

```
BASE = ord('a')
def convert(char, secret_char, secret_fac=1):
    neu_ord = (ord(char)-BASE + secret_fac*(ord(secret_char)-BASE))%26
    return chr(neu_ord + BASE)

def encrypt(text, geheim, fac=1):
    lg = len(geheim)
    res = [convert(ch, geheim[i%lg], fac) for i, ch in enumerate(text)]
    return "".join(res)

def decrypt(chiffriert, geheim):
    return encrypt(chiffriert, geheim, -1)

class UndoList(list):

    def __init__(self, seq=()):
        super().__init__(seq)
        self._undo = None

    def undo(self):
        self[:] = self._undo

    def append(self, ele):
        self._undo = self[:]
        super().append(ele)

    def extend(self, lis):
        self._undo = self[:]
        list.extend(self, lis)

    def insert(self, idx, ele):
        self._undo = self[:]
        list.insert(self, idx, ele)

    def __delitem__(self, idx):
        self._undo = self[:]
        list.__delitem__(self, idx)

    def remove(self, ele):
        self._undo = self[:]
        list.remove(self, ele)

    def __setitem__(self, idx, ele):
        self._undo = self[:]
        list.__setitem__(self, idx, ele)

def ith(gen):
    i, j = 1, 1
    for val in gen:
        if i == j:
            yield val
            i, j = 0, j+1
        i += 1

def secondonly(gen):
    first, second = set(), set()
    for ele in gen:
        if ele not in first:
            first.add(ele)
        elif ele not in second:
            second.add(ele)
        yield ele
```