Lab 6

Lab 6 对应 lecture notes 的 Lecture 6（函数），训练目标是熟练使用 C++函数，并利用函数编写更复杂的 C++程序。

第一部分：普通函数

Problem 1.

库函数是 C++预先写好的一些函数，供用户使用。下面是其中一些比较常用的库函数，编写程序测试这些函数并掌握其用法：

1. 数学相关函数，在使用时要包含 #include<math.h>

函数原型说明	功能
int abs(int x)	求整数 x 的绝对值
double fabs(double x)	求双精度实数 x 的绝对值
double cos(double x)	计算 cos(x)的值
double exp(double x)	求 e^x 的值
double floor(double x)	求不大于双精度实数 x 的最大整数
double log(double x)	求 ln x
double log10(double x)	求 $\log_{10}x$
double sin(double x)	计算 sin(x)的值
double sqrt(double x)	计算 x 的开方

2. 字符相关函数，使用时需要包含 #include<ctype.h>

函数原型说明	功能	返回值
int isalpha(int ch)	检查 ch 是否为字母	是，返回 1；否则返回 0
int isdigit(int ch)	检查 ch 是否为数字	是，返回 1；否则返回 0
int islower(int ch)	检查 ch 是否为小写字母	是，返回 1；否则返回 0
int isspace(int ch)	检查 ch 是否为空格、制表或换行符	是，返回 1；否则返回 0
int isupper(int ch)	检查 ch 是否为大写字母	是，返回 1；否则返回 0
int tolower(int ch)	把 ch 中的字母转换成小写字母	返回对应的小写字母
int toupper(int ch)	把 ch 中的字母转换成大写字母	返回对应的大写字母

3. 字符串函数（lab5 已经练习过），使用时需包含 #include<string.h>

4. 工具相关库函数，使用时需包含 #include<stdlib.h>

函数原型说明	功能	返回值
char *itoa(int i)	把整数 i 转换为字符串	转换后的字符串
int atoi(const char *s)	将字符串 s 转换为 int 类型	转换后的整数
int rand(void)	产生 0 ~ 32767 的随机整数	返回一个随机整数

Problem 2.

根据下面程序提示完成相关函数定义（允许对所给的程序进行修改和拓展）：

1.

```
int main() {
    double n, square;
    cin>>n;
    //编写函数求 n 的平方；
    cout<<square; //输出结果
    return 0;
}
```

2.

```
int main() {
    int a, b;
    cin>>a>>b;
    //编写函数交换 a 和 b
    cout<<a<<b;
    return 0;
}
```

3.

```
int main() {
    int a[20], n, i = 0;
    while(cin>>n) { //输入任意个整数（小于 20 个）
        a[i] = n;
        i++;
    }
    //编写函数对输入的数排序，并输出排序后的结果
    return 0;
}
```

4.

```
int main() {
    char c1[20], c2[20];
    gets(c1);
    gets(c2);
    //编写函数比较字符串 c1 和 c2 的大小，相等返回 0，c1>c2 返回 1，否则返回-1
    return 0;
}
```

5.

```
int main() {
    int a[20], n, i = 0;
    while(cin>>n) {//输入任意个整数（小于 20 个）
```

```
            a[i] = n;
            i++;
        }
        //编写 1 个函数，返回所输入数中的最大值和最小值（关键是如何同时返回两个值？）
        return 0;
}
```

第二部分：简单递归

递归就是函数自己调用自己，递归函数可以大大简化程序设计，有很广泛的用途。

Problem 3.

使用递归，计算 n 的阶乘，函数以整数 n 作为参数，返回 n 的阶乘。

Problem 4.

使用递归，计算第 n 个斐波那契数，斐波那契数列定义：$F(0)=0$，$F(1)=1$, $F(n)=F(n-1)+F(n-2)$（$n \geq 2$）；函数以 n 作为参数，输出计算结果。

Problem 5.

使用递归，判断给定字符串是否是回文串，函数以字符串作为参数，输出 True（是回文）或者 False（不是回文）。

第三部分：递归变递推

递归可以让程序看起来很简洁，但因为需要多次调用函数，其效率很低。因此，很多时候要尽量避免使用递归。思考下面题目，分别尝试递归和非递归解法，并对比各自执行时间的长短（怎样获取程序执行时间到网上自行搜索）。

Problem 6.

Pell 数列 $a_1, a_2, a_3, \ldots$的定义是这样的，$a_1 = 1, a_2 = 2, \ldots, a_n = 2 * a_{n-1} + a_{n-2}\ (n > 2)$。给出一个正整数 k，要求计算 Pell 数列的第 k 项模上 32767。

输入

第 1 行是测试数据的组数 n，后面跟着 n 行输入。每组测试数据占 1 行，包括一个正整数 k ($1 \leqslant k < 1000000$)。

输出

n 行，每行输出对应一个输入。输出应是一个非负整数。

样例输入

```
2
1
8
```

样例输出

```
1
408
```

Problem 7.
楼梯有 n(100 > n > 0)阶台阶,上楼时可以一步上 1 阶,也可以一步上 2 阶,也可以一步上 3 阶,编程计算共有多少种不同的走法。

输入
输入的每一行包括一组测试数据，即为台阶数 n。最后一行为 0，表示测试结束。

输出
每一行输出对应一行输入的结果，即为走法的数目。

样例输入

```
1
2
3
4
0
```

样例输出

```
1
2
4
7
```

Problem 8.
有一批易感人群住在网格状的宿舍区内，宿舍区为 n*n 的矩阵，每个格点为一个房间，房间里可能住人，也可能空着。在第一天，有些房间里的人得了流感，以后每天，得流感的人会使其邻居传染上流感，(已经得病的不变)，空房间不会传染。请输出第 m 天得流感的人数。

输入
第一行一个数字 n，n 不超过 100，表示有 n*n 的宿舍房间。
接下来的 n 行，每行 n 个字符，'.'表示第一天该房间住着健康的人，'#'表示该房间空着，'@'表示第一天该房间住着得流感的人。
接下来的一行是一个整数 m，m 不超过 100.

输出
输出第 m 天得流感的人数

样例输入

```
5
....#
.#.@.
.#@..
#....
.....
```

```
4
```

样例输出

```
16
```

第四部分：复杂递归（回溯）

认真学习回溯算法的 PPT，完成以下程序设计。

Problem 9.

在国际象棋棋盘上放置八个皇后，要求每两个皇后之间不能直接吃掉对方。

输入

无输入。

输出

按给定顺序和格式输出所有八皇后问题的解。

样例输入

样例输出

```
No. 1
1 0 0 0 0 0 0 0
0 0 0 0 0 0 1 0
0 0 0 0 1 0 0 0
0 0 0 0 0 0 0 1
0 1 0 0 0 0 0 0
0 0 0 1 0 0 0 0
0 0 0 0 0 1 0 0
0 0 1 0 0 0 0 0
No. 2
1 0 0 0 0 0 0 0
0 0 0 0 0 0 1 0
0 0 0 1 0 0 0 0
0 0 0 0 0 1 0 0
0 0 0 0 0 0 0 1
0 1 0 0 0 0 0 0
0 0 0 0 1 0 0 0
0 0 1 0 0 0 0 0
No. 3
1 0 0 0 0 0 0 0
0 0 0 0 0 1 0 0
0 0 0 0 0 0 0 1
0 0 1 0 0 0 0 0
0 0 0 0 0 0 1 0
0 0 0 1 0 0 0 0
```

```
0 1 0 0 0 0 0 0
0 0 0 0 1 0 0 0
No. 4
1 0 0 0 0 0 0 0
0 0 0 0 1 0 0 0
0 0 0 0 0 0 0 1
0 0 0 0 0 1 0 0
0 0 1 0 0 0 0 0
0 0 0 0 0 0 1 0
0 1 0 0 0 0 0 0
0 0 0 1 0 0 0 0
No. 5
0 0 0 0 0 1 0 0
1 0 0 0 0 0 0 0
0 0 0 0 1 0 0 0
0 1 0 0 0 0 0 0
0 0 0 0 0 0 0 1
0 0 1 0 0 0 0 0
0 0 0 0 0 0 1 0
0 0 0 1 0 0 0 0
以下省略
```

Problem 10.
当你站在一个迷宫里的时候，往往会被错综复杂的道路弄得失去方向感，如果你能得到迷宫地图，事情就会变得非常简单。
假设你已经得到了一个 n*m 的迷宫的图纸，请你找出从起点到出口的最短路。

输入

第一行是两个整数 n 和 m(1<=n,m<=100)，表示迷宫的行数和列数。
接下来 n 行，每行一个长为 m 的字符串，表示整个迷宫的布局。字符'.'表示空地，'#'表示墙，'S'表示起点,'T'表示出口。

输出
输出从起点到出口最少需要走的步数。

样例输入

```
3 3
S#T
.#.
...
```

样例输出

```
6
```

Problem 11.
马在中国象棋以日字形规则移动。
请编写一段程序，给定 n*m 大小的棋盘，以及马的初始位置(x，y)，要求不能重复经过棋盘上的同一个点，计算马可以有多少途径遍历棋盘上的所有点。

输入

第一行为整数 T(T < 10)，表示测试数据组数。
每一组测试数据包含一行，为四个整数，分别为棋盘的大小以及初始位置坐标 n,m,x,y。(0<=x<=n-1,0<=y<=m-1, m < 10, n < 10)

输出

每组测试数据包含一行，为一个整数，表示马能遍历棋盘的途径总数，0 为无法遍历一次。

样例输入

```
1
5 4 0 0
```

样例输出

```
32
```