**数据Data：**描述客观事物的数值、字符以及能输入机器且能被处理的各种符号集合（图像、声音等一切可以输入计算机中的都可以称作数据）

**数据元素Data Element：**组成数据的基本单位，数据集合的个体，通常作为一个整体进行考虑和处理（数据元素可用一个结点表示）

一个数据元素可有一个或多个数据项Data Item组成，数据项是有独立含义的最小单位

**数据对象Data Object：**性质相同的数据元素的集合，数据的子集

**数据结构Data Structure：**相互之间存在一种或多种特定关系的元素的集合

数据结构是数据元素的集合，数据元素之间带有一定结构

数据结构是带有结构关系（线性结构、树型结构、图结构）的数据集合

数据结构=数据元素+数据元素之间的关系

**数据类型Data Type：**由一组性质相同的值的集合以及定义在这个值集合上的操作的总称

数据类型可分为原子类型和结构类型

原子类型：其值不可分解（整型、布尔型、字符型、指针等）

结构类型：其值由若干个成分组成（结构体）

**逻辑结构：**数据元素之间逻辑关系描述（线性结构、非线性结构）

Data\_Structure=(D,R)

D是数据元素的有限集，R是D上关系的有限集

集合结构：在同一个结构中的数据元素之间除了属于同一个集合之外他们没有任何关系

线性结构：结构中的数据元素它们之间存在着1对1的关系

树形结构：结构中的数据元素它们之间存在着一对多的层次关系

图状结构：结构中的数据元素存在着任意关系

**存储结构：**逻辑结构在计算机中的存储映像（顺序结构、非顺序结构）

顺序存储：用数据元素之间物理位置的紧邻表示逻辑关系的紧邻

（数据元素之间的逻辑关系和物理地址存储的位置一一对应）

链式存储：存储时逻辑关系是紧邻的，物理位置不一定是紧邻的

（为每一个数据元素增加一项信息，即它后面跟着的那个数据元素的地址信息）

**运算集合：**在计算机中进行运算操作的集合（增、删、改、查）

解决问题方法的效率，跟数据的组织方式有关

解决问题方法的效率，跟空间的利用效率有关

解决问题方法的效率，跟算法的巧妙程度有关

例：写程序实现一个函数PrintN，使得传入一个正整数为N的参数后，能顺序打印从一到N的全部正整数。

/\*循环\*/

#include<iostream>

using namespace std;

int main()

{

void PrintN(int n);

int n;

cout<<"请输入n：";

cin>>n;

PrintN(n);

system("pause");

return 0;

}

void PrintN(int n)

{

for(int i=0;i<n+1;i++)

{

cout<<i<<endl;

}

}

/\*递归\*/

#include<iostream>

using namespace std;

int main()

{

void PrintN(int n);

int n;

cout<<"请输入n：";

cin>>n;

PrintN(n);

system("pause");

return 0;

}

void PrintN(int n)

{

if(n)

{

PrintN(n-1);

cout<<n<<endl;

}

}

在做打印运算时，对递归100000，系统罢工---

递归：一个过程或函数在其定义或说明中有直接或间接调用自身的一种方法。

递归算法效率较低，占用栈空间较大，容易发生栈溢出。

递归分为2种，直接递归和间接递归。

直接递归，比如方法A内部调用方法A自身

间接递归，比如方法A内部调用方法B，方法B内部调用方法C，方法C内部调用方法A

clock();捕捉从程序开始运行到clock();被调用时所耗费的时间。

其时间单位是clock tick，即“时间打点”

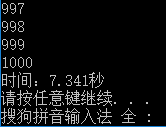
常数CLOCKS\_PER\_SEC表示一秒钟内CPU运行的时钟周期数

（CLK\_TCK：机器时钟每秒所走的时间打点数。CLK\_TCK 是等效的，但被视为已过时）

[CLOCKS\_PER\_SEC——百度百科](https://baike.baidu.com/item/CLOCKS_PER_SEC/2767233?fr=aladdin)

/\*循环\*/

#include<iostream>

#include<time.h>

using namespace std;

clock\_t start,stop;

//clock\_t是这个clock函数返回的变量类型

//定义两个变量，一个为stract，一个为stop

double t;

//t是一个double类型的变量

//用于记录最后算出来的被测函数运行的时间，以秒为时间单位

int main()

{

void PrintN(int n);

int n;

cout<<"请输入n：";

cin>>n;

//不在测试范围内的准备工作写在clock之前（输入n、预处理等）

start=clock(); /\*开始计时\*/

//从main函数开始运行，直到start被赋值，一共走过了多少个ticks，存在start中

PrintN(n); /\*被测函数\*/

//再次调用clock，存在stop中

stop=clock(); /\*停止计时\*/

//从main函数开始运行，直到stop被赋值，一共走过了多少个ticks，存在stop中

t=((double)(stop-start))/CLOCKS\_PER\_SEC;

//stop-start得到被测函数执行过程中，一共经历了多少个ticks

//除以常数CLOCKS\_PER\_SEC即机器时钟每秒所走的时间打点数，得到以秒为单位的值

//不在测试范围内的后继处理写在clock之后（输出time等）

cout<<"时间："<<t<<"秒"<<endl;

system("pause");

return 0;

}

void PrintN(int n)

{

for(int i=0;i<n+1;i++)

{

cout<<i<<endl;

}

}