

题目描述：

给你一串未加密的字符串 *str*，通过对字符串的每一个字母进行改变来实现加密，加密方式

是在每一个字母 *str[i]* 偏移特定数组元素 *a[i]* 的量，数组 *a* 前三位已经赋值：

*a[0]=1,a[1]=2,a[2]=4*。当 *i>=3* 时，数组元素 *a[i]=a[i-1]+a[i-2]+a[i-3]*，

例如：原文 *abcde* 加密后 *bdgkr*，其中偏移量分别是 *1,2,4,7,13*。

输入描述：

第一行为一个整数 *n* ( $1 \leq n \leq 1000$ )，表示有 *n* 组测试数据，每组数据包含一行，原文 *str* (只含有小写字母， $0 < \text{长度} \leq 50$ )。

输出描述：

每组测试数据输出一行，表示字符串的密文

补充说明：

解答要求

时间限制：2000ms,内存限制：64MB

示例 1

输入：

1

xy

输出：

ya

说明：

第一个字符 *x* 偏移量是 *1*，即为 *y*，第二个字符 *y* 偏移量是 *2*，即为 *a*

示例 2

输入：

2

xy

abcde

输出：

ya

bdgkr

说明：

第二行输出字符偏移量分别为 1、2、4、7、13

```
import java.util.Scanner;
```

```
import java.util.ArrayList;
```

```
import java.util.List;
```

```
// 注意类名必须为 Main, 不要有任何 package xxx 信息
```

```
public class Main {
```

```
    private static char[] chars = new  
    char[]{'a','b','c','d','e','f','g','h','i','j','k','l','m','n','o','p','q','r','s','t','u','v','w','x','y','z'};
```

```
    public static void main(String[] args) {
```

```
        Scanner scanner = new Scanner(System.in);
```

```
        int n = scanner.nextInt();
```

```
        String[] strs = new String[n];
```

```
        scanner.nextLine();
```

```
        for (int i = 0; i < n; i++) {
```

```
            String line = scanner.nextLine();
```

```
            strs[i] = line;
```

```
        }
```

```
        List<Long> cache = new ArrayList<>();
```

```
        cache.add(1L);
```

```
        cache.add(2L);
```

```
        cache.add(4L);
```

```
        for(int i = 0; i < n; i++){
```

```
            encode(cache, strs[i]);
```

```
        }
```

```
    }
```

```
    private static void encode(List<Long> cache, String src){
```

```
        if(cache.size() < src.length()){
```

```
            fullCache(cache, src.length());
```

```
        }
```

```

        StringBuilder sb = new StringBuilder();
        for(int i = 0; i < src.length(); i++){
            char c = src.charAt(i);
            int next = (int)(c - 'a' + cache.get(i) % 26);
            char res = chars[next % 26];
            sb.append(res);
        }
        System.out.println(sb.toString());
    }

    private static void fullCache(List<Long> cache, int length){
        for (int i = cache.size() - 1; i < length; i++){
            int size = cache.size();
            cache.add(cache.get(size - 1) + cache.get(size - 2) + cache.get(size - 3));
        }
    }
}

```