ~g;
    }
    return h % MAXKEY;
}
```

```
1  #include<iostream>
2  #include<cstdio>
```

```

3  #include<string>
4  #include<cstdlib>
5  #include <time.h>
6  constexpr auto MAXKEY = 20;
7  using namespace std;
8  int gethash(char* key);
9
10 int main()
11 {
12     char Arr[20][10] = {
13
14         "chunyang","qixiu","wanhua","shaolin","cangjian","badao","mingjiao","tiance",
15         "wudu","tangmen","gaibang","cangyun"
16
17         , "changge","penglai","lingxue","chenwei","tanmoti","zhen","renheyi","jiafei"
18         " };
19     int hashMap[20] = { 0 };
20     for (int i = 0; i < MAXKEY; i++) {
21         hashMap[i]=gethash((char*)Arr[i]);
22     }
23     char j[MAXKEY];
24     cin >> j;
25     int m = gethash(j);
26     for (int i = 0; i < MAXKEY; i++)
27     {
28         if (m == hashMap[i])
29         {
30             cout << i << endl;
31         }
32     }
33 }
34 int gethash(char* key) {
35     int h = 0, g;
36     while (*key) {
37         h = (h << 4) + *key++;
38         g = h & 0xf0000000;
39         if (g)
40             h ^= g >> 24;
41         h &= ~g;
42     }
43     return h % MAXKEY;
44 }

```

3 实现二叉排序树的插入

4 实现二叉排序树的左旋和右旋

5 整理红黑树的各种情况，写出伪代码，如果时间充足的同学可以直接实现红黑树的插入

\*6 实现二叉排序树的删除