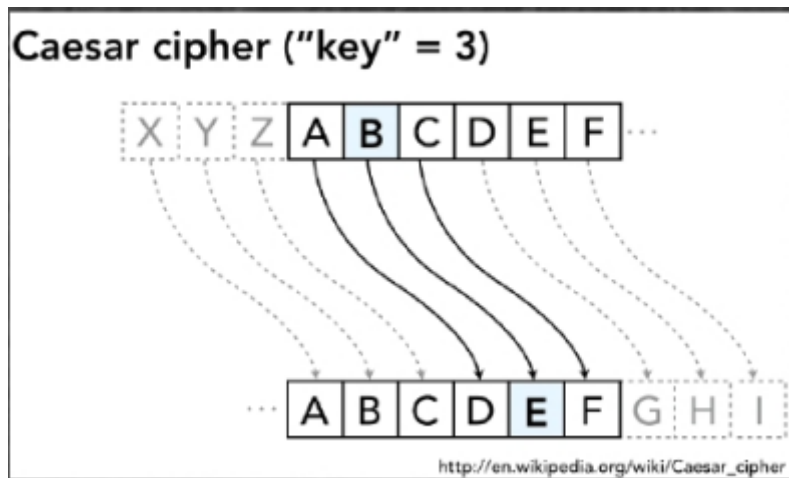


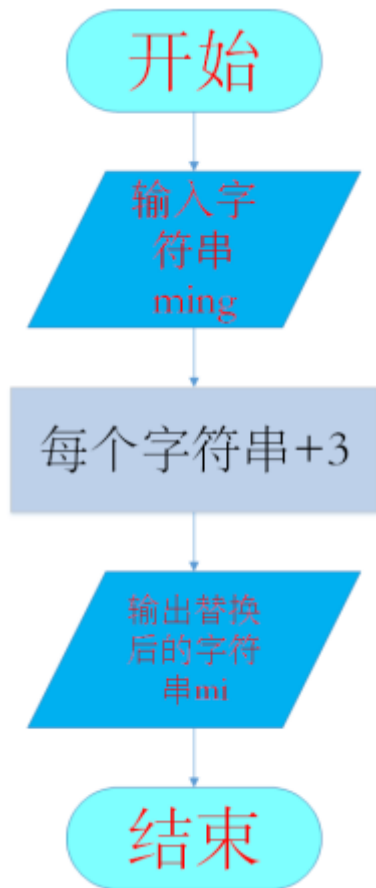
一、古罗马皇帝凯撒在打仗时曾经使用过以下方法加密军事情报：



请编写一个程序，使用上述算法加密或解密用户输入的英文字符串要求设计思想、程序流程图、源代码、结果截图。

1、程序设计思想：从键盘输入任意字符串ming，每个字符往后挪3,替换成mi,用循环算法，最终一个一个字符按序输出。

2、程序流程图：

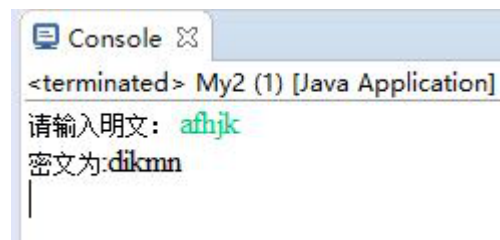


3、源代码：

```
import java.util.Scanner;
public class My2 {
    public static void main(String[] args)
    {
        Scanner in = new Scanner(System.in);
        System.out.print("请输入明文： ");
        String ming = in.nextLine();
        in.close();
        String r="";
        char mi;
        for(int i=0;i<ming.length();i++)
        {
            mi=(char)(ming.charAt(i)+3);
            r=r+mi;
        }
    }
}
```

```
    }  
    System.out.println("密文为:"+r);  
    }  
}
```

结果:



二、String.equals() 方法、整理 String 类的 Length()、charAt()、getChars()、replace()、toUpperCase()、toLowerCase()、trim()、toCharArray()使用说明、阅读笔记。

1、String.equals()方法

用来可以比较两个字符串的内容。而使用==比较两字符串变量是否引用同一字符串对象。

使用方法;

(1) String s1=**new** String("Hello");

String s2=**new** String("Hello");

System.**out**.println(s1.equals(s2));

比较构造方法的初始化对象的内容。

(2) String s3="Hello";

String s4="Hello";

System.**out**.println(s3.equals(s4));

比较两个字符串的内容。

2、Length()

获取字符串长度（字符串内包含的字符个数）。

使用方法：

(1)、String s1 = **new** String("hello");

```
System.out.println(s1.length());//6
```

(2) 、String s3="Hello";

```
System.out.println(s3.length());//6
```

3、charAt()

获取指定位置的字符。第一个字符是从0开始的。

使用方法：

```
String s3="Hello";
```

```
System.out.println(s1.charAt(1));//e
```

4、getChars(a,b,c,d)

获取从指定位置起的子串复制到字符数组中，有四个参数：

a.被拷贝字符在字符串中的起始位置(整型数字)

b.被拷贝的最后一个字符在字符串中的下标再加1（整型数字）

c.目标字符数组（数组名称）

d.拷贝的字符放在字符数组中的起始下标（整型数字）

使用方法：

```
String s1, output;  
  
char charArray[];  
  
s1 = new String( "hello there" );  
  
charArray = new char[ 5 ];  
  
s1.getChars( 0, 5, charArray, 0 );  
  
output += "\nThe character array is: ";//The character array is:hello
```

5、replace()

子串替换，`replace(char oldChar,char newChar)`：将字符串中的所有的oldChar字符替换为newChar字符。

使用方法：

```
String s1=" aBC123" ;  
  
System.out.println(s1.replace( 'C' ; '0' ));//将所有的 'C' 替换为 '0'
```

输出结果为：aB0123

6、toUpperCase()

将(当前)字符串转换为大写（英文）。

使用方法：

```
String s1=" abC123" ;  
  
System.out.println(s1.toUpperCase());//将当前字符串转换为大写
```

输出结果：ABC123

7、toLowerCase()

将(当前)字符串转换为小写（英文）。

使用方法：

```
String s1=" aBC123" ;
```

```
System.out.println(s1.toLowerCase());//将当前字符串转换为小写
```

输出结果：abc123

8、trim()

去除字符串头尾空格。

使用方法：String s1=" a b c"

```
System.out.println(s1.trim());//去除字符串头尾空格
```

输出结果：a b c

9、toCharArray()

将字符串对象转换为字符数组（空格算一个字符）

使用方法：

```
String s1=" Java" ;
```

```
char[] s2=s1.toCharArray();
```

```
System.out.println(s2);
```

输出结果：Java