  
代码：

**[cpp]** [view plaincopyprint?](http://blog.csdn.net/hackbuteer1/article/details/8017703)

1. //转载请标明出处，原文地址：http://blog.csdn.net/hackbuteer1/article/details/8017703
2. **bool** IsPrime(**int** n)
3. {
4. **int** i;
5. if(n < 2)
6. return false;
7. else if(2 == n)
8. return true;
9. if((n&1) == 0)    //n%2 == 0
10. return false;
11. for(i = 3 ; i\*i <= n ; i += 2)     //只考虑奇数
12. {
13. if(n % i == 0)
14. return false;
15. }
16. return true;
17. }
19. /\*
20. 考虑到所有大于4的质数，被6除的余数只能是1或者5
21. 比如接下来的5,7,11,13,17,19都满足
23. 所以，我们可以特殊化先判断2和3
24. 但后面的问题就出现了，因为并非简单的递增，从5开始是+2,+4,+2,+4,....这样递增的
25. 这样的话，循环应该怎么写呢？
27. 首先，我们定义一个步长变量step，循环大概是这样 for (i = 5; i <= s; i += step)
28. 那么，就是每次循环，让step从2变4，或者从4变2
29. 于是，可以这么写：
30. \*/
31. **bool** IsPrime2(**int** n)
32. {
33. **int** i, step = 4;
34. if(n < 2)
35. return false;
36. else if(2 == n || 3 == n)
37. return true;
38. if((n&1) == 0)    //n%2 == 0
39. return false;
40. if(n%3 == 0)      //n%3 == 0
41. return false;
42. for(i = 5 ; i\*i <= n ; i += step)
43. {
44. if(n % i == 0)
45. return false;
46. step ^= 6;
47. }
48. return true;
49. }
51. void print\_prime(**int** n)
52. {
53. **int** i , num = 0;
55. for(i = 0 ; ; ++i)
56. {
57. if(IsPrime2(i))
58. {
59. printf("%d  " , i);
60. ++num;
61. if(num == n)
62. break;
63. }
64. }
65. printf("\n");
66. }

//转载请标明出处，原文地址：http://blog.csdn.net/hackbuteer1/article/details/8017703

bool IsPrime(int n)

{

int i;

if(n < 2)

return false;

else if(2 == n)

return true;

if((n&1) == 0) //n%2 == 0

return false;

for(i = 3 ; i\*i <= n ; i += 2) //只考虑奇数

{

if(n % i == 0)

return false;

}

return true;

}

/\*

考虑到所有大于4的质数，被6除的余数只能是1或者5

比如接下来的5,7,11,13,17,19都满足

所以，我们可以特殊化先判断2和3

但后面的问题就出现了，因为并非简单的递增，从5开始是+2,+4,+2,+4,....这样递增的

这样的话，循环应该怎么写呢？

首先，我们定义一个步长变量step，循环大概是这样 for (i = 5; i <= s; i += step)

那么，就是每次循环，让step从2变4，或者从4变2

于是，可以这么写：

\*/

bool IsPrime2(int n)

{

int i, step = 4;

if(n < 2)

return false;

else if(2 == n || 3 == n)

return true;

if((n&1) == 0) //n%2 == 0

return false;

if(n%3 == 0) //n%3 == 0

return false;

for(i = 5 ; i\*i <= n ; i += step)

{

if(n % i == 0)

return false;

step ^= 6;

}

return true;

}

void print\_prime(int n)

{

int i , num = 0;

for(i = 0 ; ; ++i)

{

if(IsPrime2(i))

{

printf("%d " , i);

++num;

if(num == n)

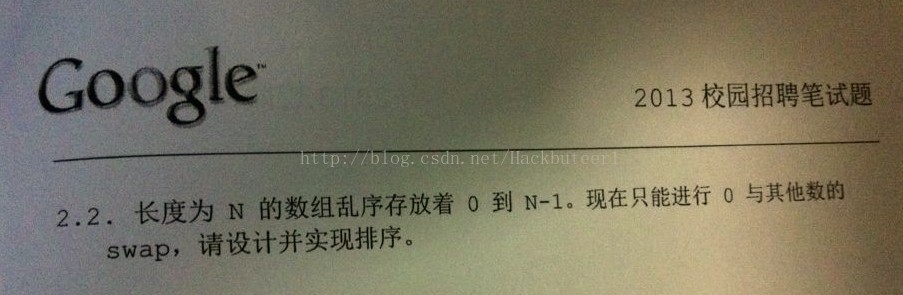
break;

}

}

printf("\n");

}

  
代码：

**[cpp]** [view plaincopyprint?](http://blog.csdn.net/hackbuteer1/article/details/8017703)

1. //转载请标明出处，原文地址：http://blog.csdn.net/hackbuteer1/article/details/8017703
2. void myswap(**int** a , **int** b , **int**\* array)
3. {
4. **int** temp = array[a];
5. array[a] = array[b];
6. array[b] = temp;
7. }
9. //利用0和其它数交换位置进行排序
10. void swap\_sort(**int**\* array , **int** len)
11. {
12. **int** i , j;
13. for(i = 0 ; i < len ; ++i)          //因为只能交换0和其他数，所以先把0找出来
14. {
15. if(0 == array[i])
16. {
17. if(i)   //如果元素0不再数组的第一个位置
18. myswap(0 , i , array);
19. break;
20. }
21. }
23. for(i = 1 ; i < len ; ++i)     //因为是0至N-1的数，所以N就放在第N的位置处
24. {
25. if(i != array[i])    //这个很重要，如果i刚好在i处，就不用交换了，否则会出错
26. {
27. for(j = i + 1 ; j < len ; ++j)
28. {
29. if(i == array[j])
30. {
31. myswap(0 , j , array);   //把0换到j处，此时j处是0
32. myswap(j , i , array);   //把j处的0换到i处，此时i处是0
33. myswap(0 , i , array);   //把i处的0换到0处
34. }
35. }//for
36. }
37. }//for
38. }

//转载请标明出处，原文地址：http://blog.csdn.net/hackbuteer1/article/details/8017703

void myswap(int a , int b , int\* array)

{

int temp = array[a];

array[a] = array[b];

array[b] = temp;

}

//利用0和其它数交换位置进行排序

void swap\_sort(int\* array , int len)

{

int i , j;

for(i = 0 ; i < len ; ++i) //因为只能交换0和其他数，所以先把0找出来

{

if(0 == array[i])

{

if(i) //如果元素0不再数组的第一个位置

myswap(0 , i , array);

break;

}

}

for(i = 1 ; i < len ; ++i) //因为是0至N-1的数，所以N就放在第N的位置处

{

if(i != array[i]) //这个很重要，如果i刚好在i处，就不用交换了，否则会出错

{

for(j = i + 1 ; j < len ; ++j)

{

if(i == array[j])

{

myswap(0 , j , array); //把0换到j处，此时j处是0

myswap(j , i , array); //把j处的0换到i处，此时i处是0

myswap(0 , i , array); //把i处的0换到0处

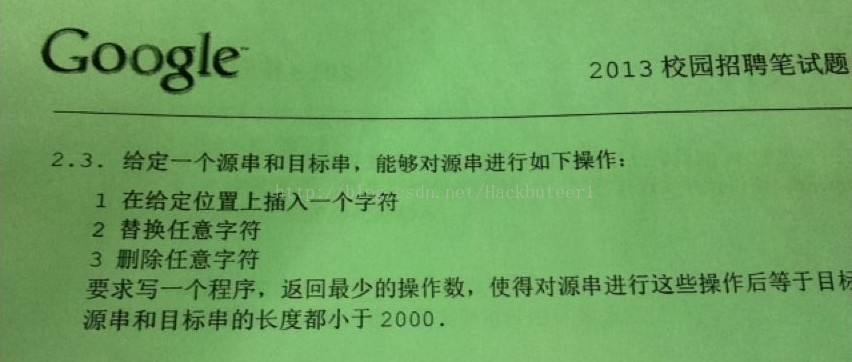
}

}//for

}

}//for

}



代码：

**[cpp]** [view plaincopyprint?](http://blog.csdn.net/hackbuteer1/article/details/8017703)

1. //转载请标明出处，原文地址：http://blog.csdn.net/hackbuteer1/article/details/8017703
2. **int** mymin(**int** a , **int** b , **int** c)
3. {
4. **int** temp = (a < b ? a : b);
5. return temp < c ? temp : c;
6. }
8. **int** min\_edit\_dic(**char**\* source , **char**\* target)
9. {
10. **int** i , j , edit , ans;
11. **int** lena , lenb;
12. lena = strlen(source);
13. lenb = strlen(target);
14. **int**\*\* distance = new **int**\*[lena + 1];
15. for(i = 0 ; i < lena + 1 ; ++i)
16. distance[i] = new **int**[lenb + 1];
17. distance[0][0] = 0;
18. for(i = 1 ; i < lena + 1 ; ++i)
19. distance[i][0] = i;
20. for(j = 1 ; j < lenb + 1 ; ++j)
21. distance[0][j] = j;
22. for(i = 1 ; i < lena + 1 ; ++i)
23. {
24. for(j = 1 ; j < lenb + 1 ; ++j)
25. {
26. if(source[i - 1] == target[j - 1])
27. edit = 0;
28. else
29. edit = 1;
30. distance[i][j] = mymin(distance[i - 1][j] + 1 , distance[i][j - 1]  + 1 , distance[i - 1][j - 1] + edit);
31. //distance[i - 1][j] + 1             插入字符
32. //distance[i][j - 1]  + 1            删除字符
33. //distance[i - 1][j - 1] + edit      是否需要替换
34. }
35. }
36. ans = distance[lena][lenb];
38. for(i = 0 ; i < lena + 1 ; ++i)
39. delete[] distance[i];
40. delete[] distance;
42. return ans;
43. }

//转载请标明出处，原文地址：http://blog.csdn.net/hackbuteer1/article/details/8017703

int mymin(int a , int b , int c)

{

int temp = (a < b ? a : b);

return temp < c ? temp : c;

}

int min\_edit\_dic(char\* source , char\* target)

{

int i , j , edit , ans;

int lena , lenb;

lena = strlen(source);

lenb = strlen(target);

int\*\* distance = new int\*[lena + 1];

for(i = 0 ; i < lena + 1 ; ++i)

distance[i] = new int[lenb + 1];

distance[0][0] = 0;

for(i = 1 ; i < lena + 1 ; ++i)

distance[i][0] = i;

for(j = 1 ; j < lenb + 1 ; ++j)

distance[0][j] = j;

for(i = 1 ; i < lena + 1 ; ++i)

{

for(j = 1 ; j < lenb + 1 ; ++j)

{

if(source[i - 1] == target[j - 1])

edit = 0;

else

edit = 1;

distance[i][j] = mymin(distance[i - 1][j] + 1 , distance[i][j - 1] + 1 , distance[i - 1][j - 1] + edit);

//distance[i - 1][j] + 1 插入字符

//distance[i][j - 1] + 1 删除字符

//distance[i - 1][j - 1] + edit 是否需要替换

}

}

ans = distance[lena][lenb];

for(i = 0 ; i < lena + 1 ; ++i)

delete[] distance[i];

delete[] distance;

return ans;

}