

第八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 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。

2. 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。

3. 结果填空 (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，只要求提交结果。
必须通过浏览器提交答案。

4. 结果填空 (满分 17 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。

5. 代码填空 (满分 7 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. 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。

7. 程序设计 (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，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。

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。

注意事项同上题

9. 程序设计 (满分 23 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。

注意事项同上题

10. 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。

注意事项同上题

标题: 迷宫

X 星球的一处迷宫游乐场建在某个小山坡上。
它是由 10x10 相互连通的小房间组成的。

房间的地板上写着一个很大的字母。
我们假设玩家是面朝上坡的方向站立, 则:
L 表示走到左边的房间,
R 表示走到右边的房间,
U 表示走到上坡方向的房间,
D 表示走到下坡方向的房间。

X 星球的居民有点懒, 不愿意费力思考。
他们更喜欢玩运气类的游戏。这个游戏也是如此!

开始的时候, 直升机把 100 名玩家放入一个个小房间内。
玩家一定要按照地上的字母移动。

迷宫地图如下：

```
-----  
UDDLUULRUL  
UURLLLRRRU  
RRUURLDLRD  
RUDDDDUUUU  
URUDLLRRUU  
DURLRLDLRL  
ULLURLLRDU  
RDLULLRDDD  
UDDUDUDULL  
ULRDLUURRR  
-----
```

请你计算一下，最后，有多少玩家会走出迷宫？
而不是在里边兜圈子。

请提交该整数，表示走出迷宫的玩家数目，不要填写任何多余的内容。

如果你还没明白游戏规则，可以参看一个简化的 4x4 迷宫的解说图：
p1.png



标题：跳蚱蜢

如图 p1.png 所示：

有 9 只盘子，排成 1 个圆圈。

其中 8 只盘子内装着 8 只蚱蜢，有一个是空盘。

我们把这些蚱蜢顺时针编号为 1~8

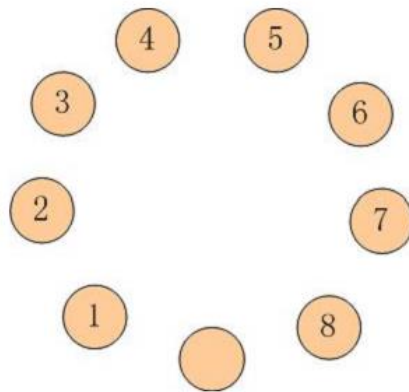
每只蚱蜢都可以跳到相邻的空盘中，

也可以再用点力，越过一个相邻的蚱蜢跳到空盘中。

请你计算一下，如果要使得蚱蜢们的队形改为按照逆时针排列，

并且保持空盘的位置不变（也就是 1-8 换位，2-7 换位,...），至少要经过多少次跳跃？

注意：要求提交的是一个整数，请不要填写任何多余内容或说明文字。



标题：魔方状态

二阶魔方就是只有 2 层的魔方，只由 8 个小块组成。

如图 p1.png 所示。

小明很淘气，他只喜欢 3 种颜色，所有把家里的二阶魔方重新涂了颜色，如下：

前面：橙色

右面：绿色

上面：黄色

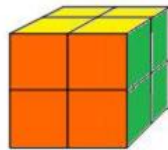
左面：绿色

下面：橙色
后面：黄色

请你计算一下，这样的魔方被打乱后，一共有多少种不同的状态。

如果两个状态经过魔方的整体旋转后，各个面的颜色都一致，则认为是同一状态。

请提交表示状态数的整数，不要填写任何多余内容或说明文字。



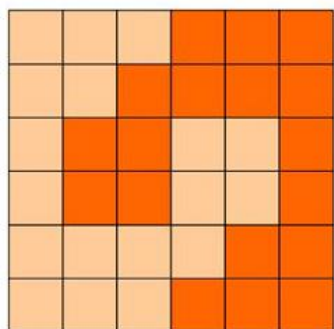
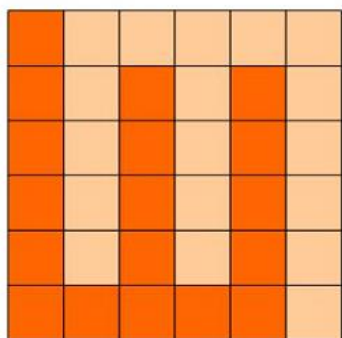
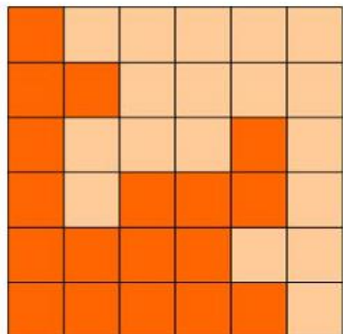
标题：方格分割

6x6 的方格，沿着格子的边线剪开成两部分。
要求这两部分的形状完全相同。

如图：p1.png, p2.png, p3.png 就是可行的分割法。

试计算：
包括这 3 种分法在内，一共有多少种不同的分割方法。
注意：旋转对称的属于同一种分割法。

请提交该整数，不要填写任何多余的内容或说明文字。



标题：字母组串

由 A,B,C 这 3 个字母就可以组成许多串。
比如: "A","AB","ABC","ABA","AACBB"

现在, 小明正在思考一个问题:
如果每个字母的个数有限定, 能组成多少个已知长度的串呢?

他请好朋友来帮忙, 很快得到了代码,
解决方案超级简单, 然而最重要的部分却语焉不详。

请仔细分析源码, 填写划线部分缺少的内容。

```
#include <stdio.h>

// a 个 A, b 个 B, c 个 C 字母, 能组成多少个不同的长度为 n 的串。
int f(int a, int b, int c, int n)
{
    if(a<0 || b<0 || c<0) return 0;
    if(n==0) return 1;

    return _____; // 填空
}

int main()
{
    printf("%d\n", f(1,1,1,2));
    printf("%d\n", f(1,2,3,3));
    return 0;
}
```

对于上面的测试数据, 小明口算的结果应该是:

6
19

注意: 只填写划线部分缺少的代码, 不要提交任何多余内容或说明性文字。

标题：最大公共子串

最大公共子串长度问题就是：

求两个串的所有子串中能够匹配上的最大长度是多少。

比如："abcdkkk" 和 "baabcdadabc",

可以找到的最长的公共子串是"abcd",所以最大公共子串长度为 4。

下面的程序是采用矩阵法进行求解的，这对串的规模不大的情况还是比较有效的解法。

请分析该解法的思路，并补全划线部分缺失的代码。

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

#define N 256
int f(const char* s1, const char* s2)
{
    int a[N][N];
    int len1 = strlen(s1);
    int len2 = strlen(s2);
    int i,j;

    memset(a,0,sizeof(int)*N*N);
    int max = 0;
    for(i=1; i<=len1; i++){
        for(j=1; j<=len2; j++){
            if(s1[i-1]==s2[j-1]) {
                a[i][j] = _____; //填空
                if(a[i][j] > max) max = a[i][j];
            }
        }
    }

    return max;
}

int main()
{
    printf("%d\n", f("abcdkkk", "baabcdadabc"));
    return 0;
}
```

```
}
```

注意：只提交缺少的代码，不要提交已有的代码和符号，也不要提交说明性文字。

描述：正则问题

考虑一种简单的正则表达式：

只由 `x () |` 组成的正则表达式。

小明想求出这个正则表达式能接受的最长字符串的长度。

例如 `((xx|xxx)x|(x|xx))xx` 能接受的最长字符串是： `xxxxxx`，长度是 6。

输入

一个由 `x()`组成的正则表达式。输入长度不超过 100，保证合法。

输出

这个正则表达式能接受的最长字符串的长度。

例如，

输入：

`((xx|xxx)x|(x|xx))xx`

程序应该输出：

6

资源约定：

峰值内存消耗（含虚拟机） < 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：“请您输入...” 的多余内容。

注意：

`main` 函数需要返回 0；

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。

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。

提交程序时，注意选择所期望的语言类型和编译器类型。

标题：包子凑数

小明几乎每天早晨都会在一家包子铺吃早餐。他发现这家包子铺有 N 种蒸笼，其中第 i 种蒸笼恰好能放 A_i 个包子。每种蒸笼都有非常多笼，可以认为是无限笼。

每当有顾客想买 X 个包子，卖包子的大叔就会迅速选出若干笼包子来，使得这若干笼中恰好一共有 X 个包子。比如一共有 3 种蒸笼，分别能放 3、4 和 5 个包子。当顾客想买 11 个包子时，大叔就会选 2 笼 3 个的再加 1 笼 5 个的（也可能选出 1 笼 3 个的再加 2 笼 4 个的）。

当然有时包子大叔无论如何也凑不出顾客想买的数量。比如一共有 3 种蒸笼，分别能放 4、5 和 6 个包子。而顾客想买 7 个包子时，大叔就凑不出来了。

小明想知道一共有多少种数目是包子大叔凑不出来的。

输入

第一行包含一个整数 N ，($1 \leq N \leq 100$)

以下 N 行每行包含一个整数 A_i ，($1 \leq A_i \leq 100$)

输出

一个整数代表答案。如果凑不出的数目有无限多个，输出 INF。

例如，

输入：

2

4

5

程序应该输出：

6

再例如，

输入：

2

4

6

程序应该输出：

INF

样例解释：

对于样例 1，凑不出的数目包括：1, 2, 3, 6, 7, 11。

对于样例 2，所有奇数都凑不出来，所以有无限多个。

资源约定：

峰值内存消耗（含虚拟机） < 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：“请您输入...” 的多余内容。

注意：

main 函数需要返回 0；

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。

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。

提交程序时，注意选择所期望的语言类型和编译器类型。

标题： 分巧克力

儿童节那天有 K 位小朋友到小明家做客，小明拿出了珍藏的巧克力招待小朋友们。小明一共有 N 块巧克力，其中第 i 块是 $H_i \times W_i$ 的方格组成的长方形。

为了公平起见，小明需要从这 N 块巧克力中切出 K 块巧克力分给小朋友们。切出的巧克力需要满足：

1. 形状是正方形，边长是整数
2. 大小相同

例如一块 6×5 的巧克力可以切出 6 块 2×2 的巧克力或者 2 块 3×3 的巧克力。

当然小朋友们都希望得到的巧克力尽可能大，你能帮小 Hi 计算出最大的边长是多少么？

输入

第一行包含两个整数 N 和 K 。 ($1 \leq N, K \leq 100000$)

以下 N 行每行包含两个整数 H_i 和 W_i 。 ($1 \leq H_i, W_i \leq 100000$)

输入保证每位小朋友至少能获得一块 1×1 的巧克力。

输出

输出切出的正方形巧克力最大可能的边长。

样例输入:

2 10

6 5

5 6

样例输出:

2

资源约定:

峰值内存消耗 (含虚拟机) < 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: “请您输入...” 的多余内容.

注意:

main 函数需要返回 0;

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.

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.

提交程序时, 注意选择所期望的语言类型和编译器类型.

标题: 油漆面积

X 星球的一批考古机器人正在一片废墟上考古.

该区域的地面坚硬如石、平整如镜.

管理人员为方便, 建立了标准的直角坐标系.

每个机器人都各有特长、身怀绝技. 它们感兴趣的内容也不相同.

经过各种测量, 每个机器人都会报告一个或多个矩形区域, 作为优先考古的区域.

矩形的表示格式为(x1,y1,x2,y2), 代表矩形的两个对角点坐标.

为了醒目, 总部要求对所有机器人选中的矩形区域涂黄色油漆.

小明并不需要当油漆工, 只是他需要计算一下, 一共要耗费多少油漆.

其实这也不难, 只要算出所有矩形覆盖的区域一共有多大面积就可以了.

注意, 各个矩形间可能重叠.

本题的输入为若干矩形, 要求输出其覆盖的总面积.

输入格式:

第一行, 一个整数 n , 表示有多少个矩形($1 \leq n < 10000$)

接下来的 n 行, 每行有 4 个整数 $x1\ y1\ x2\ y2$, 空格分开, 表示矩形的两个对角顶点坐标.
($0 \leq x1, y1, x2, y2 \leq 10000$)

输出格式:

一行一个整数, 表示矩形覆盖的总面积。

例如,

输入:

3

1 5 10 10

3 1 20 20

2 7 15 17

程序应该输出:

340

再例如,

输入:

3

5 2 10 6

2 7 12 10

8 1 15 15

程序应该输出:

128

资源约定:

峰值内存消耗 (含虚拟机) $< 256\text{M}$

CPU 消耗 $< 2000\text{ms}$

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。

注意:

main 函数需要返回 0;

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。

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。

提交程序时, 注意选择所期望的语言类型和编译器类型。