

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：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

```
#include <stdarg.h>
```

```
#include <iostream>
```

```
#include <vector>
```

```
#include <limits>
```

```
#include <unordered_map>
```

```
#include <unordered_set>
```

```
#include <algorithm>
```

```
#include <queue>
```

```
#include <array>
```

```
#include <numeric>
```

```
using namespace std;
```

```
int main() {
```

```
    int N;
```

```
    cin >> N;
```

```
    while(N)
```

```

{

    string str;

    cin >> str;

    array<int,3> offset = {1, 2, 4};

    for (int i = 0; i < 3 && i < str.size(); i++)

        str[i] = ((str[i] + offset[i]) - 'a') % 26 + 'a';

    for (int i = 3; i < str.size(); i++){

        int cur_offset = accumulate(offset.begin(), offset.end(), 0) % 26;

        str[i] = ((str[i] + cur_offset) - 'a') % 26 + 'a';

        offset[0] = offset[1];

        offset[1] = offset[2];

        offset[2] = cur_offset;

    }

    cout << str << endl;

    N--;

}

}

```