函 数、递 归

自定义函数

6.0-1

```
#include<iostream>
using namespace std;
int jiafa(int a,int b)//创建一个  加法  子函数
{
	return a+b;
}
int main()
{
	int x,y;
	cin>>x>>y;
	cout<<jiafa(x,y);//主函数 main 调用子函数 jiafa
	return 0;
}
```

6.0-2

```
#include<iostream>
using namespace std;
int zuida(int a,int b)//创建一个  选择大数  子函数
{
	if(a>b) return a;
	else return b;
}
int main()
{
	int x,y;
	cin>>x>>y;
	cout<<zuida(x,y);//主函数 main 调用子函数 zuida
	return 0;
}
```

6.0-3

```
#include<iostream>
using namespace std;// 创建一个  乘法  子函数
void cheng(int a,int b);// 对子函数的声明，由于子函数在主函数 main 之后，所以需要事先声明
int main()
{
	int x,y;
	cin>>x>>y;
	cheng(x,y);
	return 0;
}
void cheng(int a,int b)//void 表示无返回值
{// 由于输出结果直接在子函数完成，无需返回结果值，因此用 void 表示无返回值
	cout<<a*b;
}
```

6.0-4

```
#include<iostream>
using namespace std;
int d;// 全局变量，可以在整个程序中使用
int jiafa(int a,int b)
{
	int c;//c 是局部变量，只能在子函数中使用
	c=a+b;
	return d-c; //d 是全局变量
}
int main()
{
	int x,y;//x,y 是局部变量，只能在主函数 main 中使用
	cin>>d>>x>>y;//d 是全局变量
	cout<<jiafa(x,y);
	return 0;
}
```

6.0-5

```
#include<iostream>
using namespace std;
int jiafa(int a,int b)
{
	int c;//c 是局部变量，只能在子函数中使用
	c=a+b;
	return c;
}
int main()
{
	int x,y;
	int c;//c 也是局部变量，与子函数的 c 名称相同，处于不同函数中，不产生冲突
	cin>>c>>x>>y;
	cout<<c-jiafa(x,y);
	return 0;
}
```

6.0-6-a

```
#include<iostream>// 数组函数，成功交换
using namespace std;
int i;// 全局变量，i 同时用于 mian 主函数和子函数，不产生冲突
void jiaohuan(int a[],int b[])
{
	int c;
	for(i=0;i<=2;i++)
	{	c=a[i];a[i]=b[i];b[i]=c;	}
}
int main()
{
	int e[]={1,2,3};
	int f[]={4,5,6};
	jiaohuan(e,f);//e,f 数组名
	for(i=0;i<=2;i++)		cout<<e[i]<<" ";
	return 0;
}
```

6.0-6-b

```
#include<iostream>// 数组函数错误举例，代码运行不成功，数据没有成功交换
using namespace std;
void jiaohuan(int a,int b)
{
	int c;
	c=a;a=b;b=c;
}
int main()
{
	int e[]={1,2,3};
	int f[]={4,5,6};
	for(int i=0;i<=2;i++)
		jiaohuan(e[i],f[i]);
	for(int i=0;i<=2;i++)		cout<<e[i]<<" ";
	return 0;
}
```

6.0-7

```
#include<iostream>// 二维数组函数
using namespace std;
int max(int a[][4])// 列数不可为空
{
	int max=a[0][0];
	for(int i=0;i<3;i++)
	{
		for(int j=0;j<4;j++)	{	if(a[i][j]>max)	max=a[i][j];	}
	}
	return max;
}
int main()
{
	int a[][4]={1,2,3,4,5,6,7,8,9,-1,-2,-3};
	cout<<max(a);
	return 0;
}
```

纯粹素数

题目描述：纯粹素数是这样定义的：一个素数，去掉最高位，剩下的数仍为素数，再去掉剩下的数的最高位，余下的数还是素数。这样下去一直到最后剩下的个位数也还是素数。求出所有小于 3000 的四位的纯粹素数。

按从小到大的顺序输出若干个纯粹素数，每行一个。

6.0-8

```
#include<bits/stdc++.h>                    // 纯粹素数   选自东方博宜
using namespace std;
bool sushu(int n)
{                                  // 定义布尔类型函数：sushu()，用来判断是否为素数
      bool r=true;
      for(int i=2;i*i<=n;i++)
      {
            if(n%i==0)
            {
                  r=false;
                  break;
            }
      }
      if(n<=1)
      {
            r=false;
      }
      return r;// 返回 r 的值
}
int main()
{
      int i;
      for(i=1000;i<3000;i++)
      {           //sushu(i) 意思是默认 sushu(i) bool 值为 true
            if(sushu(i)&&sushu(i%1000)&&sushu(i%100)&&sushu(i%10))
                  cout<<i<<endl;
      }
      return 0;
}
```

```
1013
1097
1103
1223
1283
1307
1367
1373
1523
1607
1613
1823
1907
1997
2003
2017
2053
2083
2113
2137
2347
2383
2467
2503
2617
2647
2683
2797
2953
```

递归

有 5 个人坐在一起，问第 5 个人多少岁，他说比第 4 个人大 2 岁；问第 4 个人多少岁，他说比第 3 个人大 2 岁；问第 3 个人多少岁，他说比第 2 个人大 2 岁； 问第 2 个人多少岁，他说比第 1 个人大 2 岁，最后问第 1 个人多少岁，他说是 10 岁。请问第 5 个人多少岁？

6.2-0

```
#include<iostream>// 函数的递归（函数调用自身）选自《编程竞赛宝典》
using namespace std;
int age(int n)
{
	if(n==1)
		return 10;
	else
		return	age(n-1)+2;// 调用自身
}
int main()
{
	cout<<age(5);
	return 0;
}
```

```cpp
#include<bits/stdc++.h>// 兔子序列  本程序超时
using namespace std;
int fun(int n)
{
	if(n==1||n==2) return 1;
	else if(n<=0) return 0;
		else return fun(n-1)+fun(n-2);
}
int main()
{
	int n;
	cin>>n;
	cout<<fun(n);
	return 0;
}
#include<bits/stdc++.h>// 兔子序列。本程序运行效率高
using namespace std;
long long tu[50];
long long fun(int  n)//fun 意思是函数
{
	if(n==1||n==2)	tu[n]=1;
	if(tu[n]!=0)	return tu[n];
	// 如果数值不为 0，表示数值已经计算出，不用再计算

	if(tu[n]==0)	return tu[n]=(fun(n-1)+fun(n-2));
	// 如果数值为 0，递归，计算出兔子数量
}
int main()
{
	long long n;
	cin>>n;

	cout<<fun(n);
	return 0;
}
```

```cpp
#include<bits/stdc++.h>//2 求 s 的值
using namespace std;
long long ss[50];

long long fun(int n)//fun 意思是函数
{
	if(n==1)		ss[n]=1;
	if(ss[n]!=0)	return ss[n];

	if(ss[n]==0)	return ss[n]=(fun(n-1)+n-1);
}
int main()
{
	long long s=0;
	int i=1;
	while(s<=5000)
	{
		s=s+fun(i);
		i++;
	}

	cout<<s;
	return 0;
}
```

```cpp
#include<bits/stdc++.h>//3. 前 n 项的和
using namespace std;
long long tu[50];
long long fun(int  n)
{
	if(n==1||n==2)	tu[n]=1;

	if(tu[n]!=0)	return tu[n];

	if(tu[n]==0)	return tu[n]=(fun(n-1)+fun(n-2));
}
int main()
{
	double s=0;// 总和
	int n;
	cin>>n;
	for(int i=1;i<=n;i++)
	{
		s=s+fun(i)*1.0/fun(i+1);
	}

	printf("%.3lf",s);
	return 0;
}
```