



天天快乐编程 基础班讲义

(2020寒假)

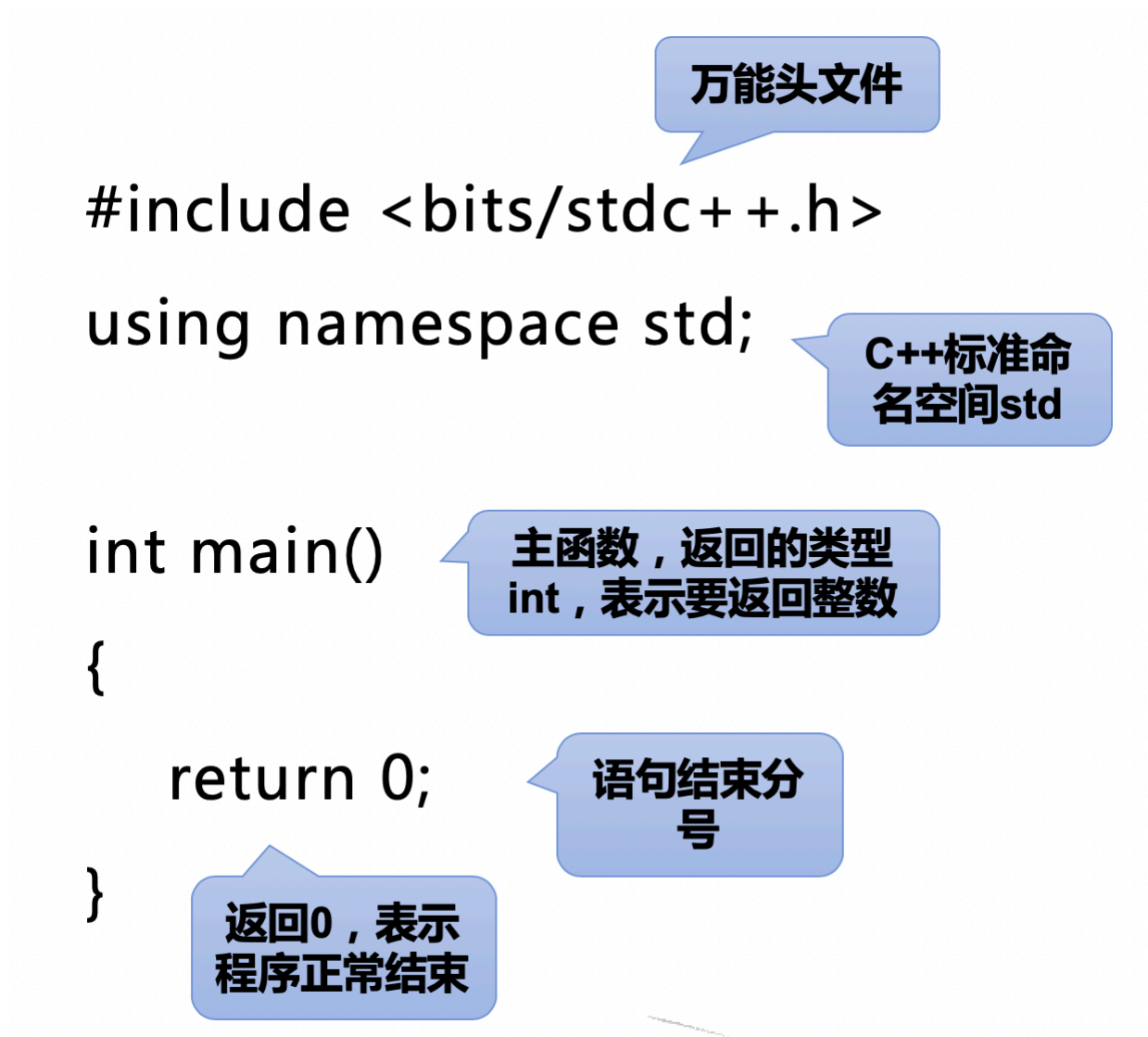
仅供天天快乐编程学员内部使用

一 前置技能	1
二 基本数据类型	3
三 基本输入输出	6
四 常见运算符	10
五 常用系统函数	12
六 选择结构	13
七 循环结构	15
八 数组	17
附录1:课后题目及解答	18
附录2:常见问题及解答	43



一 前置技能

1.C语言程序基本框架



程序从 main 函数开始执行。

main函数后面有一对小括号“int main()”和一对大括号（第5、7两行），里面有一条语句。

函数内容需要大括号隔开。

语句以分号“;”结束，分号绝对不能丢！

所有符号全部是英文半角字符！写代码时，注意关闭中文输入法。

输入程序时要注意：有的地方有空格、大小写、各种符号。

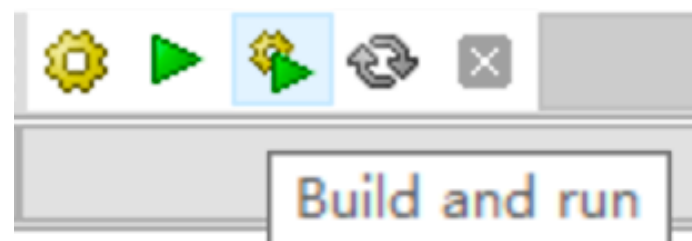
此为程序框架，所有程序都一样，你需要先把这六行代码写完再开始书写程序。

2. 软件及其快捷键

DEVCP



CodeBlocks



二 基本数据类型

1. 常见类型

在C++中，常见的数据类型有3种，分别是整型int，浮点型double（双精度），字符char。

1) int一般为4字节，表示从 -2^{31} (-2,147,483,648) 到 $2^{31} - 1$ (2,147,483,647) 的整型数据。

2) double为8字节，表示从 $-2^{1024} \sim +2^{1024}$ ，也即 $-1.79E+308 \sim +1.79E+308$ 的浮点型数据。

3) char为1字节，表示-128到+127的字符类型数据。

4) 在程序设计中也经常用到一系列字符的集合，也就是字符串（string）。

2. 定义变量和赋值

数据类型 变量名;

```
int a;
```

同一类型的变量可以用逗号隔开。变量的声明和赋初值还可以在一行完成，也就是说，在声明的时候就对这个变量进行初始化，如

```
int i=1;
```

```
int a=0,b=0;
```

```
char a=97; /*表示将十进制数字48所对应的字符0赋给变量a*/
```

```
char c=' '; /*表示将字符空格赋给变量c*/
```

```
double PI=3.1415926;
```

在定义变量时，变量名可以是字母、数字和下划线的组合。但是也不是随便的组合，要注意以下几个命名规则：

1) 变量名的开头必须是字母或下划线，不能是数字。实际编程中最常用的是以字母开头，而以下划线开头的变量名是系统专用的。

所以为了避免与系统定义的名字产生冲突，在编程的时候，除非要求这么定义，否则永远都不要使用下划线作为一个变量名的开头。

- 2) 变量名中的字母是区分大小写的。比如 a 和 A 是不同的变量名，num 和 Num 也是不同的变量名。
- 3) 变量名绝对不可以是C语言关键字，这一点一定要记住！
- 4) 变量名中不能有空格。这个可以这样理解：因为上面我们说过，变量名是字母、数字、下划线的组合，没有空格这一项。

3.C语言关键字

关键字是无法做为变量名字的，ANSI C 一共4类关键字，32个关键字。

char	short	int	long	float	double
signed	unsigned	enum	union	struct	void
for	do	while	break	continue	if
else	goto	swicth	case	default	return
auto	extern	register	static	const	volatile
sizeof	typedef				

数据类型关键字（12个），控制语句关键字（12个），存储类型关键字（4个），其它关键字（4个）



4.ASCII表

字符类型char是字符和数字值一一对应的，使用的是美国信息交换标准代码，即ASCII。

常用的有空格为32，0为48，A为65，a为97，A与a相差32。

ASCII值表							
ASCII值	控制字符	ASCII值	控制字符	ASCII值	控制字符	ASCII值	控制字符
0	NUT	32	(space)	64	@	96	`
1	SOH	33	!	65	A	97	a
2	STX	34	"	66	B	98	b
3	ETX	35	#	67	C	99	c
4	EOT	36	\$	68	D	100	d
5	ENQ	37	%	69	E	101	e
6	ACK	38	&	70	F	102	f
7	BEL	39	,	71	G	103	g
8	BS	40	(72	H	104	h
9	HT	41)	73	I	105	i
10	LF	42	*	74	J	106	j
11	VT	43	+	75	K	107	k
12	FF	44	,	76	L	108	l
13	CR	45	-	77	M	109	m
14	SO	46	.	78	N	110	n
15	SI	47	/	79	O	111	o
16	DLE	48	0	80	P	112	p
17	DCI	49	1	81	Q	113	q
18	DC2	50	2	82	R	114	r
19	DC3	51	3	83	S	115	s
20	DC4	52	4	84	T	116	t
21	NAK	53	5	85	U	117	u
22	SYN	54	6	86	V	118	v
23	TB	55	7	87	W	119	w
24	CAN	56	8	88	X	120	x
25	EM	57	9	89	Y	121	y
26	SUB	58	:	90	Z	122	z
27	ESC	59	;	91	[123	{
28	FS	60	<	92	/	124	
29	GS	61	=	93]	125	}
30	RS	62	>	94	^	126	`
31	US	63	?	95	_	127	DEL

三 基本输入输出

1.cout输出语句

cout 可用于在计算机屏幕上显示信息，程序可以显示的最简单的屏幕输出类型是终端（黑框框）输出，它只是纯文本。endl为换行(end line)。

```
cout<<i<<c<<PI;
```

```
cout<<“天天快乐编程”<<endl;
```

保留两位小数的写法

```
cout<<setiosflags(ios::fixed)<<setprecision(2)<<PI; /*不推荐*/
printf("%.2f\n",PI); /*简洁写法*/
```

2.转义字符

转义字符是一种特殊的字符常量。转义字符以反斜线“\”开头，后跟一个或几个字符。转义字符具有特定的含义，不同于字符原有的意义，故称“转义”字符。例如，在前面各例题 printf 函数的格式串中用到的“\n”就是一个转义字符，其意义是“回车换行”。转义字符 主要用来表示那些用一般字符不便于表示的控制代码。

常用的转义字符及其含义		
转义字符	ASCII 码	转义字符的含义
\n	10	回车换行
\t	9	制表符
\\	92	反斜线符\'
\'	39	单引号符
\"	34	双引号符

所以以上代码可以偷懒为

```
cout<<“天天快乐编程\n”;
```

3.cin输入语句

C++可以通过cin直接读入各种类型的变量，输入运算符>>在读入下一个输入项前会忽略前一项后面的空格。


```
int i=1;
char c;
double PI;
cin>>i>>c>>PI;
```

4.scanf和printf格式输入输出语句

scanf 和 printf 这两个函数分别称为格式输入函数和格式输出函数。其意义是按指定的

格式输入输出值。因此，这两个函数在括号中的参数表都由以下两部分组成：

“格式控制串”，参数表

格式控制串是一个字符串，必须用双引号括起来，它表示了输入输出量的数据类型。

在 printf 函数中还可以在格式控制串内出现非格式控制符，这时在显示屏幕上将原文照印。参数表中给出了输入或输出的量。当有多个量时，用逗号间隔。

```
scanf("%d %c %lf",&i,&c,&PI);
printf("%d %c %f",i,c,PI);
```

保留两位小数的新写法

```
printf("%.2f\n", PI);
```

d 以十进制形式输出带符号整数(正数不输出符号)

o 以八进制形式输出无符号整数(不输出前缀O)

x 以十六进制形式输出无符号整数(不输出前缀OX)

u 以十进制形式输出无符号整数

f 以小数形式输出单、双精度实数

e 以指数形式输出单、双精度实数

g 以%f%e中较短的输出宽度输出单、双精度实数

c 输出单个字符

s 输出字符串

标志字符为-、+、#、空格四种，其意义：

- 结果左对齐，右边填充空格

+ 输出符号(正号或负号)空格输出值为正时冠以空格，为负时冠以负号

对c, s, d, u类无影响；对o类，在输出时加前缀。对x类，在输出时加前缀0x；对e,g,f类当结果有小数时才给出小数点

输出的格式控制

%nd n为指定的输出字段的宽度。如果数据的位数小于n，则左端补以空格，若大于n，则按实际位数输出。

%0nd n为指定的输出字段的宽度。如果数据的位数小于n，则左端补以0，若大于n，则按实际位数输出。

%m.nf：输出共占m列，其中有n位小数，如数值宽度小于m左端补空格。

5.使用getline读取一行

```
string line;
```

```
getline (cin, line) ;//按回车键结束输入
```

//输入一串字符（不管多少个回车键），只要是在'#'号之前的字符都会读取并保存

```
getline (cin,line,'#');
```

6.强制转换

当然内部还存在自动类型转换，总是把表达范围小的往大的转。所以要将整型变量a转为浮点数还可以使用*1.0的方式，即可完成类型的转换，因为double的范围比int大。

(新类型名称) 表达式 就可以完成强制的类型转换。

```
char c;  
cin>>c;  
cout<<(int)c<<endl;
```

或者

```
char c;
```

```
cin>>c;  
int a=c;  
cout<<a<<endl;
```



四 常见运算符

1.算术运算符

用于各类数值运算。包括加(+)、减(-)、乘(*)、除(/)、求余(或称模运算, %)、自增(++)、自减(--)共七种。

2.关系运算符

用于比较运算。包括大于(>)、小于(<)、等于(==)、大于等于(>=)、小于等于(<=)和不等不等于(!=)六种。

3.逻辑运算符

用于逻辑运算。包括与(&&)、或(||)、非(!)三种。

4.位操作运算符

参与运算的量, 按二进制位进行运算。包括位与(&)、位或(|)、位非(~)、位异或(^)、左移(<<)、右移(>>)六种。

5.赋值运算符

用于赋值运算, 分为简单赋值(=)、复合算术赋值(+=, -=, *=, /=, %=)和复合位运算赋值(&=, |=, ^=, >>=, <<=)三类共十一种。

例如

`+=: a += b`相当于`a = a+b`

`*=: a *= b`相当于`a = a*b`

`%=: a %= b`相当于`a = a%b`

`-=: a -= b`相当于`a = a-b`

`/=: a /= b`相当于`a = a/b`

注: 交换变量, 需要借助第三个变量

`t=a; a=b; b=t;`

6.条件运算符

这是一个三目运算符，用于条件求值(?:)。

7.逗号运算符

用于把若干表达式组合成一个表达式(,)。

8.指针运算符

用于取内容(*)和取地址(&)二种运算。

9.求字节数运算符

用于计算数据类型所占的字节数(sizeof)。

10.特殊运算符

有括号(), 下标[], 成员(→, .)等几种。

注意：

1.判断语句和赋值语句是两个常用语句，判断的是关系，需要 "==" 判断字符，赋值语句需要使用 "=" 进行赋值。

2.int类型的除法为商，不是小数。如 $7/3=2$

五 常用系统函数

1.平方根: `sqrt(x)`, 表示 \sqrt{x}

2.`pow(x, y)`, 表示 x^y

3.小数绝对值: `fabs(x)`

4.向下取整: `floor(x)`

5.向上取整: `ceil(x)`

6.`sin(x)`、`cos(x)`、`acos(x)`...



六 选择结构

1.基本介绍

用 if 语句可以构成分支结构。它根据给定的条件进行判断，以决定执行某个分支程序段。

//多分支

```
if(条件判断1)
```

```
{
```

```
    执行语句1
```

```
}
```

```
else if(条件判断2)
```

```
{
```

```
    //条件1不成立时
```

```
    执行语句2
```

```
}
```

```
.....//更多的else if分支
```

```
else
```

```
{
```

```
    //上面两个都不成立时
```

```
    执行语句n
```

```
}
```

四年一闰，百年不闰，四百年又闰

```
if (y % 4 == 0)
```

```
{
```

```
    if (y % 100 != 0)
```

```
    {
```

```
        cout << "Yes" << endl;
```

```
    }
```

```
else
```

```
{
```

```

        if (y % 400 == 0)
        {
            cout << "Yes" << endl;
        }
        else
        {
            cout << "No" << endl;
        }
    }
}
else
{
    cout << "No" << endl;
}

```

2.逻辑运算符详细介绍

运算符	含义	解释
&&	与	将两个表达式连接成一个。两个表达式必须都为 true，整个表达式才为 true
	或	将两个表达式连接成一个。两个表达式只要有一个为 true，整个表达式即为 true
!	非	使一个表达式从 true 变成了 false，或者从 false 变成了 true

&&(与) 同1得1, || (或) 有1的1, ! (非) 置反

判断闰年的简单写法

```

if (y % 4 == 0 && y % 100 != 0 || y % 400 == 0)
{
    cout << "Yes" << endl;
}
else
{
    cout << "No" << endl;
}

```



```
}
```

七 循环结构

1.while语句

```
while(条件判断)
```

```
{
```

```
    循环体
```

```
}
```

条件判断结果为true时执行循环体，否则结束循环，继续运行后面的语句。

2.for语句

如果循环次数已知，用for循环更合理

```
for(循环变量初始化;循环条件;循环变量变化)
```

```
{
```

```
    循环体;
```

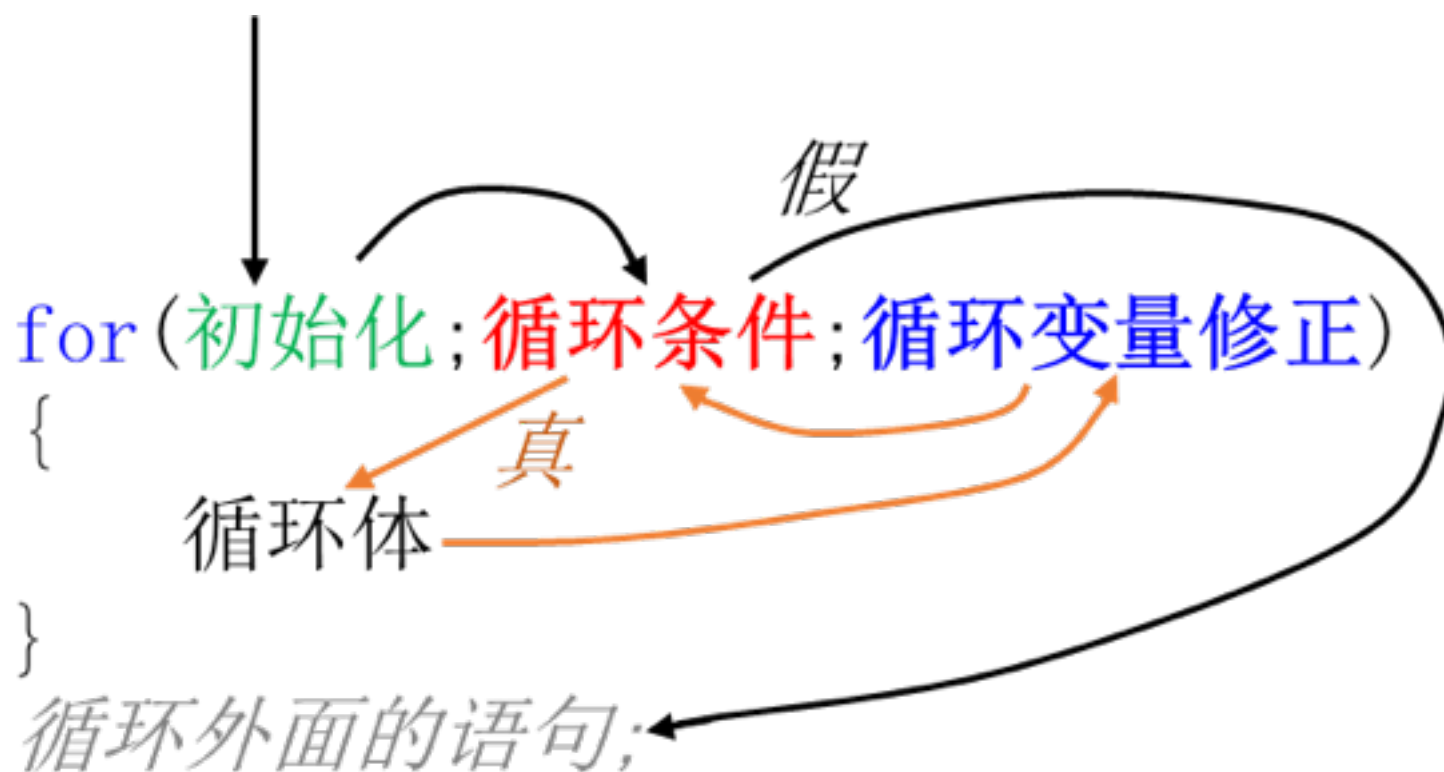
```
}
```

循环执行过程如下

- 1) 先执行循环变量初始化。
- 2) 判断循环条件，若其值为真（非 0），则执行 for 语句中指定的内嵌语句，然后执行下面
- 第 3) 步；若其值为假（0），则结束循环，转到第 5) 步。
- 3) 执行循环变量变化。
- 4) 转回上面第 2) 步继续执行。
- 5) 循环结束，执行 for 语句下面的一个语句。

注意：1for后面必须加括号，括号内必须有2个分号“;”

执行过程图示



3.do-while结构

```
do
{
    循环体
}while(条件判断);
```

do...while先执行循环体，再执行判断条件（先斩后奏），不要忘记之后分号“;”

4.continue和break

循环可以中途被打断，即使用break;

或者跳过本次循环的部分语句，即使用continue;

但只能跳过或者中断当前一重循环。

goto可以跳出多重循环。



八 数组

1.数组介绍

在程序设计中，为了处理方便，把具有相同类型的若干变量按有序的形式组织起来。一个数组可以分解为多个数组元素，这些数组元素可以是基本数据类型或是构造类型。

2.一维数组的定义方式

数据类型 数组名 [常量表达式];

方括号中的常量表达式表示数据元素的个数，也称为数组的长度。例如：

例子：

```
int a[10]; /* 说明整型数组a，有10个元素 */
```

```
Double b[10], c[20]; /* 说明浮点型数组b，有10个元素，浮点型型数组c，有20个元素 */
```

```
char ch[20]; /* 说明字符数组ch，有20个元素 */
```

对于数组类型说明应注意以下几点：

- 1) 数组的类型实际是指数组元素的取值类型。对于同一个数组，其所有元素的数据类型都是相同的。
- 2) 数组名的书写规则应符合标识符的书写规定。
- 3) 数组名不能与其它变量名相同。
- 4) 方括号中常量表达式表示数组元素的个数，如a[5]表示数组a有5个元素。但是其下标从0开始计算。因此5个元素分别为a[0], a[1], a[2], a[3], a[4]。
- 5) 不能在方括号中用变量来表示元素的个数，但是可以是符号常数或常量表达式。
- 6) 允许在同一个类型说明中，说明多个数组和多个变量。



附录1:课后题目及解答

【任务 1——TZOJ1452:Hello World】

题目描述

在一行中输出文字“Hello, World!”, 不含双引号。

输入描述

无输入

输出描述

Hello, World!

样例输入	样例输出
	Hello, World!

题目分析

直接cout输出。特别 要注意输出与题目要完全一致。如果出现 Wrong Answer, 请检查字母以及大小写是否正确, 逗号和感叹号是否为英文字母。如果出现 Presentation Error, 请检查逗号后面是否有 1 个空格。这种题目复制粘贴它不香吗? ctrl+c 复制, ctrl+v粘贴, ctrl+x剪切。

```
#include<bits/stdc++.h>
using namespace std;
int main()
{
    cout<<"Hello, World!\n";
    return 0;
}
```

【任务 2——TZOJ1001:整数求和】

题目描述

求两个整数之和。

输入描述

输入数据只包括两个整数 A 和 B。

输出描述

两个整数的和。

样例输入	样例输出
1 2	3

题目分析

了解 C 语言的数据类型、变量定义以及输入和输出等知识点。

```
#include<bits/stdc++.h>
using namespace std;
int main()
{
    int a,b;
    cin>>a>>b;
    cout<<a+b<<endl;
    return 0;
}
```

【任务 3——TZOJ1165：三个整数】

题目描述

给出三个整数，请你设计一个程序，求出这三个数的和、乘积和平均数。

输入描述

输入只有三个正整数 a、b、c。

输出描述

输出一行，包括三个的和、乘积、平均数。数据之间用一个空格隔开，其中平均数保留小数后面两位。

样例输入	样例输出
1 2 3	6 6 2.00

题目分析

需要事先了解常见的运算符、表达式和类型转换等相关知识点。此处三个数、和、乘积均为整型，仍然使用 int，但平均值为小数，需要使用浮点型的 double。保留小数输出需要使用 printf。需要注意的是两个整型数值相除后只取整数部分(商)，小数部分被忽略。若要获得小数结果，除数和被除数中必须有一个为浮点型，可以使用转型或改用浮点型常量。

```
#include<bits/stdc++.h>
using namespace std;
int main()
{
    int a,b,c;
    cin>>a>>b>>c;
    cout<<a+b+c<<" "<<a*b*c;
    printf(" %.2f\n", (a+b+c)/3.0);
    return 0;
}
```

【任务 4——TZOJ1491：求三角形面积】

题目描述

已知三角形的边长 a 、 b 和 c ，求其面积。

输入描述

输入三边 a 、 b 、 c 。

输出描述

输出面积，保留 3 位小数。

样例输入	样例输出
1 2 2.5	0.950

题目分析

计算三角形面积需要使用海伦－秦九韶公式。 $S=\sqrt{p(p-a)(p-b)(p-c)}$ ， p 为周长的一半。需要使用sqrt函数，而且输入也是实数哦。

```
#include<bits/stdc++.h>
using namespace std;
int main()
{
    double a,b,c;
    cin>>a>>b>>c;
    double p=(a+b+c)/2;
    printf("%.3f\n",sqrt(p*(p-a)*(p-b)*(p-c)));
    return 0;
}
```

【任务 5——TZOJ1499：鸡兔同笼】

题目描述

“鸡兔同笼”是我国古代著名趣题之一。大约在 1500 年前，《孙子算经》中就记载了这个有趣的问题。书中是这样叙述的：“今有雉兔同笼，上有三十五头，下有九十四足，问雉兔各几何？”这四句话的意思是：有若干只鸡兔同在一个笼子里，从上面数，有 35 个头；从下面数，有 94 只脚。求笼中各有几只鸡和兔？

现有 n 个头和 m 个脚，要你写一个程序计算到底有多少只鸡和兔。

输入描述

输入数据有一行，共 2 个整数 n 和 m ，以空格分隔。

输出描述

每组数据的输出都只有一行，分别是鸡和兔的数量。

样例输入	样例输出
2 6	1 1

题目分析

计算机可不会解方程哦，需要你自己列出来方程求解。 $2x+4y=m$ ， $x+y=n$ ， $2(n-y)+4y=m$ ， $n+2y=m$ ， $y=(m-2n)/2$ 。然后丢给计算机答案就好啦，具体的数值计算机会帮你算好的。

```
#include<bits/stdc++.h>
using namespace std;
int main()
{
    int n,m;
    cin>>n>>m;
    cout<<n-(m-2*n)/2<<" "<<(m-2*n)/2<<endl;
    return 0;
}
```

【任务 6——1492: C语言实验题——大小写转换】

题目描述

从键盘输入一个大写字母，要求改用小写字母输出。

输入描述

输入一个大写字母。

输出描述

输出对应的小写字母。

样例输入	样例输出
A	a

题目分析

之前我们讲过大小写之间相差32，但是你加上32输出就错了，因为会自动转换为范围更大的int，所以你需要强转回来。

```
#include<bits/stdc++.h>
using namespace std;
int main()
{
    char c;
    cin>>c;
    cout<<(char)(c+32)<<"\n";
    return 0;
}
```

【任务 7——TZOJ1472: C语言实验题——逆置正整数】

题目描述

输入一个三位正整数，将它反向输出。

输入描述

3 位正整数。

输出描述

输出逆置后的正整数（去除前导0）。

样例输入	样例输出
123	321

题目分析

这个题目很容易就掉进坑里，因为要去除前导0。我们其实可以将其组装起来，我们所用的十进数都不带前导0的，那就分别得到个十百位就好了。

$abc = a * 100 + b * 10 + c * 1$ ，可以理解下其中的进制关系

```
#include<bits/stdc++.h>
using namespace std;
int main()
```



```

{
    int x;
    cin>>x;
    int g=x%10,s=x/10%10,b=x/100;
    cout<<g*100+s*10+b<<"\n";
    return 0;
}

```

【任务 8——TZOJ1467: C语言实验题——两个数最大】

题目描述

求两个整数中的最大值。

输入描述

两个整数

输出描述

输出最大的值，格式为：

max=最大值

样例输入	样例输出
1 2	max=2

题目分析

建议使用我这个代码，用a作为最大值。格式不要输出错哦，一定要仔细检查。

```

#include <bits/stdc++.h>
using namespace std;
int main()
{
    int a,b,m;
    cin>>a>>b;
    m=a;
    if(b>m)
    {
        m=b;
    }
}

```

```

    cout<<"max="<<m<<endl;
    return 0;
}

```

【任务 9——TZOJ4590: 最大值-最小值】

题目描述

计算三个数中的最大值与最小值之差。

输入描述

输入包括三个整数。

输出描述

输出最大值与最小值之差。

样例输入	样例输出
1 2 3	2

题目分析

之前是不是做对了，但是这个不对呢。你需要分别执行求最大值和最小值的代码将其保存下来，再相减。这里老师还是使用了很多大括号，虽然很麻烦，但是很多同学在这之后的练习会忘掉大括号。建议每个if都要加上，之后自己熟练了就可以去掉了。

```

#include <bits/stdc++.h>
using namespace std;
int main()
{
    int a,b,c,m,n;
    cin>>a>>b>>c;
    m=a;
    if(b>m)
    {
        m=b;
    }
    if(c>m)
    {

```

```

        m=c;
    }
    n=a;
    if(b<n)
    {
        n=b;
    }
    if(c<n)
    {
        n=c;
    }
    cout<<m-n<<endl;
}

```

【任务 10——TZOJ4979: IP地址】

题目描述

IP地址是指互联网协议地址（英语：Internet Protocol Address，又译为网际协议地址），是IP Address的缩写。IP地址是IP协议提供的一种统一的地址格式，它为互联网上的每一个网络和每一台主机分配一个逻辑地址，以此来屏蔽物理地址的差异。

IP地址通常用“点分十进制”表示成（a.b.c.d）的形式，其中，a, b, c, d都是0~255(含0和255)之间的十进制整数。

请判断输入的IP地址是否合法。

输入描述

输入为一个“点分十进制”表示的IP地址，形式为（a.b.c.d）。

输出描述

判断该地址是否合法，如果合法输出YES，否则输出NO

样例输入	样例输出
10.11.12.13	YES

题目分析

这个属于if的基础题目，但cin读取整形遇到'.'之后无法读入，我们可以用一个char给他装进去，但是这个字符对我们是没用的，之后就可以正常读入了。

```
#include <bits/stdc++.h>
using namespace std;
int main()
{
    int a,b,c,d;
    char e;
    cin>>a>>e>>b>>e>>c>>e>>d;
    if(a>=0&&a<256&&b>=0&&b<256&&c>=0&&c<256&&d>=0&&d<256)
    {
        cout<<"YES\n";
    }
    else
    {
        cout<<"NO\n";
    }
    return 0;
}
```

【任务 11——TZOJ1454: C语言实验题——三个数排序】

题目描述

输入三个整数 x, y, z ，请把这三个数由小到大输出。

输入描述

输入数据包含3个整数 x, y, z ，分别用逗号隔开。

输出描述

输出由小到大排序后的结果，用空格隔开。

样例输入	样例输出
2,1,3	1 2 3

题目分析



先展示一种简单的写法，就是去第一个为最小值，接下来为次大，第三个为最大。第一个最小，那就是第一个和之后的比较，比较完不确定之后的两个大小，再比较。

```
#include <bits/stdc++.h>
using namespace std;
int main()
{
    int a,b,c,t;
    char d;
    cin>>a>>d>>b>>d>>c;
    if(a>b)
    {
        t=a;a=b;b=t;
    }
    if(a>c)
    {
        t=a;a=c;c=t;
    }
    if(b>c)
    {
        t=b;b=c;c=t;
    }
    cout<<a<<" "<<b<<" "<<c<<"\n";
    return 0;
}
```

此题还可采用“冒泡法”这一排序思想解决，本题只有 3 个数据，可以认为是排序的简单版。给定三个整数 a、b 和 c，“冒泡法”排序过程可以表示为（以 a=5,b=4,c=3 为例）：

1.第一趟比较有 3 个数，相邻两两比较需要比较 2 次，第一次是 a(5)和 b(4)的比较，将较大值交换到后面，即变成 a(4)和 b(5)，第二次是 b(5)和 c(3)的比较，同理交换后变成 b(3) 和 c(5)。此时大值 5 已经交换到了后（c 的位置），下一趟将不再参与排序。a、b、c 的结果分别为：4 3 5

2.第二趟剩下前面 2 个数，即 a(4)和 b(3)，再作一次比较将较大值交换到后面即可，即变成 a(3)和 b(4)。至此，所有排序即已经完成。

总结后，我们发现其实 n 个数的比较我们需要 n-1 趟（第 n-1 趟排序后只剩下一个数了，

不必再排序），在第 1 趟有 n-1 次相邻比较，第 2 趟需要 n-2 次，...，第 i 趟（i 从 1~n-1）则需要相邻比较的次数为 (n-i) 次，总共需要的次数为 $(n-1)+(n-2)+\dots+1=(n-1)n/2$ 次。每趟排序总有一个数排序位置确定（形象的认为有一个气泡冒出，冒泡法由此得名）。另外需要注意输入的格式是由“逗号”隔开的。

另外一种不是很好的方法是直接穷举 6 种情况，但初学者比较容易犯的错误是三个表达式之间的比较，应该转换为两两比较。如 $a < b < c$ ，实际上是先判断 $a < b$ 的结果，是一个逻辑值 0 或者 1（表示结果错或对），再将结果（0 或 1）与 c 比较，这显然是错误的。应该使用逻辑运算符“&&”（并且）来完成，即 $a < b \ \&\& \ b < c$ 。穷举的方法局限性较大，当有 N 个数需要排序时，会有 N!种情况需要讨论。

```
#include <bits/stdc++.h>
using namespace std;
int main()
{
    int a,b,c,t;
    char d;
    cin>>a>>d>>b>>d>>c;
    if(a>b)
    {
        t=a;a=b;b=t;
    }
    if(b>c)
    {
        t=b;b=c;c=t;
    }
    if(a>b)
    {
        t=b;b=a;a=t;
    }
}
```

```

    cout<<a<<" "<<b<<" "<<c<<"\n";
    return 0;
}

```

【任务 12——TZOJ5892: 苹果和虫子】

题目描述

你买了一箱 n 个苹果，很不幸的是买完时箱子里混进了一条虫子。虫子每 x 小时能吃掉一个苹果，假设虫子在吃完一个苹果之前不会吃另一个，那么经过 y 小时你还有多少个完整的苹果？

输入描述

输入仅一行，包括 n ， x 和 y （均为整数）。

输出描述

输出也仅一行，剩下的苹果个数。

样例输入	样例输出
10 4 9	7

题目分析

这个还是比较容易分析出答案的，就是用总数-吃掉的，吃掉是`ceil`函数，也就是只要有余数就会多吃一个苹果。但是这里我介绍一个好写法，就是其实我把它加上被除数-1，它余数为0的就不会被加1，其他均被+1，就是我们想要的。

不过还有个坑，就是苹果数肯定不可能是负数。

```

#include <bits/stdc++.h>
using namespace std;
int main()
{
    int n,x,y;
    cin>>n>>x>>y;
    int ans=n-(y+x-1)/x;
    if(ans<0)
    {
        ans=0;
    }
}

```



```

    cout<<ans<<endl;
    return 0;
}

```

【任务 13——TZOJ5894: 被17整除】

题目描述

将正整数 m 和 n 之间(包括 m 和 n)能被17整除的数累加, 其中 $0 < m < n < 1000$ 。

输入描述

一行, 包含两个整数 m 和 n , 其间, 以一个空格间隔。

输出描述

输出一行, 包行一个整数, 表示累加的结果。

样例输入	样例输出
50 85	204

题目分析

这个题目就是 m 到 n 的循环, 如果取余为0就加上, 但是我想讲解另一种写法。

```

#include <bits/stdc++.h>
using namespace std;
int main()
{
    int m,n;
    cin>>m>>n;
    int sum=0;
    for(int i=m;i<=n;i++)
    {
        if(i%17==0)
        {
            sum+=i;
        }
    }
    cout<<sum<<"\n";
    return 0;
}

```

这种写法是什么呢，就是我可以ceil拿到第一项，最后一项就是商，然后等差数列求和（首项+末项）*项数/2，项数怎么算你可以仔细想想哦

```
#include<bits/stdc++.h>
using namespace std;
int main()
{
    int m,n;
    cin>>m>>n;
    int x=(m+16)/17;
    int y=n/17;
    cout<<(x+y)*(y-x+1)/2*17;
    return 0;
}
```

【任务 14——TZOJ5886: 计算2的幂】

题目描述

给定非负整数 n ，求 2^n 的值，即2的 n 次方。

输入描述

一个整数 n 。 $0 \leq n < 31$ 。

输出描述

一个整数，即2的 n 次方。

样例输入	样例输出
3	8

题目分析

设初始值为1，然后多一次就多乘一个2。十进制就是乘10，如果熟悉了这个关系可以去尝试下1455：C语言实验题—数字串求和。

```
#include<bits/stdc++.h>
using namespace std;
int main()
{
    int n;
    cin>>n;
```

```

int ans=1;
for(int i=0;i<n;i++)
{
    ans=ans*2;
}
cout<<ans<<"\n";
return 0;
}

```

前面有没有看到位移运算，这个可以用右移很好实现哦。右移运算是将一个二进制位的操作数按指定移动的位数向右移动，移出位被丢弃，左边移出的空位或者一律补0，或者补符号位，这由不同的机器而定。 $\gg 1$ 其实就是 $\times 2$ ， $\gg n$ 其实就是 $\times (2^n)$ 为了保证运算符的优先级以及能编译，请记得加括号。

```

#include<bits/stdc++.h>
using namespace std;
int main()
{
    int n;
    cin>>n;
    cout<<(1<<n)<<"\n";
    return 0;
}

```

【任务 16——TZOJ4589: 素数判断】

题目描述

给定一个大于1的正整数，判断是否为素数。

输入描述

输入数据有多组，每组占一行，每行一个正整数 n ($1 < n < 10000$)。

输入以EOF结束。

输出描述

若为素数，输出Yes，否则输出No。



样例输入	样例输出
3	Yes
4	No

题目分析

素数或者说质数，是因子只有1和它本身的数。一个数不是素数，那么就有其他因子，我们可以去枚举这个因子，并使用标记变量去记录。这个题目是读取到EOF，也就是程序末尾，cin是有返回值的，把它作为判断条件即可。

```
#include<bits/stdc++.h>
using namespace std;
int main()
{
    int n;
    while(cin>>n)
    {
        int flag=1;
        for(int i=2;i<n;i++)
        {
            if(n%i==0)
            {
                flag=0;
                break;
            }
        }
        if(flag)
            cout<<"Yes\n";
        else cout<<"No\n";
    }
    return 0;
}
```

【任务 17——TZOJ1301: 统计同成绩学生人数】

题目描述

读入N名学生的成绩，将获得某一给定分数的学生人数输出。

输入描述



测试输入包含若干测试用例，每个测试用例的格式为

第1行：N

第2行：N名学生的成绩，相邻两数字用一个空格间隔。

第3行：给定分数

当读到N=0时输入结束。其中N不超过1000，成绩分数为（包含）0到100之间的一个整数。

输出描述

对每个测试用例，将获得给定分数的学生人数输出。

样例输入	样例输出
3	1
80 60 90	0
60	2
2	
85 66	
0	
5	
60 75 90 55 75	
75	
0	

题目分析

这个题目就非用数组不可了，因为你需要保留学生的成绩。这个是n等于0结束，你还可以用前一任务的写法，然后在循环体里判断n==0然后break。不过更建议你用我这种写法哦，逗号表达式判断的是逗号的，!=0在判断语句可以省略。

```
#include<bits/stdc++.h>
using namespace std;
int main()
{
```

```
    int n;
```



```

int a[1000];
while(cin>>n,n)
{
    for(int i=0;i<n;i++)
    {
        cin>>a[i];
    }
    int x,sum=0;
    cin>>x;
    for(int i=0;i<n;i++)
    {
        if(a[i]==x)
            sum++;
    }
    cout<<sum<<"\n";
}
return 0;
}

```

Bob还有另一种写法呢，就是其实我可以用数组去直接记录每个分数的人数，把分数作为下标。数组的初始化可以用={0}，这个代码不懂也没关系，加油。

```

#include<bits/stdc++.h>
using namespace std;
int main()
{
    int n;
    while(cin>>n,n)
    {
        int a[101]={0};
        for(int i=0,x;i<n;i++)
        {
            cin>>x;
            a[x]++;
        }
        int x;
        cin>>x;
    }
}

```



```

        cout<<a[x]<<"\n";
    }
    return 0;
}

```

【任务 18——TZOJ2952: 元素的删除】

题目描述

陪同Mr.true&&Eddy一同去面试的还有DavidDeng。

DavidDeng信心满满的去微软面试，一群面试者排起了长长的队伍，每个人有张面试编号。

这时，一位大美女KZ从他身边走过，DavidDeng对其一见钟情，DavidDeng拿起他的面试编号N就离开了面试队伍。

输入描述

第一行为一个整数T，表示有T组数据。

每组数据有三行：

第一行是一个整数M，表示原队伍有M个人；

第2行有M个以空格隔开的整数，即原队伍中每个人的面试编号；

第三行为DavidDeng的面试编号N。

保证没有相同号码，保证M不大于10000

输出描述

DavidDeng离开后队伍的面试编号序列。

样例输入	样例输出
1	1 8 2
4	
1 3 8 2	
3	



题目分析

这个题目是输入的另一组，T组，你可以用一个循环。但是while(T--)真的很方便呢，就是执行判断语句之后减一，然后判断是否为0。这个输出还是比较麻烦的，是输出之后才有空格，需要使用标记变量的。

```
#include<bits/stdc++.h>
using namespace std;
int main()
{
    int T;
    cin>>T;
    while(T--)
    {
        int n;
        cin>>n;
        int a[10005];
        for(int i=0,x;i<n;i++)
        {
            cin>>a[i];
        }
        int x,flag=0;
        cin>>x;
        for(int i=0;i<n;i++)
        {
            if(x!=a[i])
            {
                if(flag)cout<<" ";
                cout<<a[i];
                flag=1;
            }
        }
        cout<<"\n";
    }
    return 0;
}
```



【任务 19——TZOJ1073: 支配值数目】

题目描述

已知 $f[]$ 与 $g[]$ 两个整数数组，元素都已经从小到大排好序，请写一个程序，算出 $f[]$ 中比 $g[]$ 中元素大的对数。换句话说， $f[0]$ 比 $g[]$ 中多少个元素大， $f[1]$ 比 $g[]$ 中多少个元素大等，这些值的总和就是要求的答案。

举个例子，如果 $f[]$ 中有1, 3, 5, 7, 9，而 $g[]$ 中有2, 3, 4, 7, 8。

那么：

$f[0]$ 比 $g[]$ 中的所有元素都小；

$f[1]$ 比 $g[0]$ 大；

$f[2]$ 比 $g[0]$ 、 $g[1]$ 、 $g[2]$ 大；

$f[3]$ 比 $g[0]$ 、 $g[1]$ 、 $g[2]$ 大；

$f[4]$ 比 $g[0]$ 、 $g[1]$ 、 $g[2]$ 、 $g[3]$ 、 $g[4]$ 大；

所以答案是 $0+1+3+3+5=12$

输入描述

第一行为两个整数 m, n ($1 \leq m, n \leq 1000$)，分别代表数组 $f[]$, $g[]$ 的长度。

第二行有 m 个元素，为数组 $f[]$ 。

第三行有 n 个元素，为数组 $g[]$ 。

输出描述

输出支配值。

样例输入	样例输出
5 5 1 3 5 7 9 2 3 4 7 8	12

题目分析

双重循环去比较就好了，需要两个数组，数组一定要输入完再进行操作哦。



```

#include<bits/stdc++.h>
using namespace std;
int main()
{
    int m,n;
    cin>>m>>n;
    int f[1005],g[1005];
    for(int i=0;i<m;i++)
        cin>>f[i];
    for(int i=0;i<n;i++)
        cin>>g[i];
    int sum=0;
    for(int i=0;i<m;i++)
        for(int j=0;j<n;j++)
        {
            if(f[i]>g[j])
                sum=sum+1;
        }
    cout<<sum<<"\n";
    return 0;
}

```

其实我是因为觉得提示气人，我来教教你这个厉害的写法。

我们可以倒着来求，因为要求>的，正着无法知道相等数目的，然后让f的下标向前移，g的下标也移动，满足就要用下标推算出个数，然后不能到-1哦，下标只能是0和正整数。

```

#include<bits/stdc++.h>
using namespace std;
int main()
{
    int m,n;
    cin>>m>>n;
    int f[1005],g[1005];
    for(int i=0;i<m;i++)
        cin>>f[i];

```

```

for(int i=0;i<n;i++)
    cin>>g[i];
int i=m-1,j=n-1,sum=0;
for(;;)
{
    if(f[i]>g[j])
    {
        sum+=j+1;
        if(i)i--;
        else break;
    }
    else
    {
        if(j)j--;
        else break;
    }
}
cout<<sum<<"\n";
return 0;
}

```

【任务 20——TZOJ1166: C语言实验题——大小写转换】

题目描述

把一个字符串里所有的大写字母换成小写字母，小写字母换成大写字母。其他字符保持不变。

输入描述

输入为一行字符串，其中不含空格。长度不超过80个字符。

输出描述

输出转换好的字符串。

样例输入	样例输出
ABCDefgh123	abcdEFGH123

题目分析

字符串也是字符数组，但是其末尾是'\0'，需要多开辟一个元素。字符数组是可以直接cin进来然后自动转换的。这个题目一定要注意else if！判断大小写就是给他对应的字符就可以了，ASCII也不需要记那么多。

```
#include<bits/stdc++.h>
using namespace std;
int main()
{
    char s[85];
    cin>>s;
    for(int i=0;s[i];i++)
    {
        if(s[i]>='a'&&s[i]<='z')
            s[i]-=32;
        else if(s[i]>='A'&&s[i]<='Z')
            s[i]+=32;
    }
    cout<<s<<"\n";
    return 0;
}
```



附录2:常见问题及解答

提交状态列表中，有以下几类情形：

- **Wrong Answer**（答案错误），**绿色提示**，表示：给出了答案，但答案是错的
- **Runtime Error**（运行时错误），**绿色提示**，表示：运行到一半，程序发生了错误，没法继续运行了。
- **Compile Error**（编译错误），**蓝色提示**，表示：程序代码书写错误，程序根本运行不了。这个错误很不应该出现，因为此类错误意味着，你在本机也没法运行。当然也有可能在提交时把代码复制错了。
- **Time Limit Exceeded**（超出限定时间错误），**蓝色提示**，表示：程序运行时间超出限定还没结束，被强制终止了。这个一般在循环次数太多（甚至死循环）时发生，肯定是程序没考虑周到。（一般C语言提交的程序，运行时间限定在**1秒钟**以内，特殊题目会允许更多时间）
- **Memory Limit Exceeded**（超出限定内存错误），**蓝色提示**，表示：程序运行申请的内存空间超出了限定，被强制终止了。这个一般是定义了太大的数组、申请了太多内存、循环申请内存而没有及时释放内存导致的。（内存一般限制在**64M**以内）
- **Output Limit Exceed**（输出太多错误），**蓝色提示**，表示：程序运行时输出的数据太多了（不是多一两个，而是泛滥了），超出了系统接受的能力。
- **Presentation Error**（成功接受），**粉红色提示**，表示：你的程序几乎就对了，只是答案中多了一个空格，或少了一个空格，多了一个换行，或少了一个换行。一般发生在答案的开头多了或少了，以及末端多了或少了空格或空行。稍稍修改一下代码就完全正确了。
- **Accepted**（成功接受），**红色提示**，表示：你的程序经过在线判题系统（TZOJ）的充分判断，是完全正确的。
- **Judging**（正在判题），**蓝色提示**，表示：TZOJ正在对你的程序进行测试，你可以在几秒钟后刷新屏幕查看状态。
- **Waiting**（等待判题），**蓝色提示**，表示：TZOJ正在对别人的程序进行测试，你的还没轮到，需要稍等。你可以在几秒钟后刷新屏幕查看状态。

另见常见问题及解答 <http://tzcoder.cn/acmhome/FAQ.jsp>

