周训第一节

C/C++从入门到入土

主讲人：莫迪茵

等待鸽多

Japan Osaka-shi

Scan the QR code to add me as friend

队长 吴岱琳

壹拾壹拾一

吉布提

扫一扫上面的二维码图案，加我为朋友。

副队长 苏凯伦

Scavenger.

扫一扫上面的二维码图案，加我为朋友。

23级负责人 徐嘉哲

0nedeer

南极洲

扫一扫上面的二维码图案，加我为朋友。

23级负责人 莫迪茵

ACM是什么

ACM国际大学生程序设计竞赛（英文全称：ACM International Collegiate Programming Contest（简称ACM-ICPC或ICPC））是目前**计算机领域认可度最高的比赛**之一。

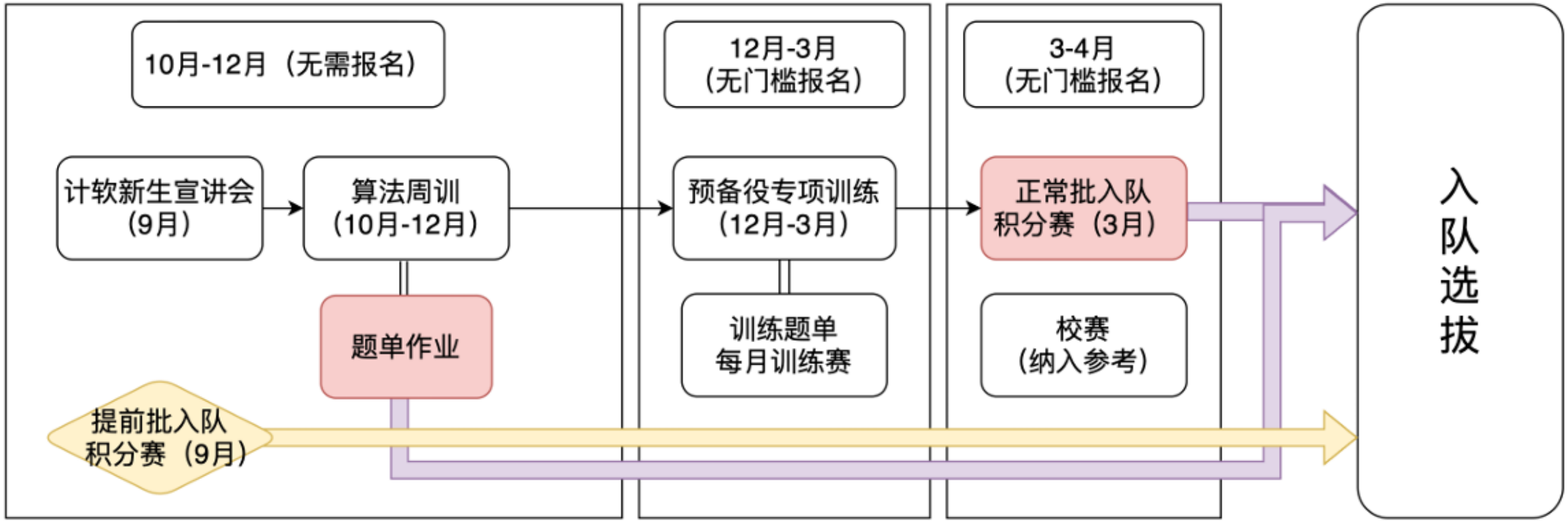
其参赛形式为**团队参赛**，一个队伍由三人组成，目的是在时长五小时的比赛 中通过合作交流、设计算法，通过数据拿到各色气球及奖牌。

简而言之，在最短的时间内以最少的提交错误次数解决编程问题的比赛。

深圳大学 ACM 集训队（SZUACM）是由 JianbinQin 老师带领的实验室团队， 是专为参加 ACM-ICPC 以及 CCPC 等一系列算法比赛而设立的**实验室集训团队**。

入队流程

招新选拔以“实力与态度”为标准，以**积分赛**和**校赛**成绩为主要入队依据，同时参考**训练题单**完成情况，进行招新选拔。



入队流程

- **算法周训**

面向**零基础**，学长学姐带你C和C++入门到提升，从基本语法到简单算法，实现轻松学习。提供**学习路径**，每周有**线下课程**讲解，附有相应的材料供参考，包括博客、视频等，同时每周有对应的**题单**用于巩固知识。

周训的内容都是**基本的数据结构和算法**，对于大家以后学习与找工作都是很有帮助的。

- **预备役训练**（寒训）

采用**无门槛报名制**，将会进行分组并分配学长学姐进行专门指导，内容会涉及一些**复杂算法**，希望大家在渡过新手期后能快速成长，达到可以入队的水平。

周训

- 前半学期(周训)**大班**上课，线下线下同步进行，都有**助教**答疑， 线上使用腾讯会议并录制，课后安排**题单**
- 题单以在线比赛的形式发布，参与比赛时务必将账户的**nickname 改成自己真实姓名**
- 后半学期(周训)**分配导师**，一位导师负责一个小组，周训的授课内容、 课后题单等由导师结合课程内容安排
- 每个月的周训结束后会组织一场针对该月所学知识的**训练赛**.

前置知识——比赛用语

- **AC** Accept的缩写，意为通过
- **WA** Wrong Answer的缩写，意为答案错误
- **RE** Runtime Error的缩写，意为运行错误 （通常是数组越界）
- **CE** Compilation Error的缩写，意为编译错误 （提交错版本等）
- **TLE** Time Limit Exceeded的缩写，意为时间超限
  （死循环或运行时间太长）
- **MLE** Memory Limit Exceeded的缩写，意为内存超限
  （通常是数组开大了）

前置知识——比赛用语

- **榜**：比赛过程中，会**实时展示**各个选手（队伍）的**过题（AC）数以及罚时数**。
- **跟榜**：实际比赛，题目难度乱序，所以经常会根据排行榜，看哪道题过得比较多来决定做那一道题。
- **封榜**：比赛即将结束时，排行榜将只显示其他队伍是否提交题目，而不显示提交的结果，称为封榜。
- **滚榜**：比赛结束后，逐一展示各个队伍的提交结果，队伍的排名在此期间会发生变换，在排行榜上进行滚动。
- **拿气球**：ICPC的传统，比赛过程中每通过（AC）一题，就可以拿到一个题目对应颜色的气球。会插在座位旁边。

The 48th ICPC World Finals

开始时间：2024-09-19 13:51:19 GMT+8

FINISHED

结束时间：2024-09-19 18:51:19 GMT+8

当前时间：05:00:00

Gold Silver Bronze Honorable First to solve problem Solved problem Attempted problem Pending judgement / Frozen

剩余时间：00:00:00

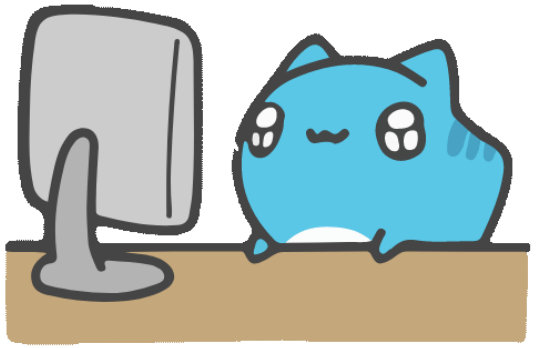
所有队伍 Europe North America Asia Pacific Asia West Northern Eurasia Africa and Arab Latin America Asia East

选项 工具 导出 统计 提交 排名

Place	Team	Solved	Penalty	A 53	B 140	C 128	D 84	E 2	F 135	G 0	H 4	I 100	J 19	K 3	L 16	Dirt	SE
1	Peking University	9	935	+ 3/158	+ 1/26	+ 2/31	+ 1/23	- 4/292	+ 1/43		+ 2/241	+ 1/55	+ 1/99		+ 1/179	30%	0.46
2	Moscow Institute of Physics and Technology	9	1212	+ 2/148	+ 2/57	+ 2/39	+ 1/82	- 4/299	+ 1/43			+ 1/52	+ 2/249	+ 2/184	+ 3/218	43%	0.47
3	Tsinghua University	9	1218	+ 2/108	+ 1/14	+ 2/54	+ 2/92	- 6/294	+ 2/30		- 6/295	+ 2/50	+ 1/244	+ 6/195	+ 4/171	59%	0.47
4	Tokyo Institute of Technology	9	1322	+ 1/124	+ 1/26	+ 2/54	+ 1/133	+ 2/200	+ 1/62	- 2/272		+ 1/81	+ 1/240		+ 5/282	40%	0.47
5	KAIST	8	868	+ 3/81	+ 1/17	+ 1/13	+ 2/155		+ 3/40		- 1/299	+ 1/29	+ 1/254	- 5/291	+ 3/139	46%	0.40
6	National University of Singapore	8	934	+ 2/129	+ 1/15	+ 1/48	+ 1/122		+ 1/35			+ 1/63	+ 1/254	- 2/299	+ 2/228	20%	0.40
7	Beijing Jiaotong University	8	960	+ 1/167	+ 1/25	+ 1/77	+ 1/105		+ 2/40			+ 1/38	+ 1/206	- 1/295	+ 1/282	11%	0.40
8	The University of Tokyo	8	1031	+ 1/111	+ 1/31	+ 3/61	+ 3/120		+ 2/33		- 16/282	+ 1/60	+ 1/278		+ 1/237	38%	0.40
9	Seoul National University	8	1112	+ 2/156	+ 1/36	+ 1/23	+ 1/90		+ 1/61			+ 2/65	+ 4/287	- 5/295	+ 3/254	46%	0.40
10	Zhejiang University	8	1166	+ 1/178	+ 1/29	+ 2/66	+ 1/130		+ 1/107			+ 1/84	+ 1/258		+ 1/294	11%	0.40
11	Massachusetts Institute of Technology	8	1324	+ 3/134	+ 4/23	+ 3/105	+ 1/146	- 7/299	+ 2/72		- 3/292	+ 6/206	+ 1/134		+ 2/224	63%	0.40

前置知识——常用oj

Online judge简称oj，在线判题机。



- 新手入门首推**洛谷**

https://www.luogu.com.cn

- Codeforces，简称**cf**

https://codeforces.com

- Virtual Judge，简称**vj**

https://vjudge.net

- 除上述外，还有AtCoder和AcWing也经常被算法选手提到，大家可以自行探索

前置知识——为什么使用C++？

- 实际比赛中，可以提交**多种语言**的代码（python，java…）
- C/C++具有**更高的运行效率**（同样的代码，使用python实现运行时间可能是C/C++的两倍以上），而题目往往有**运行时间的限制**
- C++封装了许多库（STL……），可直接调用，减少打代码的时间开销，同时支持很多灵活的语法
- **99%**竞赛选手都使用C++，知识分享、队友交流也大多使用的是C++的代码

前置知识——IDE

集成开发环境，用于提供程序开发环境的应用程序

- 常见的IDE有**Dev-C++**、**Visual Studio**、**Visual Studio Code**……

- 新手入门推荐**Dev**，轻量化，支持大多数语法，招新群公告内有下载链接

https://orz-1312128486.cos.ap-nanjing.myqcloud.com/dev-c%2B%2B.zip

- 使用其他IDE也没关系，有问题都可以随时资讯群内的学长学姐

课前小贴士

- 每个语句末尾都**以“;”结束**
- 不要！使用！任何！**中文符号**！！！
- **截屏**，问问题时不要拍照！(snipaste)
- 学会**调试**，受益终生

常用数据类型

- int　　整数，占4个字节，范围-2^31~2^31-1
- long long　　整数，占8个字节，范围-2^63~2^63-1
- float　　浮点数（小数） 4个字节
- double　　浮点数（小数）8个字节
- char　　字符型（a, b, c, 1, 2, 3, ……）

C风格输入输出

- scanf(“____”,_____);
- printf(“____”,_____);
- 常用类型字符

%d / %ld	十进制整数
%o / %O	八进制整数
%x / %X	十六进制整数
%f / %lf	浮点数
%c	字符
%s	字符串

```c
#include<stdio.h>

int main()
{
    int a;
    long long b;

    scanf("%d%ld",&a,&b);

    printf("%d %o %x\n",a,a,a);
    printf("%ld\n",b);

    return 0;
}
```

C++风格输入输出

- cin >> ___ >> ___;
- cout << ___ << ___;
  - 不用声明类型，会根据变量自动分配

```
#include<bits/stdc++.h>
using namespace std;

int main()
{
    int n;
    cin>>n;
    cout<<n<<endl;

    return 0;
}
```

代码框架

- 给大家提供一个简单的框架，方便入门

```
#include<iostream>          //头文件
using namespace std;        //命名空间

int main(){                 //主函数 程序的入口

    return 0;               //返回值
}
```

头文件

- 头文件是扩展名为 .h 的文件，包含了 **C 函数声明和宏定义**，被多个源文件中引用共享。有两种类型的头文件：程序员编写的头文件和编译器自带的头文件（stdio…）。

- 在程序中要使用头文件，需要使用 C 预处理指令 **#include** 来引用它。前面我们已经看过 stdio.h头文件，它是编译器自带的头文件。

- 引用头文件相当于复制头文件的内容，但是我们不会直接在源文件中复制头文件的内容，因为这么做很容易出错，特别在程序是由多个源文件组成的时候。

算术运算符

运算符	描述
+	把两个操作数相加
-	从第一个操作数中减去第二个操作数
*	把两个操作数相乘
/	分子除以分母
%	取模运算符，整除后的余数
++	自增运算符，整数值增加 1
--	自减运算符，整数值减少 1

关系运算符

运算符	描述	实例
==	检查两个操作数的值是否相等，如果相等则条件为真。	(A == B) 为假。
!=	检查两个操作数的值是否相等，如果不相等则条件为真。	(A != B) 为真。
>	检查左操作数的值是否大于右操作数的值，如果是则条件为真。	(A > B) 为假。
<	检查左操作数的值是否小于右操作数的值，如果是则条件为真。	(A < B) 为真。
>=	检查左操作数的值是否大于或等于右操作数的值，如果是则条件为真。	(A >= B) 为假。
<=	检查左操作数的值是否小于或等于右操作数的值，如果是则条件为真。	(A <= B) 为真。

赋值运算符

运算符	描述	实例
=	简单的赋值运算符，把右边操作数的值赋给左边操作数	C = A + B 将把 A + B 的值赋给 C
+=	加且赋值运算符，把右边操作数加上左边操作数的结果赋值给左边操作数	C += A 相当于 C = C + A
-=	减且赋值运算符，把左边操作数减去右边操作数的结果赋值给左边操作数	C -= A 相当于 C = C - A
*=	乘且赋值运算符，把右边操作数乘以左边操作数的结果赋值给左边操作数	C *= A 相当于 C = C * A
/=	除且赋值运算符，把左边操作数除以右边操作数的结果赋值给左边操作数	C /= A 相当于 C = C / A
%=	求模且赋值运算符，求两个操作数的模赋值给左边操作数	C %= A 相当于 C = C % A

ASCII码

- 计算机中**只存储01数字**，那么该**如何储存'a'，'b'，'1'等字符**呢？
- 将数字和字符进行**映射**，比如用数字1表示字符a，存储的时候只需要存数字1的二进制就可以
- 计算机中的对应——ASCII码

48 '0'
65 'A'
97 'a'

P5704 【深基2.例6】字母转换 - 洛谷 | 计算机科学教育新生态 (luogu.com.cn)

转义字符

- 对于 ASCII 编码，0~31（十进制）范围内的字符为控制字符，它们都是看不见的，即**不能在显示器上显示**，甚**至无法从键盘输入**；部分控制字符在编辑语言中还被定义为特殊用途。因此只能用转义字符的形式来表示它们。
- 不过，直接使用 ASCII 码记忆不方便，也不容易理解，所以针对常用的控制字符，各类编程语言对转义字符又定义了**简写方式**。

- \a 响铃
- \n 换行

顺序结构

- 从上到下 逐一执行
- 我们之前见到的都是顺序结构

选择结构

if,

else if,

else

题单！！

- B2005 字符三角形
- P5705 【深基2.例7】数字反转
- P5706 【深基2.例8】再分肥宅水
- P5716 【深基3.例9】月份天数
- P5707 【深基2.例12】上学迟到
- P5717 【深基3.习8】三角形分类
- P1055 [NOIP2008 普及组] ISBN 号码