# 项目简介

## 问题

返回长度为n的所有非负整数，使得每两个数字之间的绝对值之差为k。请注意，答案中的每个数字不得有前导0。例如，01有一个前导0，因此无效。结果需要按顺序返回答案。

## 思路

首先，给定输入为数字串长度n和约束条件k（相邻数字之间的绝对差），此时得出，当前一个数字确定时，后一个数字必定为前一个数字减去k或者加上k，由此可得到第二个数字，再在第二数字的基础上得到第三个数字。以此类推，当得到含n个数字的串时，即得到了一个符合要求的结果。

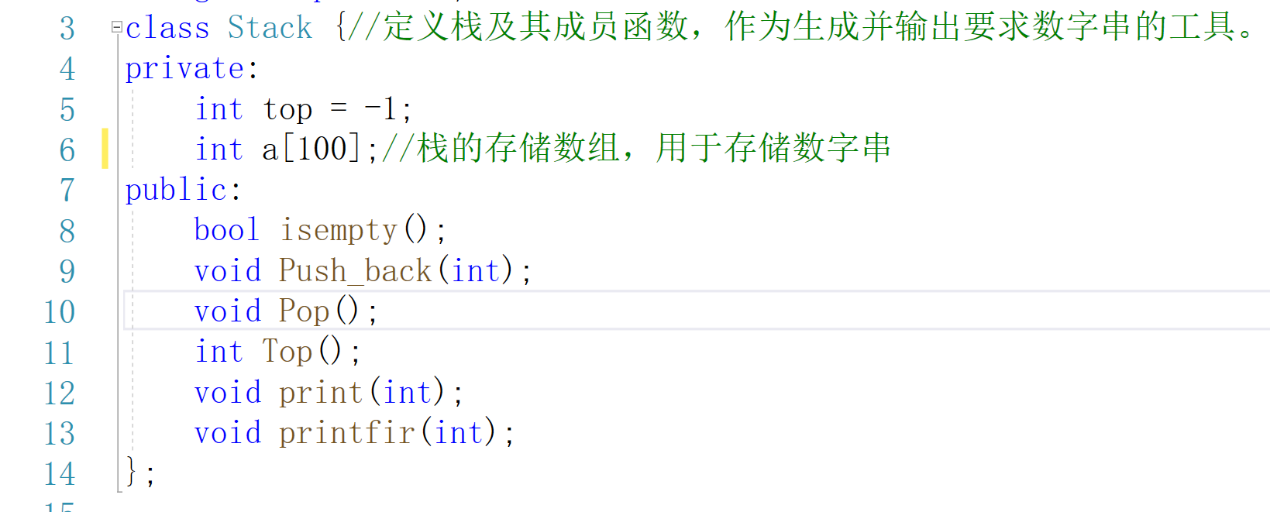
而由于需要输出全部满足条件的数字串，因此考虑按一定的逻辑顺序进行数字串的生成，从第一个到最后一个，即可完成全部输出。所以，为了排除前导0的情况，可以首先决定第一个数字，然后依次输出由1到9开头的所有满足条件的数字串。其中，除了第一个数字外，后面的每一个数字由其前一个数字决定，并且减的优先级大于加的优先级（即当减去k不满足条件后，再选择加上k；加上k也不满足条件，说明当前字符串无法生成，再对上一位进行判断调整），当得到以9开头的最后一个数字串时，全部输出完毕。

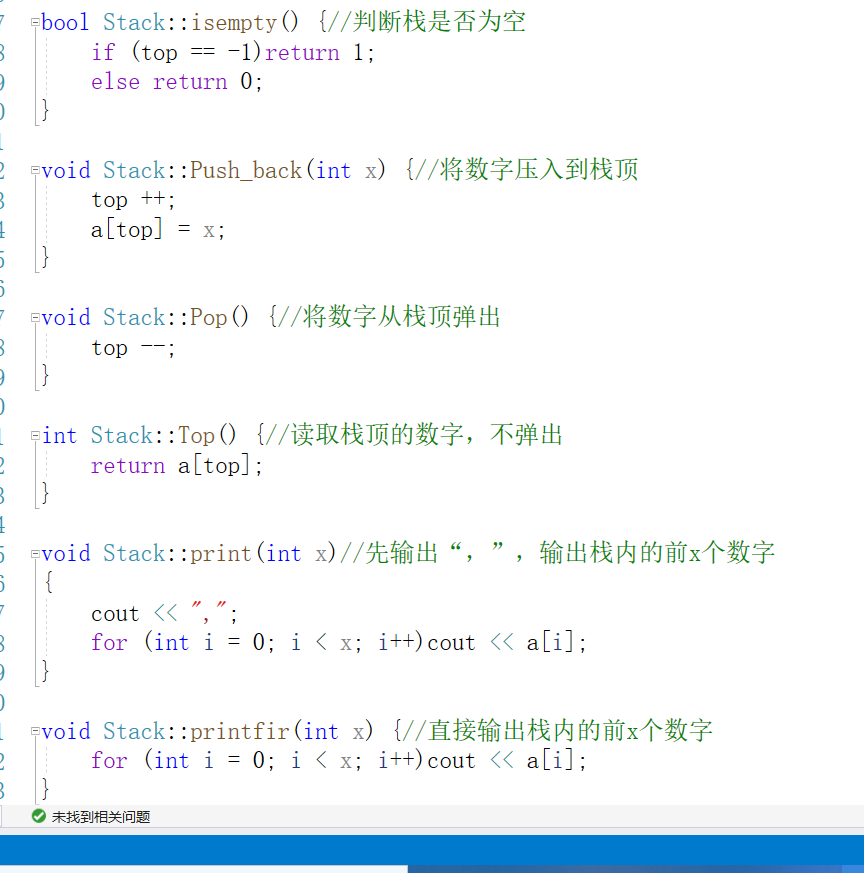
因为这种先进行最深度的尝试，再返回上一位进行调整，然后继续向深度进行尝试的方法和深度优先搜索非常相似，因此考虑以深搜为原型进行修改调整，并选择时间复杂度和空间复杂的都较小的非递归算法（即使用栈）。

## 3 重要实现

## 3.1 栈的定义

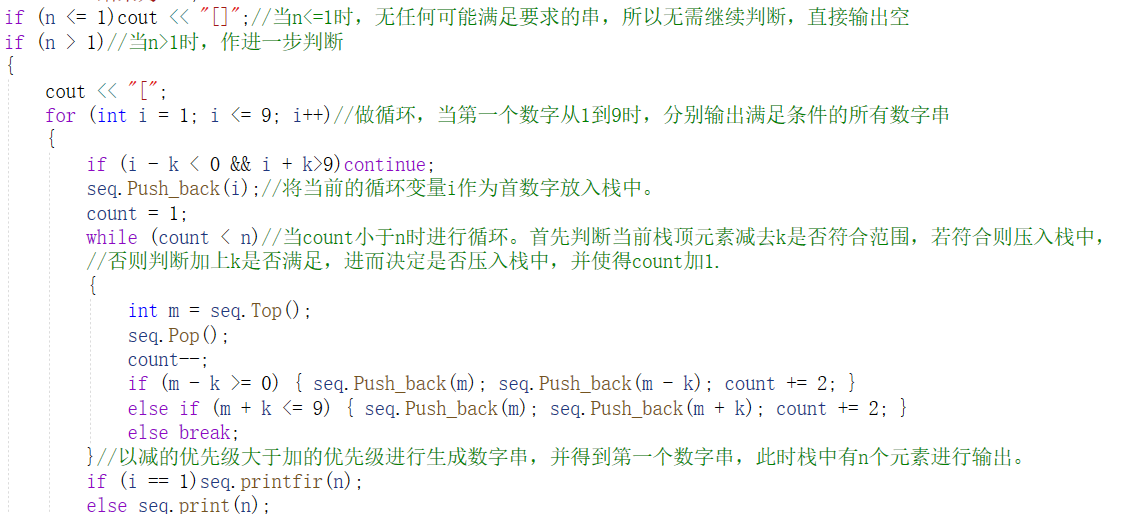
因为得到的满足条件的数字串存放在栈中，而输出时需要将数字串顺序输入，因此自定义一个Stack类，并添加需要的成员函数，以便简化后续的操作。





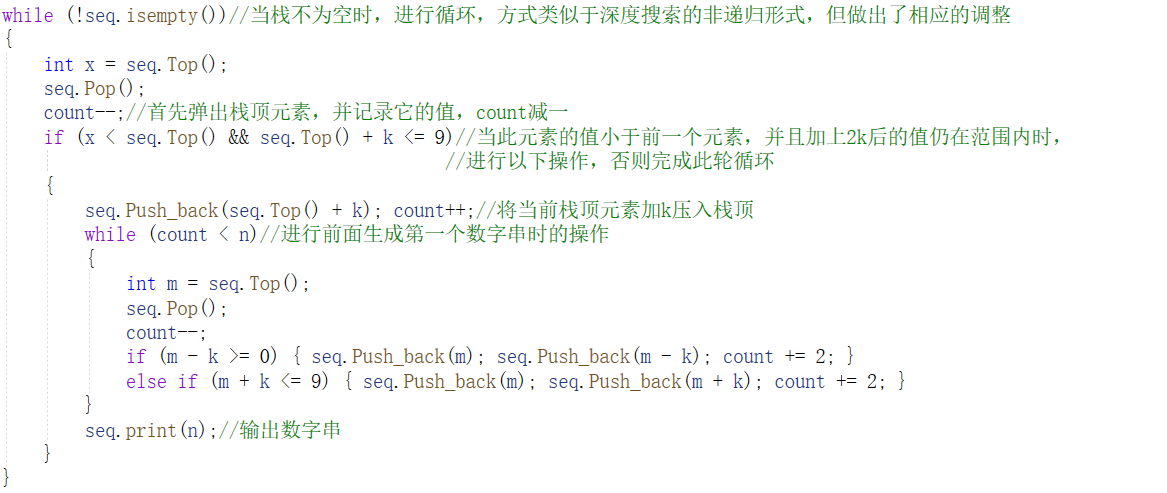
## 3.2 数字串的生成

首先，考虑到，当n=1时，必定五满足条件的数字串，因此直接输出空括号，当n>1时，进行下一步判断。



首先，有一个从1到9大的循环，考虑到，如果第一位数减去k和加上k都不满足范围，则不需要继续进行，可以开始下一轮循环。并且，如果第一位数加上k或者减去k至少有一个满足范围，则必定至少存在一个满足条件的数字串（后面的数轮流加减k即可）。

因此，当第二个数字能满足范围时，则以上述的减法优先为原则，生成第一个满足条件的数字串，并将其输出。



以上述生成的第一个数字串为基础，以深搜为思路，以减法优先为方法，进行遍历，依次输出所有满足条件的数字串。具体操作可分为以下步骤。

当栈非空时，执行以下循环：

1. 弹出栈顶的元素，并记录其值。
2. 将它与前一个数字进行比较，如果大于前一个数字，则不进行操作，直接进行下一轮循环；如果小于前一个数字，根据减法优先原则，此时前一个数字加k的情况尚未遍历到，因此将前一个数字加k压入栈中。
3. 以减法优先原则，生成数字串后面的数字，当数字串长度为n时，对数字串进行输出，并开始下一轮循环。
4. 当无满足条件的数字串存在时，栈内的数字将依次弹出，没有数字再压入，栈空，第一轮大循环结束。

当9轮大循环结束时，所有满足条件的数字串均输出，得到结果，并由用户判断是否继续运行。