

字符串加密

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：2000 ms, 内存限制：64MB

题目描述：

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.*;
```

```
/**
```

```
 * @author tanlicong
```

```
 * @since 2023/6/18
```

```
 */
```

```
public class Main {
```

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

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

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

```
        int offsetSize = 50;
```

```
        List<String> strList = new ArrayList<>(n);
```

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

```
            String str = in.next();
```

```
            strList.add(str);
```

```
            offsetSize = Integer.max(offsetSize, str.length());
```

```
        }
```

```
        long[] offsetArray = initOffsetArray(offsetSize);
```

```
        strList.forEach(str -> System.out.println(encryptStr(str, offsetArray)));
```

```
    }
```

```
public static long[] initOffsetArray(int size) {
```

```
    long[] offsetArray = new long[size];
```

```
    offsetArray[0] = 1;
```

```
    offsetArray[1] = 2;
```

```
    offsetArray[2] = 4;
```

```
    offsetArray[3] = 7;
```

```
    for (int i = 3; i < size-1; i++) {
```

```
        long currentOffset = offsetArray[i];
```

```
        long nextOffset = currentOffset + currentOffset - offsetArray[i-3];
```

```
        offsetArray[i+1] = nextOffset;
```

```
    }
```

```
    return offsetArray;
```

```
}
```

```
/**
```

```
 * 加密明文返回密文
```

```
 * @param plainText 明文
```

```
 * @param offsetArray 偏移数组
```

```
 * @return 密文
```

```
 */
```

```
public static String encryptStr (String plainText, long[] offsetArray) {
```

```
    StringBuilder encryptedStr = new StringBuilder();
```

```
    for (int i = 0; i < plainText.length(); i++) {
```

```
char c = plainText.charAt(i);
```

```
char offsetChar = (char) (((c - 'a') + offsetArray[i]) % 26 + 'a');
```

```
encryptedStr.append(offsetChar);
```

```
}
```

```
return encryptedStr.toString();
```

```
}
```

```
}
```