**5-9   判断满足条件的三位数 (15分)**

本题要求实现一个函数，统计给定区间内的三位数中有两位数字相同的完全平方数（如144、676）的个数。

### 函数接口定义：

**int** **search**( **int** n );

其中传入的参数int n是一个三位数的正整数（最高位数字非0）。函数search返回[101, n]区间内所有满足条件的数的个数。

### 裁判测试程序样例：

**#include <stdio.h>**

**#include <math.h>**

**int** **search**( **int** n );

**int** **main**(){

**int** number;

scanf("%d",&number);

printf("count=%d\n",search(number));

**return** 0;

}

*/\* 你的代码将被嵌在这里 \*/*

### 输入样例：

500

### 输出样例：

count=6

**5-10 递归求阶乘和 (15分)**

本题要求实现一个简单的递归函数，计算非负整数阶乘，并利用该函数求 1!+2!+3!+...+n! 的值。

### 函数接口定义：

**double** **fact**( **int** n );

**double** **factsum**( **int** n );

函数fact应返回n的阶乘。函数factsum应返回 1!+2!+...+n! 的值。题目保证输入输出在双精度范围内。

### 裁判测试程序样例：

**#include <stdio.h>**

**double** **fact**( **int** n );

**double** **factsum**( **int** n );

**int** **main**(){

**int** n;

scanf("%d",&n);

printf("fact(%d) = %.0f\n", n, fact(n));

printf("sum = %.0f\n", factsum(n));

**return** 0;

}

*/\* 你的代码将被嵌在这里 \*/*

### 输入样例1：

10

### 输出样例1：

fact(10) = 3628800

sum = 4037913

### 输入样例2：

0

### 输出样例2：

fact(0) = 1

sum = 0

**5-11  递归实现指数函数 (15分)**

本题要求实现一个计算*x的n次方*（*n*≥1）的递归函数。

### 函数接口定义：

**double** **calc\_pow**( **double** x, **int** n );

函数calc\_pow应返回x的n次幂的值。题目保证结果在双精度范围内。

### 裁判测试程序样例：

**#include <stdio.h>**

**double** **calc\_pow**( **double** x, **int** n );

**int** **main**(){

**double** x;

**int** n;

scanf("%lf %d", &x, &n);

printf("%.0f\n", calc\_pow(x, n));

**return** 0;

}

*/\* 你的代码将被嵌在这里 \*/*

### 输入样例：

3 4

### 输出样例：

81

**5-12 递归求Fabonacci数列 (10分)**

本题要求递归实现求Fabonacci数列项的函数。Fabonacci数列的定义如下：

*f*(*n*)=*f*(*n*−2)+*f*(*n*−1) (*n*≥2)，其中*f*(0)=0，*f*(1)=1。

### 函数接口定义：

**int** **f**( **int** n );

函数f应返回第n个Fabonacci数。题目保证输入输出在长整型范围内。建议用递归实现。

### 裁判测试程序样例：

**#include <stdio.h>**

**int** **f**( **int** n );

**int** **main**(){

**int** n;

scanf("%d", &n);

printf("%d\n", f(n));

**return** 0;

}

*/\* 你的代码将被嵌在这里 \*/*

### 输入样例：

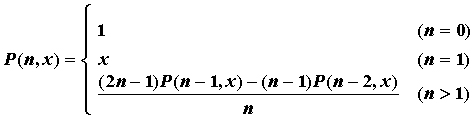
6

### 输出样例：

8

**5-13 递归计算P函数 (15分)**

本题要求递归实现下列函数*P*(*n*,*x*)的计算，其函数定义如下：



### 函数接口定义：

**double** **P**( **int** n, **double** x );

其中n是用户传入的非负整数，x是双精度浮点数。函数P返回*P*(*n*,*x*)函数的相应值。题目保证输入输出都在双精度范围内。

### 裁判测试程序样例：

**#include <stdio.h>**

**double** **P**( **int** n, **double** x );

**int** **main**(){

**int** n;

**double** x;

scanf("%d %lf", &n, &x);

printf("%.2f\n", P(n,x));

**return** 0;

}

*/\* 你的代码将被嵌在这里 \*/*

### 输入样例：

10 1.7

### 输出样例：

3.05

**5-14 数字金字塔 (15分)**

本题要求实现函数输出n行数字金字塔。

### 函数接口定义：

**void** **pyramid**( **int** n );

其中n是用户传入的参数，为[1, 9]的正整数。要求函数按照如样例所示的格式打印出n行数字金字塔。注

意每个数字后面跟一个空格。

### 裁判测试程序样例：

**#include <stdio.h>**

**void** **pyramid**( **int** n );

**int** **main**(){

**int** n;

scanf("%d", &n);

pyramid(n);

**return** 0;

}

*/\* 你的代码将被嵌在这里 \*/*

### 输入样例：

5

### 输出样例：

1

2 2

3 3 3

4 4 4 4

5 5 5 5 5

5-15 **使用函数输出水仙花数 (20分)**

本题要求编写两个函数，一个判断给定整数是否水仙花数，另一个按从小到大的顺序打印出给定区间(*m*,*n*)内所有的水仙花数。水仙花数是指一个*N*位正整数（*N*≥3），它的每个位上的数字的*N*次幂之和等于它本身。例如：153=1​3​​+5​3​​+3​3​​。

### 函数接口定义：

**int** **narcissistic**( **int** number );

**void** **PrintN**( **int** m, **int** n );

函数narcissistic判断number是否为水仙花数，是则返回1，否则返回0。

函数PrintN则打印开区间(m, n)内所有的水仙花数，每个数字占一行。题目保证100<m≤n<1000。

### 裁判测试程序样例：

**#include <stdio.h>**

**int** **narcissistic**( **int** number );

**void** **PrintN**( **int** m, **int** n );

**int** **main**(){

**int** m, n;

scanf("%d %d", &m, &n);

**if** ( narcissistic(m) ) printf("%d is a narcissistic number\n", m);

PrintN(m, n);

**if** ( narcissistic(n) ) printf("%d is a narcissistic number\n", n);

**return** 0;

}

*/\* 你的代码将被嵌在这里 \*/*

### 输入样例：

153 400

### 输出样例：

153 is a narcissistic number

370

371

5-16 **用函数输出指定范围内的Fibonacci数 (20分)**

本题要求实现一个计算Fibonacci数的简单函数，并利用其实现另一个函数，输出两正整数*m*和*n*（0<*m*≤*n*≤10000）之间的所有Fibonacci数。所谓Fibonacci数列就是满足任一项数字是前两项的和（最开始两项均定义为1）的数列。

### 函数接口定义：

**int** **fib**( **int** n );

**void** **PrintFN**( **int** m, **int** n );

其中函数fib须返回第n项Fibonacci数；函数PrintFN要在一行中输出给定范围[m, n]内的所有Fibonacci数，相邻数字间有一个空格，行末不得有多余空格。如果给定区间内没有Fibonacci数，则输出一行“No Fibonacci number”。

### 裁判测试程序样例：

**#include <stdio.h>**

**int** **fib**( **int** n );

**void** **PrintFN**( **int** m, **int** n );

**int** **main**(){

**int** m, n, t;

scanf("%d %d %d", &m, &n, &t);

printf("fib(%d) = %d\n", t, fib(t));

PrintFN(m, n);

**return** 0;

}

*/\* 你的代码将被嵌在这里 \*/*

### 输入样例1：

20 100 7

### 输出样例1：

fib(7) = 13

21 34 55 89

### 输入样例2：

2000 2500 8

### 输出样例2：

**fib**(8) = 21

No Fibonacci number

5-17 **使用函数求特殊a串数列和 (20分)**

给定两个均不超过9的正整数*a*和*n*，要求编写函数求*a*+*aa*+*aaa*++⋯+*aa*⋯*a*（*n*个*a*）之和。

### 函数接口定义：

**int** **fn**( **int** a, **int** n );

**int** **SumA**( **int** a, **int** n );

其中函数fn须返回的是n个a组成的数字；SumA返回要求的和。

### 裁判测试程序样例：

**#include <stdio.h>**

**int** **fn**( **int** a, **int** n );

**int** **SumA**( **int** a, **int** n );

**int** **main**(){

**int** a, n;

scanf("%d %d", &a, &n);

printf("fn(%d, %d) = %d\n", a, n, fn(a,n));

printf("s = %d\n", SumA(a,n));

**return** 0;

}

*/\* 你的代码将被嵌在这里 \*/*

### 输入样例：

2 3

### 输出样例：

fn(2, 3) = 222

s = 246

5-18 **函数验证哥德巴赫猜想 (20分)**

本题要求实现一个判断素数的简单函数，并利用该函数验证哥德巴赫猜想：任何一个不小于6的偶数均可表示为两个奇素数之和。素数就是只能被1和自身整除的正整数。注意：1不是素数，2是素数。

### 函数接口定义：

**int** **prime**( **int** p );**void** **Goldbach**( **int** n );

其中函数prime当用户传入参数p为素数时返回1，否则返回0；函数Goldbach按照格式“n=*p*+*q*”输出n的素数分解，其中*p*≤*q*均为素数。又因为这样的分解不唯一（例如24可以分解为5+19，还可以分解为7+17），要求必须输出所有解中*p*最小的解。

### 裁判测试程序样例：

**#include <stdio.h>**

**#include <math.h>**

**int** **prime**( **int** p );**void** **Goldbach**( **int** n );

**int** **main**(){

**int** m, n, i, cnt;

scanf("%d %d", &m, &n);

**if** ( prime(m) != 0 ) printf("%d is a prime number\n", m);

**if** ( m < 6 ) m = 6;

**if** ( m%2 ) m++;

cnt = 0;

**for**( i=m; i<=n; i+=2 ) {

Goldbach(i);

cnt++;

**if** ( cnt%5 ) printf(", ");

**else** printf("\n");

}

**return** 0;

}

*/\* 你的代码将被嵌在这里 \*/*

### 输入样例：

89 100

### 输出样例：

89 is a prime number

90=7+83, 92=3+89, 94=5+89, 96=7+89, 98=19+79

100=3+97,