

CS100 算法入门

C++语言入门

HelloWorld: C++程序的基本结构

凡是用紫色标注的内容
都是给有志从事IT/互联网工作的同学的建议
如果你仅仅为了参加竞赛
你可以忽略这些内容

```
1 //  
2 // main.cpp  
3 // test  
4 //  
5 // Created by Ge, Xiao on 25/02/2017.  
6 // Copyright © 2017 Ge, Xiao. All rights reserved.  
7 //  
8  
9 #include <iostream>  
10  
11 int main(int argc, const char * argv[]) {  
12     // insert code here...  
13     std::cout << "Hello, World!\n";  
14     return 0;  
15 }
```

→ 注释(不影响运行, 但是建议养成写注释的习惯)

→ 预处理代码(你们理解成C++版外部库导入就行了)

纳尼? 你想知道更多? 那么你自己去查一下关于C++预编译指令

→ 主函数声明(整个程序的入口)

→ 语句(一个输出语句+一个函数返回语句)

恭喜! 你已经可以写C++程序了^^

开发环境(IDE)

- 集成开发环境 (Integrated Development Environment)
- 本课程推荐的IDE是Xcode。如果你已经习惯用其他的IDE，不需要更换。语音只是工具，IDE是工具的工具

其实高手不一定需要IDE的，但你还是需要用IDE

因为你必须学会**调试**



编程界终极真理

- 没有一次性写对的程序 (真理)
- 没有不存在bug的项目 (大大的真理)

- 修bug技术谁最强？中国山东找蓝翔

C++基本数据类型

```
9 #include "basic.hpp"
10 #include <iostream>
11 #include <string> // 使用string必须加这句
12
13 void basicTypeDemo(){
14     int a=1; // 整数
15     std::cout<<"a="<<a<<std::endl;
16     // int a={1};
17     // int a{1}
18
19     double b=3.14; // 浮点数
20     std::cout<<"b="<<b<<std::endl;
21
22     int data[5]={1,2,3,4,5}; // 数组: 一定要指明大小, 否则无法编译
23     // int data[5]={} // 全部初始化为0
24     std::cout<<"data="<<data<<std::endl;
25     printf("%d %d %d %d %d\n",data[0],data[1],data[2],data[3],data[4]);
26
27     std::string s="abc"; // 字符串
28     std::cout<<"s="<<s<<std::endl;
29     std::cout<<"s[2]="<<s[2]<<std::endl; // 字符串本质上就是字符数组
30
31     // 为什么没有布尔类型? 因为c++直接使用整数0和1表示布尔类型
32     // 如果你一定要输出true/false, 那么如下
33     std::cout<<std::boolalpha<<(1==2)<<std::endl;
34 }
35
36
```

→ 老师字好小啊?? 一般竞赛题代码量50-150行 (有没有感觉上档斥了? 工程项目代码上10000是很正常的)

→ 声明变量后尽量立刻初始化。不是必须的, 但是很好的习惯可以避免不必要的问题

→ std::boolalpha这什么鬼?
请你自己去查一下关于cout格式化输出

```
a=1
b=3.14
data=0x7fff5fbff770
1 2 3 4 5
s=abc
s[2]=c
false
Program ended with exit code: 0
```

→ 这又是什么鬼?

这是内存地址(还记得16进制吗), 数组本质上是一块连续的内存区域

C++输入输出

```
35 void ioDemo(){
36     int a;
37     char b;
38     std::string s;
39
40     // 1.流式输入输出
41     std::cout<<"Please input one int, one char, one string"<<std::endl;
42     std::cin>>a>>b>>s;
43     std::cout<<"a="<<a<<" b="<<b<<" s="<<s<<std::endl;
44
45     // 2.格式化输入输出
46     std::cout<<std::endl<<"Please input one int, one char, one string"<<std::endl;
47     scanf("%d %c %s",&a,&b,&s[0]);
48     printf("a=%d b=%c s=%s\n",a,b,s.c_str());
49 }
50
51 using namespace std;
52
53 void basicOprDemo(){
54     cout<<"5/2="<<(5/2)<<endl; // 整数除法
```

```
Please input one int, one char, one string
1 a hello
a=1 b=a s=hello
```

```
Please input one int, one char, one string
2 b world
a=2 b=b s=world
Program ended with exit code: 0
```

→ string表示字符串，后面会讲到。字符串是除数值变量外使用最频繁的类型之一

→ 这个区域叫做控制台(Console)，是最基本的与程序交互的地方



C++基本运算

```
12 #include <iostream>
13 |
14 using namespace std;
15
16 void basicOprDemo(){
17     cout<<"5/2="<<(5/2)<<endl; // 整数除法
18     cout<<"5.0/2="<<(5.0/2)<<endl; // 浮点数除法
19     cout<<"5/2="<<((double)5/2)<<endl; // 类型转换运算符
20     cout<<"5%2="<<(5%2)<<endl; // 取模
21     // cout<<"5.0%2="<<(5.0%2)<<endl; // 编译错误
22
23     int a=1, b=a++, c=++a; // 自增运算符, 赋值运算符
24     printf("a=%d b=%d c=%d\n",a,b,c);
25     a*=2; // 运算后赋值运算符
26     printf("a*=2 is %d\n",a);
27
28     cout<<"1>2 is "<<(1>2)<<endl; // 记得布尔值就是0和1?
29     cout<<"3!=4 is "<<boolalpha<<(3!=4)<<endl;
30     cout<<"1>2 || 3!=4 is "<<(1>2 || 3!=4)<<endl; // 逻辑运算
31
32     // 混合运算
33     cout<<"a++%2==1 && 3!=4 || 7>=7 is "<<(a++%2==1 && 3!=4 || 6>=7)<<endl;
34 }
```

→ 重复写很多std::是不是很烦? 加这句就可以了
请你自己去查一下关于C++的命名空间是怎么回事



```
5/2=2
5.0/2=2.5
5/2=2.5
5%2=1
a=3 b=1 c=3
a*=2 is 6
1>2 is 0
3!=4 is true
1>2 || 3!=4 is true
a++%2==1 && 3!=4 || 7>=7 is false
Program ended with exit code: 0
```

与Python很像? 高级语言也就这点花头

C++运算符优先级

优先级	类型	运算符	名称或含义	优先级	类型	运算符	名称或含义		
1	自增运算	后置++	后置自增运算符	6	逻辑运算	>	大于		
		后置--	后置自减运算符			>=	大于等于		
	数组下标	[]	数组下标			<	小于		
	自定义优先级	()	圆括号			<=	小于等于		
	对象运算	.	成员选择 (对象)	7		==	等于		
		->	成员选择 (指针)	8		!=	不等于		
2	算术运算	-	负号运算符	9	位运算	&	按位与		
	类型转换	(类型)	强制类型转换	10		^	按位异或		
	自增运算	前置++	前置自增运算符	11			按位或		
		前置--	前置自减运算符	12	&&	逻辑与			
	指针运算	*	取值运算符	13	逻辑运算		逻辑或		
		&	取地址运算符	特殊运算符		?:	条件运算符		
	逻辑运算	!	逻辑非运算符		=	赋值运算符			
	位运算	~	按位取反运算符		/=	除后赋值			
	特殊运算符	sizeof	长度运算符		*=	乘后赋值			
					%=	取模后赋值			
3	算术运算	/	除		+=	加后赋值			
		*	乘		-=	减后赋值			
		%	余数 (取模)		<<=	左移后赋值			
		+	加	>>=	右移后赋值				
4		-	减	14	赋值运算	&=	按位与后赋值		
		<<	左移			^=	按位异或后赋值		
5	位运算	>>	右移					=	按位或后赋值
								15	特殊运算符

→ 绿色部分是常用运算符，必须掌握
不过你用几次很快就会很熟的

→ 黄色部分是某些特定语法用到的算符
以后用到的时候我会逐步介绍



吓到宝宝了？不要怕！记住如下规则就能应对99%情况：

- 1.括号>加减乘除>比大小
- 2.如果没把握就加括号

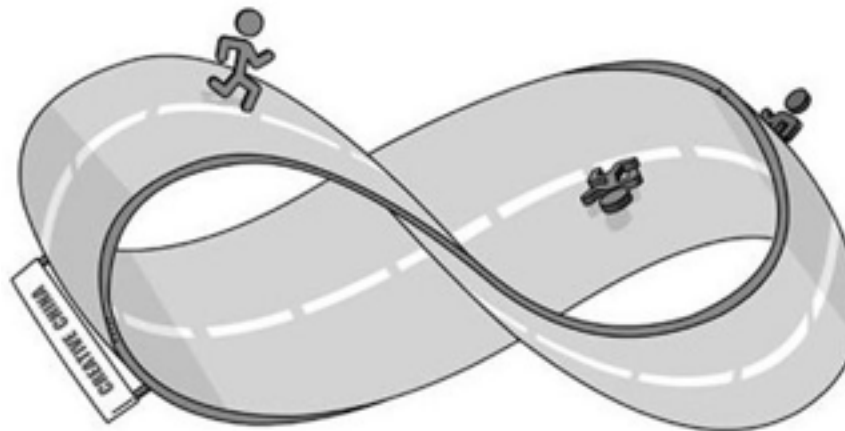
C++程序流程结构

```
12 #include <iostream>
13
14 using namespace std;
15
16 void basicStructureDemo(){
17     int a;
18     printf("input 1 int a : "); // 顺序结构
19     scanf("%d",&a);
20     printf("a=%d\n",a);
21
22     if (a%2==1){ // 分支结构
23         // if (a%2) { // 回忆一下布尔值就是0和1, 所以也可以这么写。这是一种技巧, 但是影响可读性, 一般不推荐
24             printf("%d is odd\n",a);
25         }else{
26             printf("%d is even\n",a);
27         }
28         printf("%d is %s\n",a,a%2==0?"even":"odd"); // 条件运算符。是不是觉得很拽? ^^
29
30         for (int i=0;i<a;i++){ // 循环结构
31             printf("repeat for %d times. This is No.%d\n",a,i+1);
32         }
33     }
34 }
```

三大程序流程：顺序、分支、循环
也就这点花头😓

→ 你应该知道还有两种循环叫while/do-while循环
请你自己查一下c++里这两种循环怎么用
其实，for循环是最灵活的，原则上任何while/do-while
循环都可以用for循环替代，试一下看看😏

```
input 1 int a : 3
a=3
3 is odd
3 is odd
repeat for 3 times. This is No.1
repeat for 3 times. This is No.2
repeat for 3 times. This is No.3
Program ended with exit code: 0
```

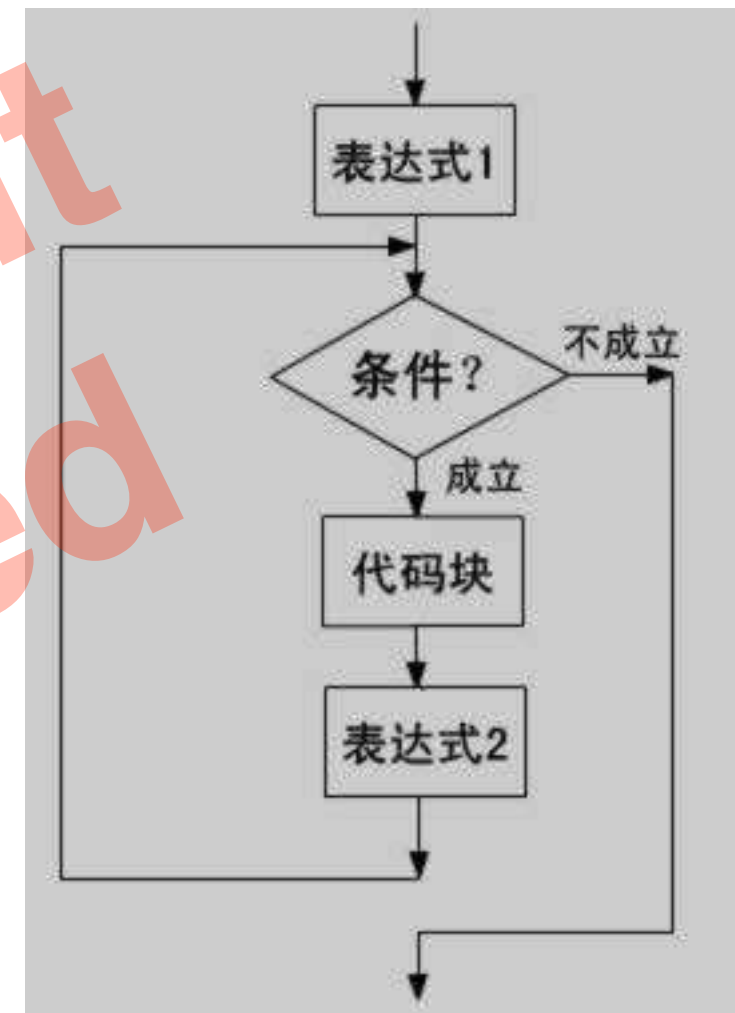


for循环：很有花头

**for (表达式1; 条件; 表达式2) {
 代码块;
}**

```
16 void forDemo(){  
17  
18     for (int i=0;i<10;i++,cout<<i<<" ");    // 循环体写到for语句  
19     cout<<endl;  
20  
21     for (int i=0,j=10;i<=j;i+=2,j-=3){    // 两个循环控制变量  
22         cout<<"("<<i<<","<<j<<") ";  
23     }  
24     cout<<endl;  
25  
26     for (int i=0;;i++) {    // 死循环  
27         cout<<"Forever running..."<<endl;  
28         if (i>5)  
29             break;    // 跳出死循环  
30     }  
31     cout<<endl;  
32 }  
33
```

```
1 2 3 4 5 6 7 8 9 10  
(0,10) (2,7) (4,4)  
Forever running...  
Forever running...  
Forever running...  
Forever running...  
Forever running...  
Forever running...  
Forever running...
```



怎样学习一种新语言？

郑重声明🤨

- 1.下次你学第3+门语言的时候，不要再来找我了，**你自己搞定**
- 2.这是唯一一节正儿八经讲语法的课。后面会以算法逻辑为主



小case



精通日韩等8国语言 温州姑娘被称中国最牛中学生

2010年06月07日 06:57:19 来源：温州网-温州商报 查看评论 手机看新闻 字体：大 中 小

核心提示：精通英语、日语、韩语等8种语言，会用100多种语言向国外友人问好。永嘉的周由希成为世博唯一乡村馆形象大使...

第一种算法：暴力枚举

最大和子串问题

输入一个整数序列，输出其中总和最大的连续子序列之和

样例输入：

9

13 -21 11 -31 32 22 -12 33 23

样例输出：

98

(注：对应的子序列为32 22 -12 33 23)

不要鄙视简单粗暴，有时候最直接的办法就是最简单的办法



最大和子串问题：第一次调试

```
14 void countMaxSubStr(){
15     // 输入
16     int n;
17     scanf("%d",&n);
18     int a[n];
19     for(int i=0;i<n;i++){
20         scanf("%d",&a[i]);
21     }
22
23     int max=0; // 当前最大值
24     for (int left=0;left<n;left++){ // 子串的左边界
25         for (int right=left;right<n;right++){ // 子串的右边界
26             int sum=0; // 求和
27             for (int i=left;i<right;i++){
28                 sum+=a[i];
29             }
30             if (sum>max){ // 求最大值
31                 max=sum;
32             }
33         }
34     }
35     cout<<"MaxSubStr Sum="<<max<<endl;
36 }
37
```

```
9
13 -21 11 -31 32 22 -12 33 23
MaxSubStr Sum=75
Program ended with exit code: 0
```

→ so我们就来简单粗暴地枚举所有子串~

→ 竞赛中你可以用a,b,i,i2,k,kk这种简单粗暴的变量名
不过我建议你尽量用有意义一些的变量名，像
left,right,sum这种。这是一个好习惯

如果你有志从事编程行业，那么你必须学会用有意义的变量名。否则一定会被同事鄙视

纳尼？结果不对？
来调试一下！



纳尼？

最大和子串问题：解答

```
14 void countMaxSubStr(){
15     // 输入
16     int n;
17     scanf("%d",&n);
18     int a[n];
19     for(int i=0;i<n;i++){
20         scanf("%d",&a[i]);
21     }
22
23     int max=0; // 当前最大值
24     for (int left=0;left<n;left++){ // 子串的左边界
25         for (int right=left;right<n;right++){ // 子串的右边界
26             int sum=0; // 求和
27             for (int i=left;i<=right;i++){
28                 sum+=a[i];
29             }
30             if (sum>max){ // 求最大值
31                 max=sum;
32             }
33         }
34     }
35     cout<<"MaxSubStr Sum="<<max<<endl;
36 }
37
```

→ 差一个字符就可能导致错误（引发血案）这是程序设计竞赛的特征



```
9
13 -21 11 -31 32 22 -12 33 23
MaxSubStr Sum=98
Program ended with exit code: 0
```


作业

3.最小和子串问题 (minsubstr.cpp)

题目不用我在写一遍了把😏，最大改成最小

4. (选做) 写一个程序输出所有如下三位数等式，满足1-9每个数字出现一次且仅出现一次 (equation.cpp)

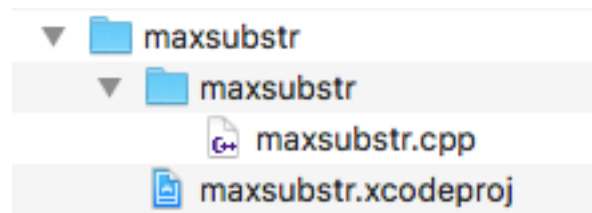
例如：346+581=927



5. (选做) 我写的最大和子串问题用了三重循环，这是最无脑的办法。有没有更快的方法？两重循环行不行？一重循环行不行？写一个你能想到的最快的方法，并且自己验证正确性 (minsubstrbetter.cpp)

(写出正确一重循环解答的有红包^^)

PS: xcode项目结构 (Mac系统)



→ 因为Xcode必须创建工程, 所以你的每个程序应该长成这样的目录结构

→ *.xcodeproj是工程入口文件, 双击这个文件就可以在xcode打开这个工程

→ *.cpp就是你需要提交的代码 (这个课上每个问题都对应一个cpp文件), 交作业的时候提交这个文件就可以了。工程创建的时候默认名字是main.cpp, 改一下就可以了