第5章 函　　数

5.1　相关知识点

1. 函数的概述

(1) 函数是C语言中模块化程序设计的最小单位，是构成C程序的基本模块。一个C程序可以由一个或多个源程序文件组成，一个源程序文件又可以由一个或多个函数组成。

(2) C语言程序的执行从main()函数开始，无论调用多少函数，最终在main()函数中结束整个程序的运行。

(3) C程序的所有函数之间都是平行关系，不存在函数的嵌套定义。

(4) 从用户的角度对函数分类：

① 标准库函数：由编译系统提供；

② 用户自定义函数：解决用户的专门需要。

(5) 从函数的参数角度对函数分类：

① 无参数函数：函数定义与调用时不涉及参数，只用于执行指定的一组操作；

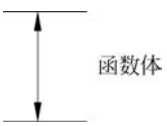
② 有参数函数：主调函数可以将数据传给被调用函数使用，被调用函数中的数据也可以带回给主调函数使用。

2. 函数的定义

函数的定义是指对函数功能的确立，包括定义函数名、函数返回值类型、函数形参及其类型、函数体等，和使用变量名一样，函数在使用之前必须先定义。

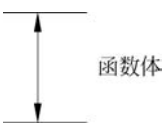
(1) 无参数函数定义的基本格式。

```
返回值类型　函数名()
{
    声明语句序列
    可执行语句序列
}
```



(2) 有参数函数定义的基本格式。

```
返回值类型　函数名(类型 形式参数1,类型 形式参数2,…)
{
    声明语句序列
    可执行语句序列
}
```



(3) 函数定义的说明：

① 函数名是函数的唯一标识，用于说明函数的功能，代表此函数在内存中的起始位置。函数名标识符的命名规则与变量的命名规则相同；

② 函数体必须用一对花括号包围，花括号{}是函数体的定界符。在函数内部定义的变量只能在函数体内访问，称为内部变量；

③ 函数头部参数表里的变量，称为形式参数(简称形参)，也是内部变量，即只能在函数体内访问，形式参数的个数和类型均由函数的功能来决定，形参变量只有在函数被调用时才占用内存空间，调用结束后所占空间即被释放。实参对形参的传递数据是单向传递(值传递)；

④ 可以定义空函数，即函数体无任何语句；

⑤ 函数的类型是指函数返回值的类型(缺省时为整型)。若不返回任何值，则应将其类型定义为 void 类型，它告诉编译器，该函数不接收来自调用程序的任何数据；

⑥ 函数的返回值是通过函数体中的 return 语句来完成的，return 语句用来指明函数将返回给主调函数的值是什么，只要执行到它，就立刻返回函数的调用者，return 后的表达式值即函数的返回值。注意，函数中的 return 语句可以有多个，但执行到一个 return 语句即立刻返回函数调用者，并且 return 语句一次也只能返回一个值，其返回值的数据类型应与函数定义时的返回值类型一致，如不一致，以函数定义的返回值类型为准。

3. 函数调用的一般形式

(1) 函数调用的一般形式：函数名(实参表列)。

(2) 函数调用方式：

① 函数语句：将函数调用单独作为一条语句，如 printf("book")；

② 函数表达式：函数调用出现在另一个表达式中，如 c=2 * max(a,b)；

③ 函数参数：函数调用作为另一个函数调用时的参数，如 m=max(a,max(b,c))。

(3) 函数调用时，实参的数量必须与形参相等，它们的类型必须匹配。

4. 对被调用函数的声明

(1) 被调用函数必须是一个已经存在的函数。

(2) 如果使用库函数应在文件头加上 #include 命令，以便将有关的库函数所在的头文件包含到本源程序文件中来。

(3) 对于用户自定义函数，函数的定义部分应出现在该函数被调用之前。否则，在调用函数之前应作引用性声明。

(4) 引用性声明的方法：

```
返回值类型  函数名(形式参数定义表);
```

如：int add(int x,int y)；

函数声明中形式参数 x,y 的变量名可以省略，但形式参数的类型不能省略。

5. 函数的嵌套调用

所谓函数的嵌套调用是指一个函数在被调用时其本身又调用了其他函数。

6. 函数的递归调用

在调用一个函数的过程中，直接或间接地调用函数自身叫作函数的递归调用，分为直接

递归调用和间接递归调用。不论是直接递归调用还是间接递归调用,必须有一个使调用终止的条件,不然的话调用将陷入无终止状态。

7. 变量的作用域

程序中被花括号括起来的区域叫做语句块,变量的作用域规则是:每个变量仅在定义它的语句块(包含下级语句块)内有效。

(1) 局部变量:在一个语句块内定义的变量,称为局部变量,其作用范围为该语句块内部。主函数中定义的变量也只在主函数中有效;不同的函数中可以定义相同的变量名,它们代表不同的局部变量,系统为其分配的内存地址是不相同的;形参也是局部变量,在定义它的函数内有效。

(2) 全局变量:不在任何语句块内定义的变量称为外部变量(或叫做全局变量)。其作用域为整个程序。如果想在全局变量的定义点之前引用该全局变量,需要用关键字 extern 作提前引用说明。全局变量可以被本源程序文件的所有函数共享,一个函数对全局变量的值的改变将会影响到其他函数对该变量的引用;当全局变量名与局部变量名相同时,则在该局部变量的有效范围内全局变量被屏蔽。

8. 变量的存储类型

变量定义的一般形式为:存储类型 数据类型 变量名;

(1) 用户使用的内存空间分为:

① 程序区:存放程序的代码;

② 常量存储区:存放程序中的常量;

③ 静态存储区:存放全局变量和静态的局部变量;

④ 动态存储区:存放函数的局部变量、函数的形参变量、函数调用时的现场保护和返回地址等。

(2) 变量的存储类型:指编译器为变量分配内存的方式。

① 自动变量(auto):语句块中定义的变量不作特殊说明都为自动局部变量,存储在动态存储区,当语句块调用结束后,它们所占用的存储空间即被释放,例如函数内部定义的变量就是局部变量,每次进入函数时都为其重新分配内存空间,函数结束时,释放为其分配的空间。自动变量在定义的时候不会自动初始化,因此,如果不对其赋初值,则它的值是一个不确定的值;

② 静态变量(static):存储在静态存储区。函数内定义的静态变量称为静态局部变量,静态局部变量只能在定义它的函数内被访问,但其值在函数调用结束后不消失而保留原值。局部静态变量是在编译时赋初值的,在定义时如果不赋初值,编译时系统自动赋初值 0;

③ 寄存器变量(register):寄存器变量就是用寄存器存储的变量,现代编译器能自动优化程序,自动把普通变量优化为寄存器变量,一般无须特别声明变量为 register;

④ 外部变量(extern):即全局变量,在所有函数的外部定义的变量,它可以被程序中的所有函数所引用,但如果要在定义点之前或者在其他文件中使用它,需要使用关键字 extern 对其进行引用性说明。外部变量保存在静态存储区中,在程序运行期间分配固定的存储单元,其生存期是整个程序的运行期。没有显示初始化的外部变量由编译程序自动初始化为 0。

9. 内部函数与外部函数

(1) 内部函数：在函数定义时加上 static，即

```
static 返回值类型　函数名(形参表)
```

内部函数又称为静态函数，这样的函数只限在所在的文件中调用。

(2) 外部函数：在函数定义时加上 extern，即

```
extern 返回值类型　函数名(形参表)
```

函数被冠以 extern 说明函数为外部函数，可以被其他文件中的函数所调用，当一个函数在定义时未说明 static 时，隐含的类型为 extern。

5.2 实 验 目 的

(1) 掌握 C 语言中定义函数的方法。

(2) 掌握函数间参数传递和返回值传递的方法。

(3) 掌握函数嵌套调用和递归调用的方法。

5.3 实 验 内 容

5.3.1 程序设计

1. 编制程序

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。例如 6＝3＋3，8＝3＋5，…，18＝5＋13。将 6～100 的偶数都表示成两个奇素数之和，输出时一行打印 5 组。

【指导】

这个问题是德国数学家哥德巴赫(C. Goldbach，1690—1764)于 1742 年 6 月 7 日在给大数学家欧拉的信中提出的，所以被称作哥德巴赫猜想。同年 6 月 30 日，欧拉在回信中认为这个猜想可能是真的，但他无法证明。现在，哥德巴赫猜想的一般提法是：每个大于等于 6 的偶数，都可表示为两个奇素数之和；每个大于或等于 9 的奇数，都可表示为三个奇素数之和，其实，后一个命题就是前一个命题的推论，18 和 19 世纪，所有的数论专家对这个猜想的证明都没有作出实质性的推进，直到 20 世纪才有所突破。1966 年，我国年轻的数学家陈景润，在经过多年潜心研究之后，成功地证明了"1＋2"，也就是“任何一个大偶数都可以表示成一个素数与另一个素因子不超过 2 个的数之和”。这是迄今为止，这一研究领域最佳的成果，距摘取这颗“数学王冠上的明珠”仅一步之遥，在世界数学界引起了轰动。"1＋2" 也被誉为陈氏定理。

数学上证明哥德巴赫猜想很难，但利用计算机的强大计算能力验证哥德巴赫猜想却很容易，用穷举算法可以对一个不小于 6 的偶数进行验证，算法是：对不小于 6 的偶数 n，x 从最小奇素数 3 开始，判断 x 是否为素数，如果 x 为素数，则 y＝6－x，再判断 y 是否是素数，如果是，则找到。程序需要多次判断一个数是否为素数，所以设计 prime()函数来判断一个数是否为素数，是则返回值 1，否则返回值 0。

【流程图】

prime()函数流程图见图 5.1。

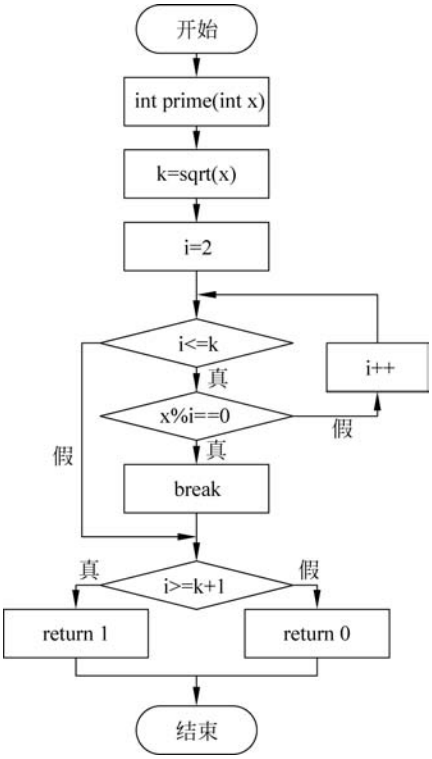


图 5.1　prime()函数流程图

主函数流程图见图 5.2。

【参考程序】

```
#include<stdio.h>
#include<math.h>

int prime(int x)        //判断 x 是否为素数,是返回值 1,否则返回值 0
{
    int i,j,k;
    k=sqrt(x);
    for(i=2;i<=k;i++)
    {
        if(x%i==0)
            break;
    }
    if(i>=k+1)
        return 1;
    else
        return 0;
}
```

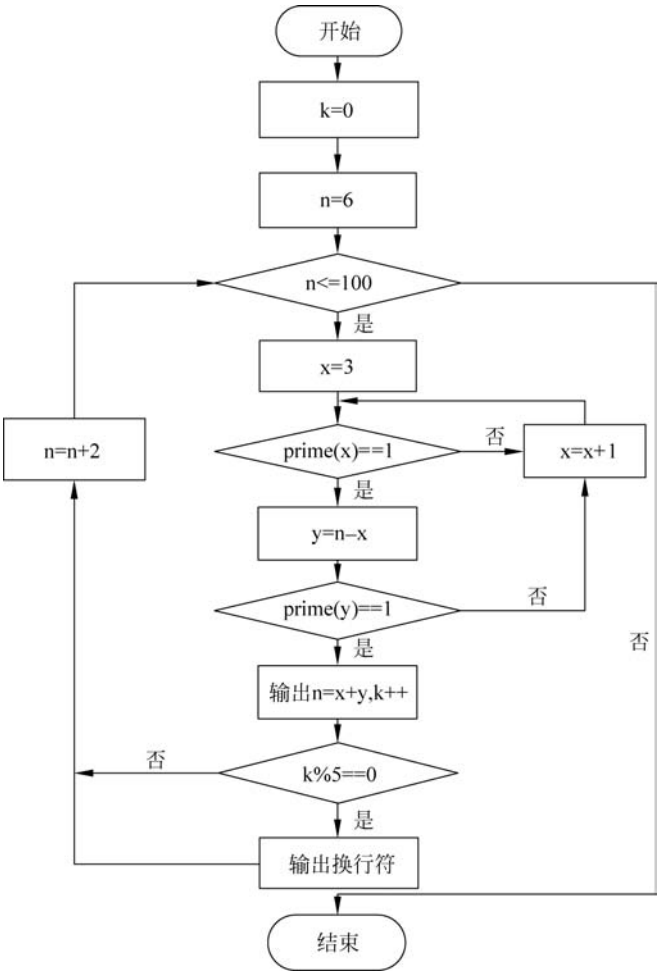


图 5.2　主函数流程图

```
int main()
{
    int n,i,j,k = 0,x,y;
    for(n = 6;n < = 100;n = n + 2)          //从 6 开始对每个偶数求解
    {
        for(x = 3;;x = x + 2)               //从最小奇素数 3 开始,对每个奇数进行试验
        {
            if(prime(x) == 1)               //调用函数判断 x 是否为素数
            {
                y = n - x;                  //如果 x 为素数,则求 y
                if(prime(y) == 1)           //调用函数判断 y 是否为素数
                {
                    printf(" % 2d = % 2d + % 2d ",n,x,y);     //x,y 均为素数,输出
                    k++;
                    if(k % 5 == 0)          //用 k 作为计数器,来判断是否换行
                        printf("\n");
```

```
                    break;      //如果 x,y 均为素数,输出后跳出循环,求解下 1 个偶数
                }
            }
        }
    }
    return 0;
}
```

【说明】

主函数中的第二个 for 循环语句,没有循环条件,意味着永远为真,但程序不会出现死循环,原因是不小于 6 的偶数肯定能分解为两个素奇数之和,执行完输出 x,y 的值后会执行 break 语句来跳出循环。大家可以试着把不小于 6 的偶数分解为两个奇素数之和的功能写成 1 个独立的函数。

2. 编制程序

请编制程序计算飞机超重行李费用。每位旅客的免费行李额:持成人或儿童客票的头等舱(舱位代码为 F)旅客为 40kg,公务舱(舱位代码为 C)旅客为 30kg,经济舱旅客(舱位代码为 Y)为 20kg。搭乘同一航班前往同一目的地的两个(含)以上的同行旅客,如在同一时间、同一地点办理行李托运手续,其免费行李额可以按照各自的客票价等级标准合并计算,超重行李费率以每公斤按超重行李票填开当日所适用的单程直达经济舱正常票价的 1.5% 计算,收费总金额以元为单位,尾数四舍五入。现要求设计一个函数求超重行李需要的费用,在主函数中输入旅客的人数,舱位(假设同行的旅客的舱位相同),行李重量,经济舱正常票价,在主函数中输出超重行李费用。(头等舱舱位代码为 F、公务舱舱位代码为 C、经济舱舱位代码为 Y。)

【指导】

该题目比较简单,除主函数外,还设计了两个函数,rounding()函数是对一个浮点数进行四舍五入操作,overweight()函数是求超重行李的费用。我们将不同舱位旅客所能携带的免费行李额和超重行李费率定义为符号常量。

【流程图】

rounding()函数流程图见图 5.3。

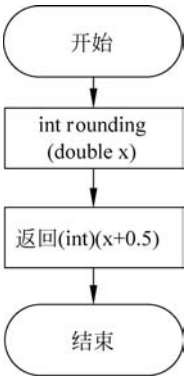


图 5.3　rounding()函数流程图

overweight()函数流程图见图 5.4。

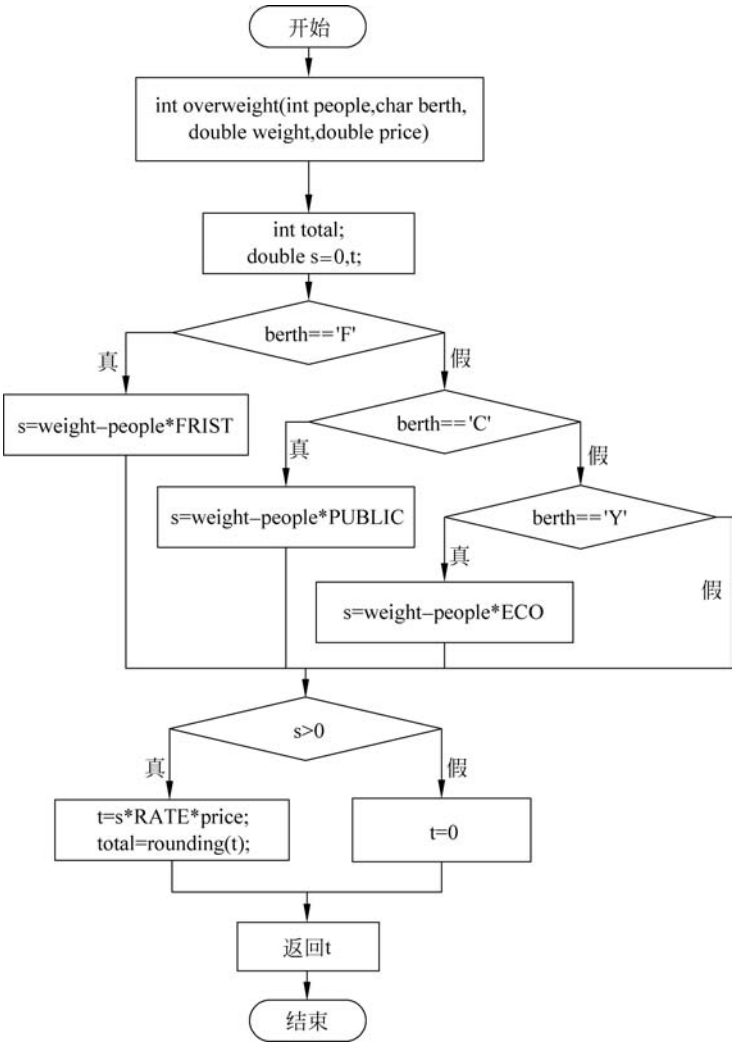


图 5.4　overweight()函数流程图

主函数流程图很简单,所以省略。

【参考程序】

```
#include <stdio.h>
#define FRIST 40
#define PUBLIC 30
#define ECO 20
#define RATE 0.015
int rounding(double x)              //四舍五入
{
    return (int)(x+0.5);
}

int overweight(int people,char berth,double weight,double price)  //计算超重行李
```

```
{
    int total;
    double s = 0,t;
    if(berth == 'F')s = weight - people * FRIST;
    else if(berth == 'C')s = weight - people * PUBLIC;
    else if(berth == 'Y')s = weight - people * ECO;
    if(s > 0)
    {
        t = s * RATE * price;
        total = rounding(t);
    }
    else
        total = 0;
    return(total);
}

int main()
{
    int people,total;
    char berth;
    double weight, price;
    printf("请输入人数 舱位 行李总重量 经济舱标准价格\n");
    scanf("%d %c %lf %lf",&people,&berth,&weight,&price);
    total = overweight(people,berth, weight, price);
    printf("\n超重行李费用为:%d\n",total);
}
```

5.3.2 程序填空

给定程序中,函数 fun()的功能是:将形参 n 中,每一位数字是偶数的取出,并按原来从高位到低位的顺序组成一个新的数,并作为函数值返回。

例如,从主函数输入一个整数:27638496,函数返回值为:26846。请在程序的下画线处填入正确的内容并删除下画线,使程序得出正确的结果。注意不得增行或删行,也不得更改程序的结构!

【程序填空】

```
#include <stdio.h>
unsigned long fun(unsigned long n)
{
    unsigned long x = 0, s, i;
    int t;
    s = n;
    i = ____【1】____;
    while(____【2】____)
    {
        t = s % 10;
        if(t % 2 == 0)
        {
            x = x + t * i;
```

```
            i = ____【3】____;
        }
        s = s/n;
    }
    return x;
}

int main()
{
    unsigned long n = -1;
    while(n > 99999999||n < 0)
    {
        printf("Please input(0 < n < 100000000): ");
        scanf("%ld",&n);
    }
    printf("\nThe result is: %ld\n", ____【4】____);
    return 0;
}
```

【参考答案】

【1】1

【2】s 或 s>0 或 s!=0

【3】i * 10

【4】fun(n)

【程序分析】

本题主函数中输入一个无符号长整型数 n,将 n 作为函数参数传到 fun()函数中,fun()函数的功能是将形参 n 中,各位上为偶数的数取出,并按原来从高位到低位的顺序组成一个新的数,并作为函数值返回。在第 3 个实验的例题中,我们学会了如何将一个整数的每一位拆开,再来判断每一位数是否是偶数,是则组成新的数。

(1)【1】处,对变量 i 赋初值,根据 i 的使用规则来看,i 应等于 1。

(2)【2】处,while 循环要求计算后的 s 大于 0,则应继续拆,所以填 s 或 s>0 或 s!=0 均可。

(3)【3】处,i 是 t 的位权,因此每循环一次,i 要乘以 10。

(4)【4】处,此处应填函数调用,将变量 n 作为函数的参数传递。

5.3.3 程序改错

以下程序的功能是求如下表达式:

$$S=1+\frac{1}{1+2}+\frac{1}{1+2+3}+\cdots+\frac{1}{1+2+3+\cdots+n}$$

请修改程序中 FOUND 注释下面语句中存在的错误,使程序能得出正确的结果。注意:不可以增加或删除程序行,也不可以更改程序的结构。

【程序改错】

```
#include <stdio.h>
main()
{
```

```
    int n;
    float fun(int n);
    printf("Please input a number:");
    / ********** FOUND ********** /
    print("%d",n) ;
    printf("%10.6f\n",fun(n));
}

/ ********** FOUND ********** /
fun(int n)
{
    int i,j,t;
    float s;
    s = 0;
/ ********** FOUND ********** /
    while(i = 1;i <= n;i++);
    {
        t = 0;
        for(j = 1;j <= i;j++)
            t = t + j;
        / ********** FOUND ********** /
        s = s + 1/t;
    }
    return s;
}
```

【参考答案】

(1) print("%d",n) ;改为 scanf("%d",&n)。

(2) fun(int n)改为 float fun(int n)。

(3) while(i=1;i<=n;i++);改为 for(i=1;i<=n;i++)或 for(i=1;i<n+1;i++)。

(4) s=s+1/t;改为 s= s + 1.0 /t;或 s+=1.0/t;或 s= s + 1 /(float)t;或 s+=1.0/(float)t。

【程序分析】

(1) 由题意可知,此处应该是输入 n 的值。

(2) 由题意和主函数中对 fun()函数的声明可知,fun()函数的返回值应该是 float 类型,如果定义函数的时候省略函数的类型,默认是 int 类型。

(3) 此处 while 循环的语法是错误的,应该是 for 循环,并且注意不能有分号,如果有分号则意味着循环的内容为空语句。

(4) fun()函数中定义的 t 为整型变量,C 语言规定,两个整型数相除的结果为整型,所以表达式 1/t 的结果是整型,当 t 为 1 时,1/t 的值为 1,当 t>1 时,1/t 的值为 0,显然不符合题目要求,所以至少需要将分子和分母中的 1 个数的类型变为浮点型。

5.4 思 考 题

(1) 验证哥德巴赫猜想的第二部分:每个大于等于 9 的奇数,都可表示为三个奇素数之和。将大于等于 9 的奇数分解为三个奇素数之和写成一个函数,从键盘输入任一大于等于

9 的奇数，调用该函数，在函数中输出这三个奇素数，例如输入 9，输出 9＝3＋3＋3。

（2）用递归方法求 1＋2＋3＋4＋…＋n。

（3）判断整数 x 是否是同构数。若是同构数，函数返回 1；否则返回 0。x 的值由主函数从键盘读入，要求不大于 100。说明：所谓“同构数”是指这样的数，这个数出现在它的平方数的右边。例如：输入整数 5，5 的平方数是 25，5 是 25 中右侧的数，所以 5 是同构数。

（4）从低位开始取出长整型变量 s 奇数位上的数，依次构成一个新数放在 t 中。例如：当 s 中的数为：7653421 时，t 中的数为：7541。在主函数中输入数 s，编写一个函数实现题目要求的功能，将构成的新数返回主函数输出。

（5）用递归法求 n 阶勒让德多项式的值，递归公式为：

$$P_n(x)=\begin{cases}1 & (n=0)\\ x & (n=1)\\ ((2n-1)\cdot x-P_{N-1}(x)-(n-1)\cdot P_{n-2}(x))/n & (n\geqslant 1)\end{cases}$$