**《高级语言程序设计》课后作业**

# 第4章 基本编程技术

院系&专业：\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 学生姓名：\_\_\_\_\_\_\_\_\_\_\_\_ 学号：\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## 填空题

1. 在编写程序寻找完全平方数、水仙花数或同构数时可以使用相同的策略，即\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_策略。

**生成与检查**

1. 程序中已在一个程序片段的前后两个位置调用标准库中的函数clock，把该函数的返回值保存在整型变量t0和t1中，请配合使用符号常量CLOCKS\_PER\_SECOND，求出以秒为单位的时间间隔（保留小数部分）：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**(double)(t1 - t0) / CLOCKS\_PER\_SECOND**

1. 使用标准库中的函数time获取当前时间，以此作为种子数，用标准库中的函数srand设定随机数种子： \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**srand(time(0));**

1. 使用标准库中的随机数生成函数rand，生成 [10, 100] 之间的整数随机数：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**rand() % 91 + 10;**

**交互式程序**

1. 在运行过程中需要由用户输入数据并输出数据给用户的程序称为\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**0**

1. 程序中使用“cin >> ”输入数据，当用户输入非法数据时，提取运算符“>>”的返回值为\_\_\_\_\_\_
2. 程序中有预处理命令“#include <sstream>”，并有语句“istringstream inss ("10 15 29 18");”，该语句的含义是：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_，然后程序中可以从inss 读取一个数据到整型变量n：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**inss >> n;**

**定义一个输入字符串流inss并绑定到字符串 "10 15 29 18"**

1. 程序中有预处理命令“#include <fstream>”，并有语句“ifstream input ("data.txt");”，该语句的含义是：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**定义一个输入文件流input并绑定到文件 data.txt**

1. 在程序中需要输入字符变量的值时，可以使用标准库函数\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_或cin的成员函数cin.get。

**getchar()**

**EOF**

1. 从输入文件流中读取字符数据到int类型变量，如果到达文件末尾，则变量得到的值为\_\_\_\_\_\_\_\_\_\_\_\_\_
2. 标准库中有一批字符分类函数，它们的名称都以“is”开头，对满足条件的字符返回非0值，否则返回0值。其返回值常被作为逻辑值使用，用于控制程序流程。名称和功能与此类似的函数被称为\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_。

**谓词函数**

## 编程题

|  |  |
| --- | --- |
| 1、一个三位的十进制整数，如果它的三个数字的立方和等于这个数的值，就称为一个“水仙花数”。例如水仙花数153 = 13 + 53 + 33。写程序找出 [100, 999] 范围内的所有水仙花数并输出。  int main() { //求水仙花数  **int i, j, k, n;**  **cout <<"水仙花数：\n";**  **for (n = 100; n < 1000; n++) {**  **i = n /100; //百位数**  **j = n /10 %10; //十位数**  **k = n % 10; //个位数**  **if (n == i \* i \* i + j \* j \* j + k \* k \* k)**  **cout << n << "\t";**  **cout << endl;**  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  return 0;  } | 2、Fibonacci数列 {*F*n} 的定义如下：*F*1 = 1, *F*2 = 1, *…… F*n = *Fn*−1 + *Fn*−2 (n > 2)。请写程序打印输出Fibonacci的前20项。  int main() { //求Fibonacci数列  **int n =20, k = 2, a = 1, b = 1, tmp;**  **cout << a << "\t" << b << "\t";**  **while (k < n) {**  **tmp = b; //暂存**  **b = b + a; //递推**  **a = tmp; //跟进**  **++k;**  **cout << b << "\t";**  **}**  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  return 0;  } |