

第七届蓝桥杯大赛个人赛省赛（软件类）

C/C++ 大学 A 组

考生须知：

- 考试开始后，选手首先下载题目，并使用考场现场公布的解压密码解压试题。
 - 考试时间为 4 小时，时间截止后，提交答案无效。
 - 在考试强制结束前，选手可以主动结束考试（需要身份验证），结束考试后将无法继续提交或浏览答案。
 - 选手可浏览自己已经提交的答案，被浏览的答案允许拷贝。
 - 对同一题目，选手可多次提交答案，以最后一次提交的答案为准。
 - 选手切勿在提交的代码中书写“姓名”、“考号”、“院校名”等与身份有关的信息或其它与竞赛题目无关的内容，否则成绩无效。
 - 选手必须通过浏览器方式提交自己的答案，选手在其它位置的作答或其他方式提交的答案无效。
 - 试题包含三种类型：“结果填空”、“代码填空”与“程序设计”。
- 结果填空题：**要求选手根据题目描述直接填写结果，**求解方式不限**，不要求源代码，把结果填空的答案直接通过网页提交即可，不要书写多余的内容。
- 代码填空题：**要求选手在弄清给定代码工作原理的基础上填写缺失的部分，使得程序逻辑正确、完整。
- 把代码填空的答案（仅填空处的答案，**不包括题目已存在的代码或符号**）直接通过网页提交即可，不要书写多余的内容。
- 使用 ANSI C/ANSI C++ 标准，不要依赖操作系统或编译器提供的特殊函数。
- 程序设计题目：**要求选手设计的程序对于给定的输入能给出正确的输出结果，**考生的程序只有能运行出正确结果才有机会得分**。
- 注意：在评卷时使用的输入数据与试卷中给出的示例数据可能是不同的，**选手的程序必须是通用的**，不能只对试卷中给定的数据有效。
- 对于编程题目，要求选手给出的解答完全符合 ANSI C++ 标准，不能使用诸如绘图、Win32 API、中断调用、硬件操作或与操作系统相关的 API。
- 代码中允许使用 STL 类库。
- 注意：main 函数结束必须返回 0**
- 注意：所有依赖的函数必须明确地在源文件中 `#include <xxx>`，不能通过工程设置而省略常用头文件。
- 所有源码必须在同一文件中，调试通过后，拷贝提交。
- 提交时，注意选择所期望的编译器类型。

1. 结果填空（满分 3 分）

问题的描述在考生文件夹下对应题号的“题目.txt”中，相关的参考文件在同一目录中，请先阅读题目，不限解决问题的方式，只要求提交结果。

必须通过浏览器提交答案。

2. 结果填空（满分 5 分）

问题的描述在考生文件夹下对应题号的“题目.txt”中。相关的参考文件在同一目录中。请先阅读题目，不限解决问题的方式，只要求提交结果。
必须通过浏览器提交答案。

3. 结果填空 (满分 11 分)

问题的描述在考生文件夹下对应题号的“题目.txt”中。相关的参考文件在同一目录中。请先阅读题目，不限解决问题的方式，只要求提交结果。
必须通过浏览器提交答案。

4. 代码填空 (满分 9 分)

问题的描述在考生文件夹下对应题号的“题目.txt”中。相关的参考文件在同一目录中。请先阅读题目，不限解决问题的方式。
只要求填写缺失的代码部分，千万不要画蛇添足，填写多余的已有代码或符号。
必须通过浏览器提交答案。

5. 代码填空 (满分 13 分)

问题的描述在考生文件夹下对应题号的“题目.txt”中。相关的参考文件在同一目录中。请先阅读题目，不限解决问题的方式。
只要求填写缺失的代码部分，千万不要画蛇添足，填写多余的已有代码或符号。
必须通过浏览器提交答案。

6. 结果填空 (满分 15 分)

问题的描述在考生文件夹下对应题号的“题目.txt”中。相关的参考文件在同一目录中。请先阅读题目，不限解决问题的方式，只要求提交结果。
必须通过浏览器提交答案。

7. 结果填空 (满分 19 分)

问题的描述在考生文件夹下对应题号的“题目.txt”中。相关的参考文件在同一目录中。请先阅读题目，不限解决问题的方式，只要求提交结果。
必须通过浏览器提交答案。

8. 程序设计 (满分 21 分)

问题的描述在考生文件夹下对应题号的“题目.txt”中。相关的参考文件在同一目录中。请先阅读题目，必须通过编程的方式解决问题。

注意：在评卷时使用的输入数据与试卷中给出的示例数据可能是不同的。**选手的程序必须是通用的**，不能只对试卷中给定的数据有效。

仔细阅读程序的输入、输出要求，千万不要输出没有要求的、多余的内容，例如：“请

您输入 xx 数据: ”。

建议仔细阅读示例，不要想当然!

程序处理完一个用例的数据后，立即退出 (return 0)，千万不要循环等待下一个用例的输入。

程序必须使用标准输入、标准输出，以便于机器评卷时重定向。

对于编程题目，要求选手给出的解答完全符合 ANSI C++ 标准，不能使用诸如绘图、Win32API、中断调用、硬件操作或与操作系统相关的 API。

代码中允许使用 STL 类库。

注意: main 函数结尾需要 return 0

注意: 所有依赖的函数必须明确地在源文件中 #include <xxx>， 不能通过工程设置而省略常用头文件。

所有代码放在同一个源文件中，调试通过后，拷贝提交该源码。

提交时，注意选择所期望的编译器类型。

9. 程序设计 (满分 25 分)

问题的描述在考生文件夹下对应题号的“题目.txt”中。相关的参考文件在同一目录中。请先阅读题目，必须通过编程的方式解决问题。

注意: 在评卷时使用的输入数据与试卷中给出的示例数据可能是不同的。**选手的程序必须是通用的**，不能只对试卷中给定的数据有效。

仔细阅读程序的输入、输出要求，千万不要输出没有要求的、多余的内容，例如: “请您输入 xx 数据: ”。

建议仔细阅读示例，不要想当然!

程序处理完一个用例的数据后，立即退出 (return 0)，千万不要循环等待下一个用例的输入。

程序必须使用标准输入、标准输出，以便于机器评卷时重定向。

对于编程题目，要求选手给出的解答完全符合 ANSI C++ 标准，不能使用诸如绘图、Win32API、中断调用、硬件操作或与操作系统相关的 API。

代码中允许使用 STL 类库。

注意: main 函数结尾需要 return 0

注意: 所有依赖的函数必须明确地在源文件中 #include <xxx>， 不能通过工程设置而省略常用头文件。

所有代码放在同一个源文件中，调试通过后，拷贝提交该源码。

提交时，注意选择所期望的编译器类型。

10. 程序设计 (满分 29 分)

问题的描述在考生文件夹下对应题号的“题目.txt”中。相关的参考文件在同一目录中。请先阅读题目，必须通过编程的方式解决问题。

注意: 在评卷时使用的输入数据与试卷中给出的示例数据可能是不同的。**选手的程序必须是通用的**，不能只对试卷中给定的数据有效。

仔细阅读程序的输入、输出要求，千万不要输出没有要求的、多余的内容，例如: “请您输入 xx 数据: ”。

建议仔细阅读示例，不要想当然!

程序处理完一个用例的数据后，立即退出（return 0），千万不要循环等待下一个用例的输入。

程序必须使用标准输入、标准输出，以便于机器评卷时重定向。

对于编程题目，要求选手给出的解答完全符合 ANSI C++ 标准，不能使用诸如绘图、Win32 API、中断调用、硬件操作或与操作系统相关的 API。

代码中允许使用 STL 类库。

注意：main 函数结尾需要 return 0

注意：所有依赖的函数必须明确地在源文件中 #include <xxx>， 不能通过工程设置而省略常用头文件。

所有代码放在同一个源文件中，调试通过后，拷贝提交该源码。

提交时，注意选择所期望的编译器类型。

网友年龄

某君新认识一网友。

当问及年龄时，他的网友说：

“我的年龄是个 2 位数，我比儿子大 27 岁，

如果把我的年龄的两位数字交换位置，刚好就是我儿子的年龄”

请你计算：网友的年龄一共有多少种可能情况？

提示：30 岁就是其中一种可能哦。

请填写表示可能情况的种数。

注意：你提交的应该是一个整数，不要填写任何多余的内容或说明性文字。

生日蜡烛

某君从某年开始每年都举办一次生日 party，并且每次都要吹熄与年龄相同根数的蜡烛。

现在算起来，他一共吹熄了 236 根蜡烛。

请问，他从多少岁开始过生日 party 的？

请填写他开始过生日 party 的年龄数。

注意：你提交的应该是一个整数，不要填写任何多余的内容或说明性文字。

方格填数

如下的 10 个格子

```

+--+--+--+
|  |  |  |
+--+--+--+
|  |  |  |
+--+--+--+
|  |  |  |
+--+--+--+

```

(如果显示有问题，也可以参看【图 1.jpg】)

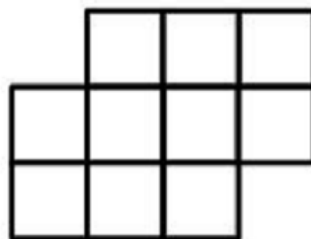
填入 0~9 的数字，要求：连续的两个数字不能相邻。
(左右、上下、对角都算相邻)

一共有多少种可能的填数方案？

请填写表示方案数目的整数。

注意：你提交的应该是一个整数，不要填写任何多余的内容或说明性文字。

0~9的数字填入，连续数字不相邻



对角也算相邻

快速排序

排序在各种场合经常被用到。
快速排序是十分常用的高效率的算法。

其思想是：先选一个“标尺”，
用它把整个队列过一遍筛子，
以保证：其左边的元素都不大于它，其右边的元素都不小于它。

这样，排序问题就被分割为两个子区间。
再分别对子区间排序就可以了。

下面的代码是一种实现，请分析并填写划线部分缺少的代码。

```
#include <stdio.h>

void swap(int a[], int i, int j)
{
    int t = a[i];
    a[i] = a[j];
    a[j] = t;
}

int partition(int a[], int p, int r)
{
    int i = p;
    int j = r + 1;
    int x = a[p];
    while(1){
        while(i<r && a[++i]<x);
        while(a[--j]>x);
        if(i>=j) break;
        swap(a,i,j);
    }
    _____;
    return j;
}

void quicksort(int a[], int p, int r)
{
    if(p<r){
        int q = partition(a,p,r);
        quicksort(a,p,q-1);
        quicksort(a,q+1,r);
    }
}

int main()
{
    int i;
    int a[] = {5,13,6,24,2,8,19,27,6,12,1,17};
    int N = 12;
```

注意：只填写缺少的内容，不要书写任何题面已有代码或说明性文字。

下面的代码把一个整数的二进制表示的最右边的连续的 1 全部变成 0
如果最后一位是 0，则原数字保持不变。

请仔细阅读程序，填写划线部分缺少的代码。

第 7/14 页

}

注意：只填写缺少的内容，不要书写任何题面已有代码或说明性文字。

寒假作业

现在小学的数学题目也不是那么好玩的。
看看这个寒假作业：

$$\begin{aligned}\square + \square &= \square \\ \square - \square &= \square \\ \square \times \square &= \square \\ \square \div \square &= \square\end{aligned}$$

(如果显示不出来，可以参见【图 1.jpg】)

每个方块代表 1~13 中的某一个数字，但不能重复。
比如：

$$\begin{aligned}6 + 7 &= 13 \\ 9 - 8 &= 1 \\ 3 * 4 &= 12 \\ 10 / 2 &= 5\end{aligned}$$

以及：

$$\begin{aligned}7 + 6 &= 13 \\ 9 - 8 &= 1 \\ 3 * 4 &= 12 \\ 10 / 2 &= 5\end{aligned}$$

就算两种解法。（加法，乘法交换律后算不同的方案）

你一共找到了多少种方案？

请填写表示方案数目的整数。

注意：你提交的应该是一个整数，不要填写任何多余的内容或说明性文字。

$$\begin{array}{l} \square + \square = \square \\ \square - \square = \square \\ \square \times \square = \square \\ \square \div \square = \square \end{array}$$

剪邮票

如【图 1.jpg】，有 12 张连在一起的 12 生肖的邮票。
现在你要从中剪下 5 张来，要求必须是连着的。
(仅仅连接一个角不算相连)
比如，【图 2.jpg】，【图 3.jpg】中，粉红色所示部分就是合格的剪取。

请你计算，一共有多少种不同的剪取方法。

请填写表示方案数目的整数。

注意：你提交的应该是一个整数，不要填写任何多余的内容或说明性文字。

△

1	2	3	4
5	6	7	8
9	10	11	12

△

△

1	2	3	4
5	6	7	8
9	10	11	12

△

△

1	2	3	4
5	6	7	8
9	10	11	12

△

四平方和

四平方和定理，又称为拉格朗日定理：

每个正整数都可以表示为至多 4 个正整数的平方和。

如果把 0 包括进去，就正好可以表示为 4 个数的平方和。

比如：

$$5 = 0^2 + 0^2 + 1^2 + 2^2$$

$$7 = 1^2 + 1^2 + 1^2 + 2^2$$

(^符号表示乘方的意思)

对于一个给定的正整数，可能存在多种平方和的表示法。

要求你对 4 个数排序：

$0 \leq a \leq b \leq c \leq d$

并对所有的可能表示法按 a, b, c, d 为联合主键升序排列，最后输出第一个表示法

程序输入为一个正整数 N ($N < 5000000$)

要求输出 4 个非负整数，按从小到大排序，中间用空格分开

例如，输入：

5

则程序应该输出：

0 0 1 2

再例如，输入：

12

则程序应该输出：

0 2 2 2

再例如，输入：

773535

则程序应该输出：

1 1 267 838

资源约定：

峰值内存消耗 $< 256M$

CPU 消耗 $< 3000ms$

请严格按照要求输出，不要画蛇添足地打印类似：“请输入...” 的多余内容。

所有代码放在同一个源文件中，调试通过后，拷贝提交该源码。

注意：main 函数需要返回 0

注意：只使用 ANSI C/ANSI C++ 标准，不要调用依赖于编译环境或操作系统的特殊函数。

注意：所有依赖的函数必须明确地在源文件中 `#include <xxx>`，不能通过工程设置而省略常用头文件。

提交时，注意选择所期望的编译器类型。

密码脱落

X 星球的考古学家发现了一批古代留下来的密码。

这些密码是由 A、B、C、D 四种植物的种子串成的序列。

仔细分析发现，这些密码串当初应该是前后对称的（也就是我们说的镜像串）。

由于年代久远，其中许多种子脱落了，因而可能会失去镜像的特征。

你的任务是：

给定一个现在看到的密码串，计算一下从当初的状态，它要至少脱落多少个种子，才可能会变成现在的样子。

输入一行，表示现在看到的密码串（长度不大于 1000）

要求输出一个正整数，表示至少脱落了多少个种子。

例如，输入：

ABCBA

则程序应该输出：

0

再例如，输入：

ABDCDCBABC

则程序应该输出：

3

资源约定：

峰值内存消耗 < 256M

CPU 消耗 < 1000ms

请严格按照要求输出，不要画蛇添足地打印类似：“请输入...” 的多余内容。

所有代码放在同一个源文件中，调试通过后，拷贝提交该源码。

注意：main 函数需要返回 0

注意：只使用 ANSI C/ANSI C++ 标准，不要调用依赖于编译环境或操作系统的特殊函数。

注意：所有依赖的函数必须明确地在源文件中 `#include <xxx>`， 不能通过工程设置而省略常用头文件。

提交时，注意选择所期望的编译器类型。

最大比例

X 星球的某个大奖赛设了 M 级奖励。每个级别的奖金是一个正整数。
并且，相邻的两个级别间的比例是个固定值。
也就是说：所有级别的奖金数构成了一个等比数列。比如：
16,24,36,54
其等比值为：3/2

现在，我们随机调查了一些获奖者的奖金数。
请你据此推算可能的最大的等比值。

输入格式：

第一行为数字 N ($0 < N < 100$)，表示接下的一行包含 N 个正整数

第二行 N 个正整数 X_i ($X_i < 1\,000\,000\,000\,000$)，用空格分开。每个整数表示调查到的某人的奖金数额

要求输出：

一个形如 A/B 的分数，要求 A、B 互质。表示可能的最大比例系数

测试数据保证了输入格式正确，并且最大比例是存在的。

例如，输入：

3
1250 200 32

程序应该输出：

25/4

再例如，输入：

4
3125 32 32 200

程序应该输出：

5/2

再例如，输入：

3
549755813888 524288 2

程序应该输出：

4/1

资源约定：

峰值内存消耗 < 256M

CPU 消耗 < 3000ms

请严格按照要求输出，不要画蛇添足地打印类似：“请您输入...” 的多余内容。

所有代码放在同一个源文件中，调试通过后，拷贝提交该源码。

注意: `main` 函数需要返回 0

注意: 只使用 ANSI C/ANSI C++ 标准，不要调用依赖于编译环境或操作系统的特殊函数。

注意: 所有依赖的函数必须明确地在源文件中 `#include <xxx>`， 不能通过工程设置而省略常用头文件。

提交时，注意选择所期望的编译器类型。