**第1关：求前n项之和**

**本关任务：编写递归函数，通过递归调用求前n项之和**

**测试输入：2 预期输出：3**

**测试输入：10 预期输出：55**

**// 用递归调用求前n个自然数之和**

**#include <iostream>**

**using namespace std;**

**// 返回前n个自然数之和**

**int sumn(int n);**

**int main()**

**{**

**int n;**

**cin >> n;**

**cout<<sumn(n);**

**return 0;**

**}**

**int sumn(int n)**

**{**

**//TODO-BEGIN  通过递归调用计算前n个数之和**

**if(n==1)**

**return 1;**

**else**

**return sumn(n-1)+n;**

**//TODO-END**

**}**

**第2关：求x的n次幂**

**本关任务：编写递归函数，通过递归调用求x的n次幂**

**根据提示，在右侧编辑器补充代码，通过递归调用求x的n次幂**

**测试输入：2 3 预期输出：8**

**测试输入：5 2 预期输出：25**

**// 用递归调用求幂**

**#include <iostream>**

**using namespace std;**

**// 返回x的n次幂**

**float powern(float x, int n);**

**int main()**

**{**

**float x;**

**int n;**

**cin >> x >> n;**

**cout<<powern(x, n);**

**return 0;**

**}**

**float powern(float x, int n)**

**{**

**//TODO -BEGIN 通过递归调用计算x的n次方**

**if(n==1)**

**return x;**

**else return x\*powern(x,n-1);**

**//TODO-END**

**}**

**第3关：求字母序号**

**本关任务：编写递归函数，通过递归调用求用户输入的字母在字母表中的序号**

**测试输入：A 预期输出：1**

**测试输入：a 预期输出：1**

**// 利用递归调用求字母的序号**

**// 大写A和小写a的序号均为1**

**#include <iostream>**

**using namespace std;**

**//返回字母ch的序号**

**int rank\_of\_alphabet(char ch);**

**int main()**

**{**

**char ch;**

**// 若输入不是字母，则重新输入**

**do**

**{**

**cin >> ch;**

**}while((ch<'A' || ch > 'Z') && (ch < 'a' || ch > 'z'));**

**cout << rank\_of\_alphabet(ch);**

**return 0;**

**}**

**int rank\_of\_alphabet(char ch)**

**{**

**//TODO  通过递归调用返回字母表中的序号**

**if (ch>='A'&&ch<='Z') return ch-'A'+1;**

**else return ch-'a'+1;**

**//TODO-END**

**}**

**第4关：逆序输出**

**本关任务：编写递归函数，通过递归调用，将一整数逆序输出。**

**测试输入：1234 预期输出：4321**

**测试输入：123400 预期输出：004321**

**#include <iostream>**

**using namespace std;**

**/\*函数reverse\_out：逆序输出整数x的各位\*/**

**void reverse\_out(int x);**

**int main()**

**{**

**int n;**

**cin >> n;**

**reverse\_out(n);**

**return 0;**

**}**

**void reverse\_out(int x)**

**{**

**// TODO -Begin**

**if(x==0);**

**else{**

**cout<<x%10;**

**reverse\_out(x/10);**

**}**

**//TODO-END**

**}**