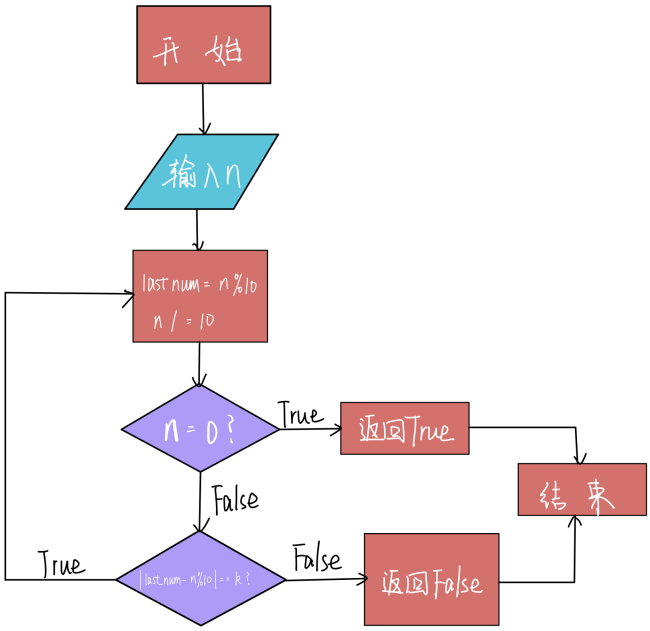
说明文档

暴力验证法

思路:

从10^{n-1}遍历到10^n-1对每个数检验一下符不符合相邻两位之间相差k。

检验函数思路:



时间复杂度

按位检测 : $O(n)$

从10^{n-1}遍历到10^n-1: $O(10^n)$

总时间复杂度 : $O(n*10^n)$

实现:

计算10的幂次使用快速幂节约时间:

```
LL fastpow(LL n, LL multi)
{
    LL ans = 1, base = n;
    while (multi)
    {
        if (multi & 1) ans = (ans * base);
        base = (base * base);
        multi >>= 1;
    }
    return ans;
}
```

检验函数:

```
bool check(LL _Source,int k)
{
    int pos=_Source%10;
    _Source /= 10;
    while (_Source > 0)
    {
        if (abs(_Source % 10 - pos) != k)
            return false;
        pos = _Source % 10;
        _Source /= 10;
    }
    return true;
}
```

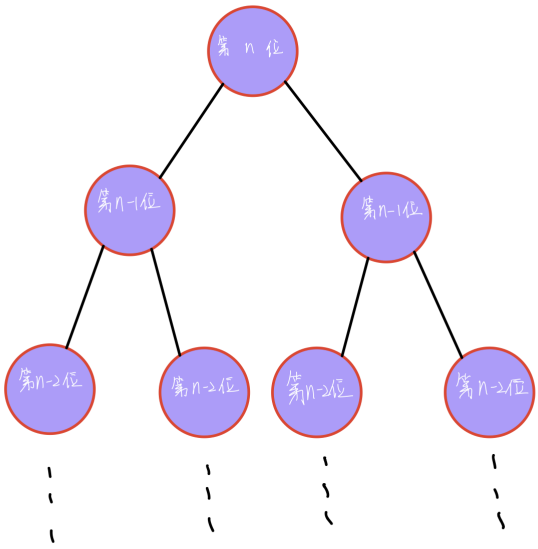
主函数&&n<2的处理：

```
int main()
{
    freopen("solution2_out.txt", "w", stdout);
    int n, k;
    scanf("%d%d", &n, &k);
    if (n < 2)
        return 0;
    LL start = fastpow( 10,n-1);
    LL end = fastpow(10,n);
    LL ans = 0;
    for (LL i = start; i < end; i++)
    {
        if (check(i,k)) {
            if (ans != 0)
                printf(",");
            printf("%lld",i);
            ans++;
        }
    }
    return 0;
}
```

按位构造法

思路

从最高位到最低位按照要求进行构造，就是对9个深度为n的二叉树进行深度优先搜索再带点可行性剪枝。



时间复杂度

每个位下面有两个选择，加2或者减2，所以时间复杂度$O(2^n)$

实现

深度优先搜索

```
bool dh = true;
void dfs(LL ans, int k, int depth,int last)
{
    if (depth == 0)
    {
        if (dh == true)
            dh = false;
        else
            printf(",");
        printf("%lld", ans);
        return;
    }
    if (last - k >= 0)
        dfs(ans * 10 + last - k, k, depth - 1, last - k);
    if (last + k <= 9&&k!=0)
        dfs(ans * 10 + last + k, k, depth - 1, last + k);
    return;
}
```

主函数

```
int main()
{
    freopen("solution2_out.txt", "w", stdout);
    int n, k;
    scanf("%d%d", &n, &k);
    if (n < 2)
        return;
    for (int i = 1; i <= 9; i++)
    {
        dfs(i, k, n - 1, i);
    }
}
```

总结

1. 这题可以用验证法和构造搜索法，其中构造搜索法时间复杂度最低，一般来说从题目的条件进行构造比验证要快，比如筛法求范围素数就比强行验证要快不少。
2. 本体如果是求个数而不要求输出的话可以用类似快速幂的方法求个数，但是要求输出的话类似快速幂的方法时间复杂度就会因为要记录而变得非常高。