

书籍推荐：

《啊哈 C 语言！逻辑的挑战（修订版）》、《啊哈！算法》 作者：啊哈磊
《C++ 信息学奥赛零基础特训》

《编程竞赛宝典》（C++ 语言和算法入门 基础算法，张新华著）
《信息学竞赛金牌导航》
《一本通》（董永建著）

《我的第一本算法书》（APP 算法动画图解） 作者：[日] 石田保辉、[日] 宫崎修一
《算法图解》 作者：[美] Aditya Bhargava 译者：袁国忠

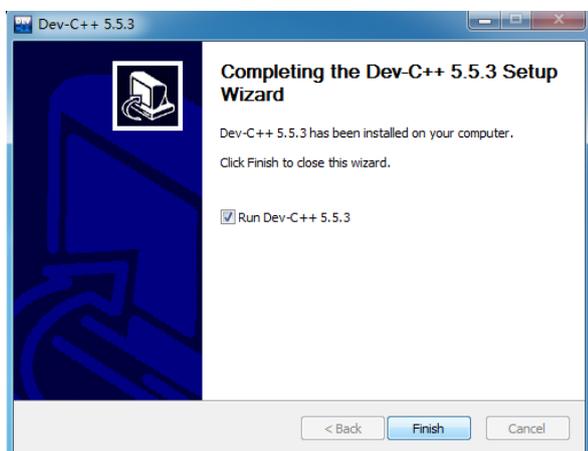
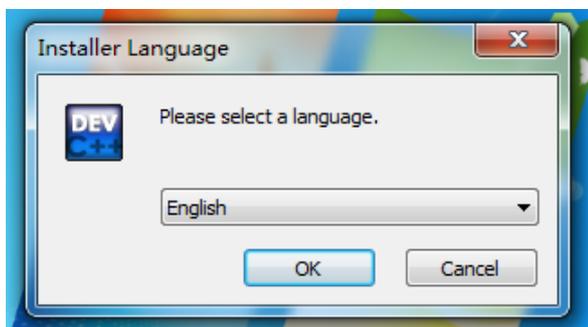
《小学生 c++ 趣味编程》（上册、下册） 作者：潘洪波

《CCF 中学生计算机程序设计》（入门篇、基础篇、提高篇）

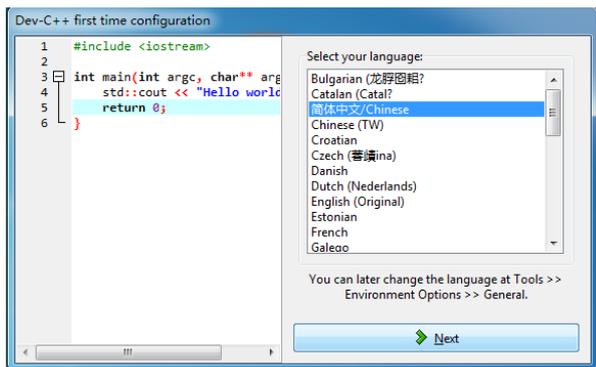


Dev-C++5.5.3 使用介绍

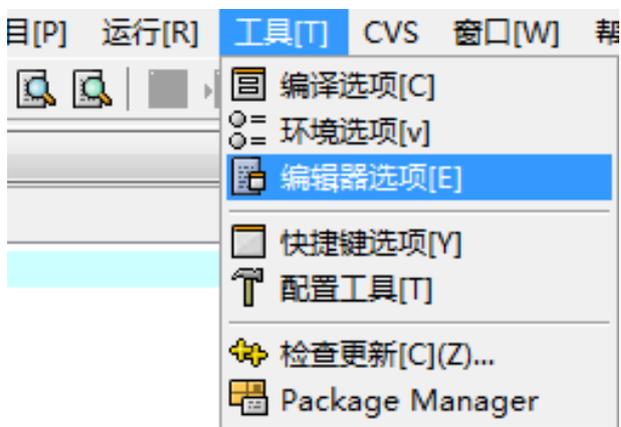
第一步：点击运行



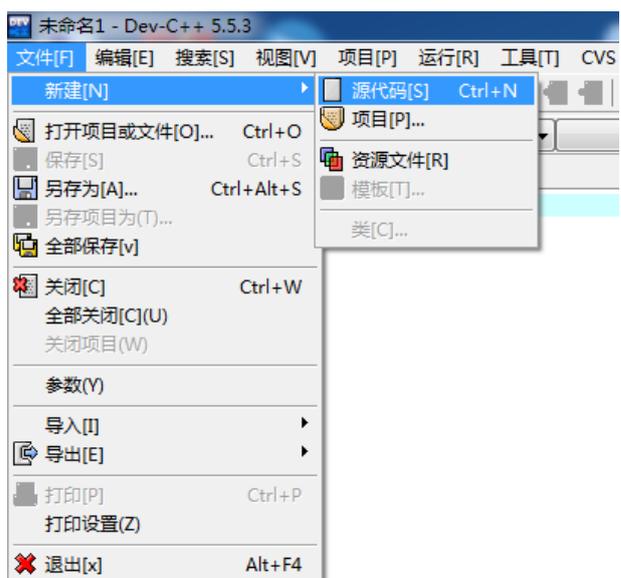
第二步：选择简体中文



第三步：点击“工具”，点击“编辑器选项”，把字号调整大。



第四步：点击“文件”，点击“新建”，点击“源代码”，就可以写代码了。



第五步：写完代码，点击“运行”，“编译运行”，程序运行起来，弹出黑色窗口。

顺序结构

输出输入

C++ 语言输入输出

1.0-1

```
#include<iostream>
using namespace std;
int main()
{
    cout<<"nihao"<<endl;
    return 0;
}
```

头文件
命名空间
主函数

1.0-2

```
#include<iostream>
using namespace std;
int main()
{
    cout<<"你好";
    cout<<"大家好";
    return 0;
}
```

cout 输出
" " 双引号
只有双引号之内可以中文输入，其他代码只能英文输入，否则程序无法运行。

1.0-3

```
#include<iostream>
using namespace std;
int main()
{
    cout<<"Ni hao!"<<endl;
    cout<<"dajiahao"<<endl;
    return 0;
}
```

endl 是 end of line, 的简单写法, 作用是输出结束, 然后输出下一行。
! 标点符号, 中文输入和英文输入不同, 认真观察。
大写 w
空格

1.0-4

```
#include<iostream>
using namespace std;
int main()
```

```
{
```

```
    int a; _____ int 是告诉计算机，输入的是整数
```

```
    cin>>a; _____ cin 输入
```

```
    cout<<a<<endl;
```

```
    return 0; _____ 输入：98
```

```
}
```

输出：98

1.0-5

```
#include<iostream>
using namespace std;
int main()
```

```
{
```

```
    float b; _____ float 是告诉计算机，输入的是浮点数（小  
    cin>>b; _____ 数）
```

```
    cout<<b<<endl; _____ 输入：9.8
```

```
    return 0; _____ 输出：9.8
```

```
}
```

1.0-6

```
#include<iostream>
using namespace std;
int main()
```

```
{
```

```
    char a; _____ char 是告诉计算机，输入的是字符（a,h,*,  
    cin>>a; _____ 等都是字符）
```

```
    cout<<" "<<a<<endl; _____ 输入：*  
    cout<<a<<a<<a<<endl; _____ 输出：  
    cout<<" "<<a<<endl; _____ *  
    return 0; _____ " " 空格
```

```
}
```

部分数据类型表

C++ 为程序员提供了种类丰富的内置数据类型和用户自定义的数据类型。下表列出了七种基本的 C++ 数据类型：

类型	关键字
布尔型	bool
字符型	char
整型	int
浮点型	float
双浮点型	double
无类型	void
宽字符型	wchar_t

类型	位	范围
char	1 个字节	-128 到 127 或者 0 到 255
unsigned char	1 个字节	0 到 255
signed char	1 个字节	-128 到 127
int	4 个字节	-2147483648 到 2147483647
unsigned int	4 个字节	0 到 4294967295
signed int	4 个字节	-2147483648 到 2147483647
short int	2 个字节	-32768 到 32767
unsigned short int	2 个字节	0 到 65,535
signed short int	2 个字节	-32768 到 32767
long int	8 个字节	-9,223,372,036,854,775,808 到 9,223,372,036,854,775,807
signed long int	8 个字节	-9,223,372,036,854,775,808 到 9,223,372,036,854,775,807
unsigned long int	8 个字节	0 to 18,446,744,073,709,551,615
float	4 个字节	+/- 3.4e +/- 38 (~7 个数字)
double	8 个字节	+/- 1.7e +/- 308 (~15 个数字)
long double	16 个字节	+/- 1.7e +/- 308 (~15 个数字)
wchar_t	2 或 4 个字节	1 个宽字符

注意：不同系统会有所差异。

c 语言输入输出

1.0-7

```
#include<cstdio>
using namespace std;
int main()
{
    printf("%d",16);
    return 0;
}
```

头文件与 c++ 不同

printf 输出

%d 输出格式: 整数

1.0-8

```
#include<cstdio>
using namespace std;
int main()
{
    printf("%f\n",16.7);
    printf("%f",30/7);
    return 0;
}
```

%f 输出格式: 浮点数 (小数)

/n 换行

"%f" 30/7 这样写导致错误

"%f" 30.0/7 这样写是对的

输出结果显示:

准确到小数点后 6 位

1.0-9

```
#include<cstdio>
using namespace std;
int main()
{
    printf("%.3f\n",30.0/7.0);

    printf("%.5f\n",30.0/7.0);

    printf("%9.2f",30.0/7.0);

    return 0;
}
```

%.3f 小数点后保留 3 位数
(四舍五入)

%.5f 小数点后保留 5 位数

%9.2f 小数点后保留 2 位数,
改变数字 9 大小, 会发现数据
输出格式的变化

1.0-10

```
#include<cstdio>
using namespace std;
int main()
{
    printf("%d %f\n",30/7,30/7);
    printf("%d %f\n",30/7,30.0/7.0);
    return 0;
}
```

空格

30/7 与 30.0/7.0 输出数据不同，仔细观察。

%f 30/7 容易导致错误

1.0-11

```
#include<cstdio>
using namespace std;
int main()
{
    float a;
    scanf("%f",&a);
    printf("%f",a/7);
    return 0;
}
```

30

4.285714

scanf 输入

1.0-12

```
#include<cstdio>
using namespace std;
int main()
{
    printf("%d %d %d",30%7,30%(-7),(-30)%7);
    return 0;
}
```

% 在这里是模，除的余数

1.0-13

```
#include<cstdio>
using namespace std;
int main()
{
    printf("%c",102);
    return 0;
}
```

%c 输出格式：字符

102 为什么会变成 f 是因为在计算机中，字符都是有编号的 (ASCII 表)

1.0-14

```
#include<cstdio>
using namespace std;
int main()
{
    int a;
    scanf("%d",&a);
    printf("%d",a);
    return 0;
}
```

%d 输入格式: 整数

测试:

输入: 16

输出: 16

1.0-15

```
#include<cstdio>
using namespace std;
int main()
{
    float a;
    scanf("%f",&a);
    printf("%f",a);
    return 0;
}
```

%f 输入格式: 浮点数 (小数)

测试:

输入: 16.7

输出: 16.700001

1.0-16

```
#include<cstdio>
using namespace std;
int main()
{
    char a;
    scanf("%c",&a);
    printf("%c",a);
    return 0;
}
```

%c 输入格式: 字符

测试:

输入: r

输出: r

算术运算

1.1-1

```
#include<iostream>
using namespace std;
int main()
{
    cout<<56+12<<endl;
    cout<<56-20<<endl;
    cout<<56*2<<endl;
    cout<<56/3<<endl;
    cout<<10%3<<endl;
    return 0;
}
```

+ 加号
- 减号
* 乘号
/ 除号
% 模 (除的余数)

1.1-2

```
#include<iostream>
using namespace std;
int main()
{
    cout<<20+50*2<<endl;
    cout<<(20+50)*2<<endl;

    cout<<100-60/3<<endl;
    cout<<(100-60)/3<<endl;

    cout<<20+10%3<<endl;
    cout<<(20+10)%3<<endl;

    return 0;
}
```

模样差不多，但是运算结果不同

* / % 优先级高于
+ -
乘法、除法、模 优先级高于加法、减法

整数

1.1-3

```
#include<iostream>
using namespace std;
int main(){
    float a,b;
    cin>>a>>b;
    cout<<a*b<<endl;
    return 0;
}
```

设计程序，计算长方形面积，输入边长，
输出面积；
输入：2 4 输出：8

1.1-4

```
#include<iostream>
using namespace std;
int main()
{
    float a,b;
    cin>>a>>b;
    cout<<a*b/2<<endl;
    return 0;
}
```

输入底和高，输出三角形面积。
输入：13 15
输出：97.5

注意：如果把 a,b 定义为整数，结果会是 97。

1.1-5 设计程序，计算正方形面积；

输入：4 输出：16

1.1-6 设计程序，计算长方形周长；

输入：4 6 输出：20

1.1-7 设计程序，计算三角形周长；

输入：4 5 6 输出：15

输入：5 6 7 输出：18

1.1-8 设计程序，计算长方体体积。

输入：2 3 5 输出：30

浮点数

1.1-9

```
#include<cstdio>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    printf("%.3f\n",30.0/7.0);
```

```
    printf("%.5f\n",30.0/7.0);
```

```
    printf("%9.2f",30.0/7.0);
```

```
    return 0;
```

```
}
```

```
#include<cstdio>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    float a;
```

```
    scanf("%f",&a); // 输入
```

```
    printf("%.2f\n",a); // 输出 \n 换行符
```

```
    printf("%9.2f",a); // 输出 9 是场宽为 9 .2 是保留小数点后 2 位数
```

```
    return 0;
```

```
}
```

```
#include<cstdio>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    float a,b,c;
```

```
    scanf("%f%f%f",&a,&b,&c); // 输入
```

```
    printf("%.2f\n",a+b+c); // 输出 \n 换行符
```

```
    printf("%9.2f",a+b+c); // 输出 9 是场宽为 9 .2 是保留小数点后 2 位数
```

```
    return 0;
```

```
}
```

%.3f 小数点后保留 3 位数

%.5f 小数点后保留 5 位数

%9.2f 小数点后保留 2 位数，
改变数字 9 大小，会发现数据
输出格式的变化

1.1-10

```
#include<iostream>
#include<iomanip>
using namespace std;
int main()
{
    cout<<fixed<<setprecision(12)<<30.8/7.1<<endl;
    return 0;
}
```

fixed<<setprecision 格式函数

这段代码作用是把计算结果准确到小数点后 12 位。

笔记本 3 元 9 角 1 本，老师给 a 元 b 角， $(0 \leq a \leq 1000, 0 \leq b \leq 9)$ ，最多买几个笔记本？

输入：2 个整数，a 和 b。

输出：1 个整数，表示几本笔记本。

输入：78 9 输出： 20

输入：66 6 输出： 17

1.1-11

```
#include<iostream> // 筱雅
using namespace std;
int main()
{
    int a, b, x, s;
    cin>>a>>b;
    x=a*10+b;
    if(x%(3*10+9)==0)
        cout<<x/(3*10+9);
    else
        cout<<x/(3*10+9);
    return 0;
}
```

价钱统计（2016 海淀区竞赛题）

【题目描述】 夏天到了，超市里摆满了各种各样的应季水果。现在知道：西瓜的价钱是每斤 1.2 元；桃子的价钱是每斤 3.5 元；葡萄的价钱是每斤 4.5 元；苹果的价钱是每斤 5 元。现在分别给出上述四种所购买的斤数（均不超过 20），请你编写程序帮助售货员阿姨计算并依次输出顾客购买四种水果需要的钱数及总钱数。

【输入格式】 输入文件 `jiaqian.in` 中只有一行，包含四个符合题目要求的非负数，依次表示需要购买西瓜、桃子、葡萄和苹果的斤数。两两之间用一个空格分隔。

【输出格式】 输出文件 `jiaqian.out` 中共有五行，每行仅包含一个数，依次代表购买西瓜、桃子、葡萄、苹果所需的钱数，以及购买四种的总钱数。

所有数据均采用四舍五入保留一位小数。（若小数点后为 0 也可仅输出整数）

【样例输入】 4 3 2.5 6

【样例输出】

4.8

10.5

11.3

30.0（仅输出 30 也正确）

56.6

1.1-12

```
#include<bits/stdc++.h>// 筱雅
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    float x1, x2, x3, x4, a, b, c, d, e;
```

```
    float f1=1.2, f2=3.5, f3=4.5, f4=5.0;
```

```
    cin>>x1>>x2>>x3>>x4;
```

```
    a=x1*f1;
```

```
    b=x2*f2;
```

```
    c=x3*f3;
```

```
    d=x4*f4;
```

```
    e=a+b+c+d;
```

```
    printf("%.1f\n%.1f\n%.1f\n%.1f\n%.1f", a, b, c, d, e);
```

```
    return 0;
```

```
}
```

时间

1.1-13

```
#include<iostream>
using namespace std;
int main(){
    int a;
    cin>>a;
    cout<<a/60<<" 小时 "<<endl;
    cout<<a%60<<" 分钟 ";
    return 0;
}
```

输入分钟，输出小时和分钟。

输入：80

输出：1 小时

20 分钟

1.1-14

```
#include<iostream>
using namespace std;
int main() {
    int a, b, c;
    cin>>a>>b>>c;
    cout<<3600*a+60*b+c;
    return 0;
}
```

输入小时、分、秒，转换为秒。

输入：2 3 4 （代表2小时3分钟4秒）

输出：7384 （7384 秒）

1.1-15

```
#include<iostream>
using namespace std;
int main() {
    int a, b, c, m;
    cin>>m;
    a=m/3600;// 小时
    b=m%3600/60;// 分钟
    c=m%60;// 秒
    cout<<a<<" 小时 "<<b<<" 分 "<<c<<" 秒 ";
    return 0;
}
```

输入秒，转换为小时、分、秒。

输入：7384 （7384 秒）

输出：2 3 4 （代表2小时3分钟4秒）

爸爸今天工作，从 a 时 b 分到 c 时 d 分，请问，用了几小时几分钟？

输入：1 20 1 50

输出：0 30

输入：2 50 4 20

输出：1 30

拆位

1.1-16

```
#include<iostream>
using namespace std;
int main()
{
    int n1, n2, n3, x;
    cin>>x;
    n3=x/100; // 百位数
    n2=x/10%10; // 十位数
    n1=x%10; // 个位数
    cout<<100*n1+10*n2+n3;
    return 0;
}
```

输入：123

输出：321

变量和常量

变量

1.2-1

```
#include<iostream>
using namespace std;
int main()
{
    int shengao=60;
    cout<<shengao<<endl;

    shengao=shengao+3;
    cout<<shengao<<endl;

    shengao=shengao+5;
    cout<<shengao<<endl;
    return 0;
}
```

小明前年身高 60 厘米，去年长了 3 厘米，今年又长了 5 厘米，问：小明这几年身高是多少？

这 3 个 shengao 都是 60

这 2 个 shengao 都是 63

这个 shengao 是 68

1.2-2

```
#include<iostream>
using namespace std;
int main()
{
    int shengao=60;
    cout<<shengao<<endl;
    shengao+=3;
    cout<<shengao<<endl;
    shengao+=5;
    cout<<shengao<<endl;
    return 0;
}
```

上一段代码容易看懂，但是比较繁琐。这段代码作用跟上一段一样，简洁了很多。使用 +=，代码简洁又好看！

可以变化的叫做变量！

小明买的裤子 40 厘米，长了，他想剪掉 5 厘米。回家一说，全家人都答应帮他剪裤子。吃完晚饭，爷爷剪掉 5 厘米，睡觉前，奶奶剪掉 5 厘米，早晨起床后，妈妈剪掉 5 厘米，中午爸爸回家又剪掉 5 厘米，问，现在裤子多长？

1.2-3

```
#include<iostream>
using namespace std;
int main()
{
    int kuzi=40;          cout<<kuzi<<endl;  —— 小明买的裤子

    kuzi=kuzi-5;        cout<<kuzi<<endl;  —— 爷爷剪的
    kuzi=kuzi-5;        cout<<kuzi<<endl;  —— 奶奶剪的
    kuzi=kuzi-5;        cout<<kuzi<<endl;  —— 妈妈剪的
    kuzi=kuzi-5;        cout<<kuzi<<endl;  —— 爸爸剪的

    return 0;
}
```

1.2-4

这样写代码简洁！

```
#include<iostream>
using namespace std;
int main()
{
    int kuzi=40;

    kuzi-=5;
    kuzi-=5;
    kuzi-=5;
    kuzi-=5;

    cout<<kuzi<<endl;
    return 0;
}
```

一棵竹子，前天高 2 米，昨天增长了十分之一，今天又增长了十分之一，问：竹子现在多高？

1.2-5

```
#include<iostream>
using namespace std;
int main()
{
    float a;
    a=2;          cout<<a<<endl;
    a=a*1.1;     cout<<a<<endl;
    a=a*1.1;     cout<<a<<endl;

    return 0;
}
```

虽然模样一样，都是 a，
数值大小可不一样，会变化，
叫变量。

1.2-6

```
#include<iostream>
using namespace std;
int main()
{
    float a;
    a=2;
    a*=1.1;
    a*=1.1;
    cout<<a<<endl;
    return 0;
}
```

用 *= 来写，是不是代码
干净利落？

小明买了一块大蛋糕，重 3 斤，昨天切掉一半，今天又切掉一半，问：现在蛋糕多大？

1.2-7

```
#include<iostream>
using namespace std;
int main()
{
    float a;
    a=3;          cout<<a<<endl;
    a=a/2;       cout<<a<<endl;    —— 昨天切掉一半
    a=a/2;       cout<<a<<endl;    —— 今天切掉一半

    return 0;
}
```

1.2-8

这样写代码简洁！

```
#include<iostream>
using namespace std;
int main()
{
    float a;
    a=3;
    a/=2;
    a/=2;
    cout<<a<<endl;
    return 0;
}
```

变量交换。

a 桶装 10 升水，b 桶装 20 升水，如何交换 2 桶水？

1.2-9

```
#include<iostream>
using namespace std;
int main()
{
    int a,b,c;
    a=10;
    b=20;
    c=a;
    a=b;
    b=c;
    cout<<a<<" "<<b<<endl;
    return 0;
}
```



a 10升水



b 20升水

如何把两桶水相互换过来？

1, 先找一个c桶,



2, 然后把a桶的水倒入c桶

3, 然后, 把b桶的水倒入a桶

4, 最后, 把c桶水倒入b桶。

变量的自增和自减

1.2-10

```
#include<iostream>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    int a,b;
```

```
    a=10;
```

```
    b=20;
```

```
    a++;
```

```
    b--;
```

```
    cout<<a<<" "<<b<<endl;
```

```
    return 0;
```

```
}
```

++ 自增运算符

-- 自减运算符

_____ a++ 就是 a=a+1

b-- 就是 b=b-1

1.2-11

```
#include<iostream>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    int a,b;
```

```
    a=10;
```

```
    b=20;
```

```
    ++a;
```

```
    --b;
```

```
    cout<<a<<" "<<b<<endl;
```

```
    return 0;
```

```
}
```

1.2-12

```
#include<iostream>
using namespace std;
int main()
{
    int a,b;
    a=10;
    b=20;
    a=b++;
    cout<<a<<" "<<b<<endl;
    return 0;
}
```

1.2-13

```
#include<iostream>
using namespace std;
int main()
{
    int a,b;
    a=10;
    b=20;
    a=b--;
    cout<<a<<" "<<b<<endl;
    return 0;
}
```

单独使用自增或者自减,

a++,++a

a--,--a

结果一样。

但是, 在赋值语句中用时, 就不同了。

1.2-14

```
#include<iostream>
using namespace std;
int main()
{
    int a,b;
    a=10;
    b=20;
    a=++b;
    cout<<a<<" "<<b<<endl;
    return 0;
}
```

1.2-15

```
#include<iostream>
using namespace std;
int main()
{
    int a,b;
    a=10;
    b=20;
    a=--b;
    cout<<a<<" "<<b<<endl;
    return 0;
}
```

常量

1.2-16

```
#include<iostream>//const 常量
using namespace std;
int main()
{
    const int n=18;
    n=n*n;// 错误写法, 不运行, 不生效
    cout<<n;
}
```

用 const 定义 n 为常量, 只能读, 不能改。

```
#include<iostream>
const int n=10;// 数组最大 10 个数字, const 常量
using namespace std;
int a[n];
int main() {
    for (int i=0; i<n; i++) cin>>a[i];
    for (int i=0; i<n; i++) cout<<a[i]<<" ";
    return 0;
}
```

数组大小设定, 不可改变。

1.2-17

```
#include<iostream>
using namespace std;
int main()
{
    const float PI=3.14159265;
    float radius, area;
    radius=10;
    area=PI*radius*radius;
    cout<<area<<endl;
    return 0;
}
```

常量一般用大写字母表示, 以区别变量。

常量通常有常识性的、明确的含义, 例如 PI 代表圆周率。

radius 半径
area 面积

数据转换

整型、浮点型转换

1.3-1

相同大小的两数相除，结果不同。

```
#include<iostream>
using namespace std;
int main()
{
    cout<<50/3<<endl;
    cout<<50/3.0<<endl;
    cout<<50.0/3<<endl;
    return 0;
}
```

1.3-2

计算数据前要先定义数据类型，否则容易出错！

```
#include<iostream>
using namespace std;
int main()
{
    cout<<50/3<<endl;
    cout<<50.0/3<<endl;
    return 0;
}
```

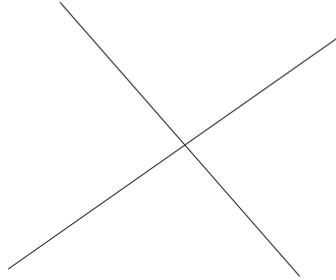
这里，没有告诉我们是什么数据类型。

这里，告诉我们 50 是浮点型数据。

观察一下结果，哪个更准确？

整型、长整型转换

```
#include<iostream>
using namespace std;
int main()
{
    cout<<1234567890*3;
    return 0;
}
```



这段代码运行，结果是错误的，该如何处理？

大多数时候，需要设置数据的类型，否则计算机无法计算或者容易出错。

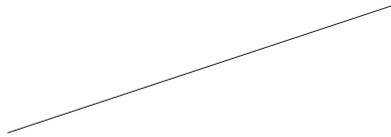
1.3-3

```
#include<iostream>
using namespace std;
int main()
{
    long long a;
    a=1234567890*3;
    cout<<a<<endl;
    return 0;
}
```

这段代码运算结果还是错误的，因为没有设置1234567890这个数字是哪个类型。

1.3-4

```
#include<iostream>
using namespace std;
int main()
{
    long long a, b, c, d;
    a=b=c=1234567890;
    d=a+b+c;
    cout<<d<<endl;
    return 0;
}
```



用 long long 把参与运算的所有数据都定义成长整型，结果才是正确的（告诉计算机，我要计算的是一个很大的数）。

字符型、整型转换

1.3-5

```
#include<iostream>
using namespace std;
int main() {
    int a, b;
    char c, d;
    a='b';
    b='B';
    c=65;
    d=97;
    cout<<a<<endl;
    cout<<b<<endl;
    cout<<c<<endl;
    cout<<d<<endl;
    return 0;
}
```

定义 a 为整型数据，但是却等于一个字符，字符和整数之间为什么可以转换？

大写、小写对应的数字不一样？

1.3-6

```
#include<iostream>
using namespace std;
int main()
{
    int a, b;
    a='b';
    b='D';
    cout<<a-b<<endl;
    cout<<a+b<<endl;
    cout<<a*b<<endl;
    cout<<a/b<<endl;
    cout<<a%b<<endl;
    return 0;
}
```

还可以加、减、乘、除、模？

计算机中，字符、符号、及各种命令是有编号的，请看 ASCII 表。

ASCII 对照表

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	NUL	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	DC1	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

数学进阶

一本通 1033 线段：已知线段的两个端点的坐标 $A(X_a, Y_a)$ ， $B(X_b, Y_b)$ ，求线段 AB 的长度，保留到小数点后 3 位。

输入

第一行是两个实数 X_a, Y_a ，即 A 的坐标。

第二行是两个实数 X_b, Y_b ，即 B 的坐标。

输入中所有实数的绝对值均不超过 10000。

输出：一个实数，即线段 AB 的长度，保留到小数点后 3 位。

输入样例

1 1

2 2

输出样例

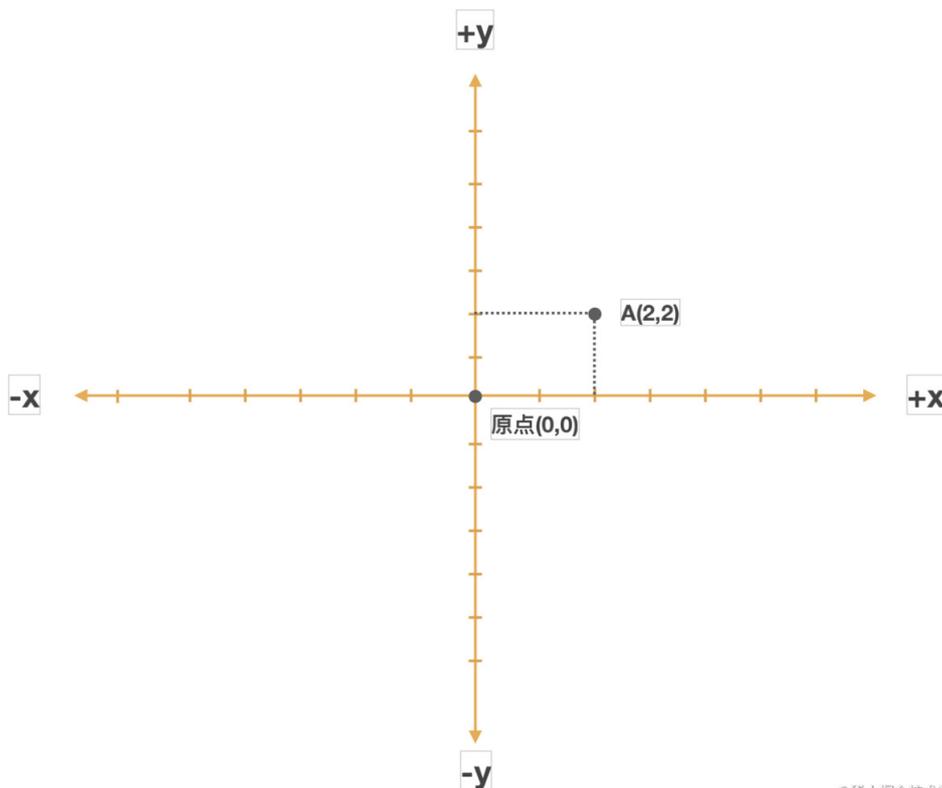
1.414

讲解视频：

https://www.bilibili.com/video/BV1hY411P78K/?spm_id_from=333.788

<https://www.bilibili.com/video/BV1bx411M7Ya?t=39.1>

<https://www.bilibili.com/video/BV1AT41117tN?t=210.3>



1.4-2-a

```
#include <bits/stdc++.h> // 开方函数: double sqrt(double x)
using namespace std;
int main()
{
    double xa, ya, xb, yb;
    scanf("%lf %lf %lf %lf", &xa, &ya, &xb, &yb);
    printf("%.3f", sqrt((xa-xb)*(xa-xb) + (ya-yb)*(ya-yb)));
    return 0;
}
```

1.4-2-b

```
#include <bits/stdc++.h> // 勾股定理
using namespace std;
int main() // 乘方函数: double pow(double a, double b)
{
    double xa, ya, xb, yb;
    cin>>xa>>ya>>xb>>yb;
    cout<<fixed<<setprecision(3)<<sqrt(pow(xa-xb, 2) + pow(ya-yb, 2));
    return 0;
}
```

一本通 1034: 计算三角形面积

时间限制: 1000 ms

内存限制: 65536 KB

【题目描述】平面上有一个三角形，它的三个顶点坐标分别为 (x_1, y_1) , (x_2, y_2) , (x_3, y_3) ，那么请问这个三角形的面积是多少，精确到小数点后两位。

【输入】输入仅一行，包括 6 个双精度浮点数，分别对应 $x_1, y_1, x_2, y_2, x_3, y_3$ 。

【输出】输出也是一行，输出三角形的面积，精确到小数点后两位。

【输入样例】0 0 4 0 0 3

【输出样例】6.00

讲解视频

<https://www.bilibili.com/video/BV1aY411w7hi/?t=6.7>

2. 海伦公式: 已知三条边的长度求三角形面积

设 $a = |\vec{a}|, b = |\vec{b}|, c = |\vec{c}|$,

已知 $\vec{a} = (x_2 - x_1, y_2 - y_1), \vec{b} = (x_3 - x_1, y_3 - y_1), \vec{c} = (x_3 - x_2, y_3 - y_2)$

有:

- $a = |\vec{a}| = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$
- $b = |\vec{b}| = \sqrt{(x_3 - x_1)^2 + (y_3 - y_1)^2}$
- $c = |\vec{c}| = \sqrt{(x_3 - x_2)^2 + (y_3 - y_2)^2}$
- 半周长 $p = \frac{1}{2}(a + b + c)$
- 海伦公式, 三角形面积 $S = \sqrt{p(p - a)(p - b)(p - c)}$

1.4-3

```
#include <bits/stdc++.h> // 一本通 1034 计算三角形面积 海伦公式
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    double x1, y1, x2, y2, x3, y3, a, b, c, p;
```

```
    cin >> x1 >> y1 >> x2 >> y2 >> x3 >> y3;
```

```
    a = sqrt((x2-x1)*(x2-x1) + (y2-y1)*(y2-y1)); // |a|
```

```
    b = sqrt((x3-x1)*(x3-x1) + (y3-y1)*(y3-y1)); // |b|
```

```
    c = sqrt((x3-x2)*(x3-x2) + (y3-y2)*(y3-y2)); // |c|
```

```
    p = (a + b + c) / 2; // p 半周长
```

```
    cout << fixed << setprecision(2) << sqrt(p*(p-a)*(p-b)*(p-c)) << endl;
```

```
    return 0;
```

```
}
```