

# 蓝桥杯全国软件和信息技术专业人才大赛组委会

## 第十一届蓝桥杯全国软件和信息技术专业人才大赛个人赛

### (软件类) **C/C++**和 **Java** 组竞赛规则及说明

#### 1. 组别

竞赛分为: C/C++大学研究生组, C/C++大学 A 组, C/C++大学 B 组, C/C++大学 C 组, Java 大学研究生组, Java 大学 A 组, Java 大学 B 组, Java 大学 C 组共 8 个组别。

每位选手只能申请参加其中一个组别的竞赛。各个组别单独评奖。

研究生只能报研究生组。

一本院校(985、211)本科生只能报大学 A 组以上组别。

其它本科院校本科生可报大学 B 组及以上组别。

其它高职、高专院校可自行选择报任意组别。

#### 2. 竞赛赛程

预赛(省赛)时长: 4 小时。 8 个组别同时进行。

决赛时长: 4 小时。分上下午两个场次(每位选手只参加其中一个场次)。

详细赛程安排以组委会公布信息为准。

#### 3. 竞赛形式

个人赛, 一人一机, 全程机考。

选手机器通过局域网连接到各个赛场的竞赛服务器。

选手答题过程中无法访问互联网, 也不允许使用本机以外的资源(如 USB 连接)。

竞赛系统以“服务器-浏览器”方式发放试题、回收选手答案。

#### 4. 参赛选手机器环境

选手机器配置:

X86 兼容机器, 内存不小于 1G, 硬盘不小于 60G

操作系统: WindowsXP、Windows7 或 Windows10

C/C++语言开发环境:

- Dev-cpp 5.4.0 支持 ANSI C, ANSIC++, STL
- C/C++ API 帮助文档(中文, chm 格式)

Java 语言开发环境:

- JDK 1.6
- Eclipse Helios for JavaSE
- API 帮助文档(中文, chm 格式)

## 5. 试题形式

竞赛题目完全为客观题型。

根据选手所提交答案的测评结果为评分依据。

共有两种题型。

### 5.1. 结果填空题

- 题目描述一个具有确定解的问题。要求选手对问题的解填空。
- 不要求解题过程，不限制解题手段（可以使用任何开发语言或工具，甚至是手工计算），只要求填写最终的结果。
- 最终的解是一个整数或者是一个字符串，最终的解可以使用 ASCII 字符表达。

### 5.2. 编程大题

题目包含明确的问题描述、输入和输出格式，以及用于解释问题的样例数据。

编程大题所涉及的问题一定是有明确客观的标准来判断结果是否正确，并可以通过程序对结果进行评判。

选手应当根据问题描述，编写程序（使用 Python 编写）来解决问题，在评测时选手的程序应当从标准输入读入数据，并将最终的结果输出到标准输出中。

在问题描述中会明确说明给定的条件和限制，明确问题的任务，选手的程序应当能解决在给定条件和限制下的所有可能的情况。

选手的程序应当具有普遍性，不能只适用于题目的样例数据。

为了测试选手给出解法的性能，评分时用的测试用例可能包含大数据量的压力测试用例，选手选择算法时要尽可能考虑可行性和效率问题。

## 6. 试题考查范围

试题考查选手解决实际问题的能力，对于结果填空题，选手可以使用手算、软件、编程等方法解决，对于编程大题，选手只能编程解决。

竞赛侧重考查选手对于算法和数据结构的灵活运用能力，很多试题需要使用计算机算法才能有效的解决。

考查范围包括：以下范围中标\*的部分只限于 C/C++大学研究生组、C/C++大学 A 组、Java 大学研究生组、Java 大学 A 组。

**C/C++程序设计基础：**包含使用 C/C++编写程序的能力。该部分不考查选手对某一语法的理解程度，选手可以使用自己喜欢的语句编写程序。选手可在 C 语言程序中使用标准 C 的库函数，在 C++语言程序中使用标准 C++的库函数（包括 C 库、STL 等）。

**Java 程序设计基础：**包含使用 Java 编写程序的能力。该部分不考查选手对某一语法的理解程度，选手可以使用自己喜欢的语句编写程序。选手可在程序中使用 JDK 中自带的类，但不能使用其它的第三方类。

**计算机算法：**枚举、排序、搜索、计数、贪心、动态规划、图论、数论、博弈论\*、概率论\*、计算几何\*、字符串算法等。

**数据结构：**数组、对象/结构、字符串、队列、栈、树、图、堆、平衡树/线段树、复杂数据结构\*、嵌套数据结构\*等。

## 7. 答案提交

选手只有在比赛时间内提交的答案内容是可以用来评测的，比赛之后的任何提交均无

效。

选手应使用考试指定的网页来提交代码，任何其他方式的提交（如邮件、U 盘）都不作为评测依据。

选手可在比赛中的任何时间查看自己之前提交的代码，也可以重新提交任何题目的答案，对于每个试题，仅有最后的一次提交被保存并作为评测的依据。在比赛中，评测结果不会显示给选手，选手应当在没有反馈的情况下自行设计数据调试自己的程序。

对于每个试题，选手应将试题的答案内容拷贝粘贴到网页上进行提交。

程序中应只包含计算模块，不要包含任何其他模块，比如图形、系统接口调用、系统中断等。对于系统接口的调用都应通过标准库来进行。

程序中引用的库应该在程序中以源代码的方式写出，在提交时也应当和程序的其他部分一起提交。

## 8. 评分

全部使用机器自动评分。

对于结果填空题，题目保证只有唯一解，选手的结果只有和解完全相同才得分，出现格式错误或有多余内容时不得分。

对于编程大题，评测系统将使用多个评测数据来测试程序。每个评测数据有对应的分数。选手所提交的程序将分别用每个评测数据作为输入来运行。对于某个评测数据，如果选手程序的输出与正确答案是匹配的，则选手获得该评测数据的分数。

评测使用的评测数据一般与试题中给定的样例输入输出不一样。因此建议选手在提交程序前使用不同的数据测试自己的程序。

提交的程序应严格按照输出格式的要求来输出，包括输出空格和换行的要求。如果程序没有遵循输出格式的要求将被判定为答案错误。请注意，程序在输出的时候多输出了内容也属于没有遵循输出格式要求的一种，所以在输出的时候请不要输出任何多余的内容，比如调试输出。

C/C++选手请务必选择正确的编译器，如果编译器选手错误，可能导致编译不通过而得 0 分。C/C++选手请务必让主函数的返回值为 0，当返回非 0 时会认为程序执行错误而得 0 分。

Java 选手请务必不要使用 `package` 语句，并确保自己的主类名称为 `Main`，否则会导致评测系统运行时找不到主类而得 0 分。

## 9. 样题

样题 1：矩形切割（结果填空题）

### 【问题描述】

小明有一些矩形的材料，他要从这些矩形材料中切割出一些正方形。

当他面对一块矩形材料时，他总是从中间切割一刀，切出一块最大的正方形，剩下一块矩形，然后再切割剩下的矩形材料，直到全部切为正方形为止。

例如，对于一块两边分别为 5 和 3 的材料（记为  $5 \times 3$ ），小明会依次切出  $3 \times 3$ 、 $2 \times 2$ 、 $1 \times 1$ 、 $1 \times 1$  共 4 个正方形。

现在小明有一块矩形的材料，两边长分别是 2019 和 324。请问小明最终会切出多少个正方形？

### 【答案提交】

这是一道结果填空题，你只需要算出结果后提交即可。本题的结果为一个整数，在提交答案时只填写这个整数，填写多余的内容将无法得分。

说明：以上是问题描述的部分，选手做题时可以直接手算答案，即按照题目意思一步一步切割，最后得到切出的矩形个数，手算可能花费一些时间。如果选手在手算时使用除法等方式加快速度，时间可能少一些。如果选手编写程序来计算，可以减少手算中出现的失误。

本题答案为：21

样题 2：特别数的和（编程大题）

**【问题描述】**

小明对数位中含有 2、0、1、9 的数字很感兴趣（不包括前导 0），在 1 到 40 中这样的数包括 1、2、9、10 至 32、39 和 40，共 28 个，他们的和是 574。

请问，在 1 到  $n$  中，所有这样的数的和是多少？

**【输入格式】**

输入一行包含一个整数  $n$ 。

**【输出格式】**

输出一行，包含一个整数，表示满足条件的数的和。

**【样例输入】**

40

**【样例输出】**

574

**【评测用例规模与约定】**

对于 20% 的评测用例， $1 \leq n \leq 10$ 。

对于 50% 的评测用例， $1 \leq n \leq 100$ 。

对于 80% 的评测用例， $1 \leq n \leq 1000$ 。

对于所有评测用例， $1 \leq n \leq 10000$ 。

说明：本题是一道编程题，选手需要编写一个程序来解决问题。以下给出一个参考程序，选手所编写的其他程序只要能给出正确的结果即可得分。

以下是 C/C++ 语言的参考代码（此代码对于 C 和 C++ 两种语言通用，对于 C++ 语言，一般会将前面的 `#include` 语句改为对应的 C++ 库）

```
#include <stdio.h>
```

```
int main()
```

```
{
```

```
    int n, i, ans = 0;
```

```
    scanf("%d", &n);
```

```
    for (i = 1; i <= n; ++i)
```

```
    {
```

```
        int t = i, ok = 0;
```

```
        while (t > 0)
```

```
        {
```

```
            int g = t % 10;
```

```
            if (g == 2 || g == 0 || g == 1 || g == 9)
```

```
                ok = 1;
```

```
            t = t / 10;
```

```
        }
```

```
        if (ok)
```

```

        ans += i;
    }
    printf("%d\n", ans);
    return 0;
}

```

以下是 Java 语言的参考代码

```

import java.util.*;

public class Main
{
    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt(), ans = 0;
        for (int i = 1; i <= n; ++i)
        {
            int t = i;
            boolean ok = false;
            while (t > 0)
            {
                int g = t % 10;
                if (g==2 || g==0 || g==1 || g==9)
                    ok = true;
                t = t / 10;
            }
            if (ok)
                ans += i;
        }
        System.out.println(ans);
    }
}

```

## 10. 其它注意事项

(1) 选手必须符合参赛资格，不得弄虚作假。资格审查中一旦发现问题，则取消其报名资格；竞赛过程中发现问题，则取消竞赛资格；竞赛后发现问题，则取消竞赛成绩，收回获奖证书及奖品等，并在大赛官网上公示。

(2) 参赛选手应遵守竞赛规则，遵守赛场纪律，服从大赛组委会的指挥和安排，爱护竞赛赛场地的设备。

(3) 竞赛采用机器阅卷+少量人工辅助。选手需要特别注意提交答案的形式。必须仔细阅读题目的输入、输出要求以及示例，不要随意添加不需要的内容。

(4) 使用 Java 语言时，注意主类名必须是：**Main**，不能使用 **package** 语句。使用 C/C++ 语言时，注意主函数需要 **return 0**；

(5) **C** 组与 **C++** 组选手提交答案时，一定要注意选择 **C** 或 **C++**（即编译器类型）。因为使用机器阅卷，很可能会因为选手选择了错误的编译器，而使自己代码无法编译通过。