

一、

1.

输出： TestTestTestTestTest （跟了一个空格）

原因： for 如果没有花括号， 默认只跟后面一个语句，  $i=0; i<5$  说明循环 5 次 Test

2.

输出： 6 8

原因：  $i++$  是先执行指令再变成  $i+1$ ,  $++i$  是指先把  $i$  变成  $i++$  再执行语句指令。

$++i < 10$ ,  $i$  此时变成 6, 之后输出 6, 输出后  $i++$ ,  $i$  变成 7 再  $++i$  变成 8 后输出

8, 输出 8 后加一变成 9,  $9+1 < 10$  不成立循环结束

3.

输出： 1 3 7 15 31 63

原因：

$i=0$ ,  $++i$  后变成 1 输出,  $i$  乘 2 变成 2;

$i=2$ ,  $++i$  后变成 3 输出,  $i$  乘 2 变成 6;

$i=6$ ,  $++i$  后变成 7 输出,  $i$  乘 2 变成 14;

$i=14$ ,  $++i$  后变成 15 输出,  $i$  乘 2 变成 30;

$i=30$ ,  $++i$  后变成 31 输出,  $i$  乘 2 变成 62;

$i=62$ ,  $++i$  后变成 63 输出,  $i$  乘 2 变成 126 大于 100 停止循环;

4.

输出： 没有输出，因为程序永远结束不了

原因： while( $i < 10$ ); 后面跟了一个分号，导致空循环，没有 cout 也没有  $++i$ 。去掉分号后，经检验是完全正确的

## 二、

①可以计算 a 的 n 次方 (n 逐渐减小, 每减 1, 结果乘以 a 一次)

②

不完美：

整数溢出：如果结果超出了 long long 类型能够表示的范围，会导致错误

太慢效率太低：一次一次乘会影响计算速度，比如 a 的四次方可以变成平方和平方相乘

func 表示不明确，最差好歹也写个“chengfang”啊

n 我觉得应该改成 unsigned int，防止出现负数情况

## 三、

① $i < 9$

② $j < i$

③ $\text{cout} << i;$

或者

① $i < 8$

② $j \leq i$

③ $\text{cout} << i + 1;$

两种发现都可以