

目的：检查飞行器的飞行计划会不会进入禁飞区

过程：1. 禁飞区的信息存在 file 中（需要 load 到程序）

2. 飞行计划在另一个 file 中（要 load）

3. 两个文件名被指定为命令行参数（首先是禁飞区域文件名，然后是飞行计划文件名）

4. 检查检查飞行计划，以确保无人机不会进入禁飞区

细节（正式运行前）：

1. Exit：

程序在没有正确的命令行参数的情况下运行（输入的命令行参数不正确）：

[exit](#)

error message: 'Invalid command line arguments. Usage: <noflyzones> <flightplan>'

2. Exit code 1:

两个文件名是 fopen 可以理解的相对或绝对路径名。

任意一个不能打开，用 perror 输出 error message: "Cannot open X file" 和操作系统具体的错误信息。[Exit with exit code 1.](#)

X：两个文件之一

（先检查禁飞区文件是否能访问和有效，再检查飞行计划文件）

3. Exit code 2:

1 禁飞区文件形式

- a. 行由一个换行符“\n”分隔。
- b. 以字符“#”开头的行是注释，它们的内容应该被忽略，直到行的末尾。
- c. 空行（不带字符的行）应该被忽略。
- d. 描述圆形禁飞区域的每一行将包含由一个或多个空白字符分隔的 3 个整数。每个整数 ≥ 0 且 $< 10,000$ 。
 - 第一个数字是禁飞区中心的 x 坐标。
 - 第二个数字是禁飞区中心的 y 坐标。
 - 第三个数字是禁飞区的半径。
- e. 与该格式不匹配的非空白非注释行无效。Error message: “No-fly zone file invalid”。[Exit code 2.](#)
- f. 文件中禁止区域的数量没有限制。可能是 0。每个禁飞区的半径将始终 > 0 。您可以假设没有禁飞区域重叠。

4. Exit code 3:

错误！未定义书签。飞行计划文件形式

- a. 行由一个换行符“\n”分隔。
- b. 以字符“#”开头的行是注释，它们的内容应该被忽略，直到行的末尾。
- c. 空白行（不含字符的行）应该被忽略。
- d. 描述航点的每一行都将包含由一个或多个空格字符分隔的 2 个整数。每个整数将大于或

等于零且小于 10,000。

- 第一个数字是路点中心的 x 坐标。
- 第二个数字是路点中心的 y 坐标。
- e. 与该格式不匹配的非空白非注释行无效。Error message: "Flight plan file invalid". [Exit code 3](#)
- f. 文件中的路点数没有限制。总是至少有两个连续的路点，他们不会具有相同的坐标。

细节（正式运行时）：-----是否进入禁飞区

1. 第一种 invalid：一个或两个航点在禁区里

2. 第二种：航点不在里面，两个航点之间的线段穿过禁飞区

2.1 写出公式 1 （有 ≥ 1 个在 0 到 1 之间的实根 = 线段穿过了禁飞区）

2.2 求 Discrim

2.2.1 Discrim < 0 则 valid. ["Flight plan valid." Exit with code 0](#)

2.2.2 Else：求根。

2.2.2.1 根满足在 0 到 1 之间 invalid. ["Invalid flight plan. \n" "Enters restricted area around X, Y."](#) (x,y) 是禁飞区中心 [Exit with code 4](#)

2.2.2.2 else：valid. ["Flight plan valid." Exit with code 0](#)

例外

[Exit code 5:](#)

程序因为不能分配内存而需要退出，[error message："Unable to allocate memory. " exit code 5](#)

[Exit code 6:](#)

因之前未提到的原因以外的原因需要退出，print error message。Exit code 6。