Q：为什么本地编译正常，提交上去显示编译错误？

A：可能有以下几种原因

1. 编译器差异：例如，VS 默认引用了一些头文件，而 GCC 没有，典型的例子是 VS 下不显示#include <string.h>也可以使用 memset 等函数，但 GCC 下必须引用#include <string.h>。我们建议大家使用 GCC 以缩小编译器差异，如果是 Windows，建议使用 Mingw-w64 的 x86_64 版本，下载地址：https://sourceforge.net/projects/mingw-w64/files/
2. 操作系统差异：平台使用 Linux 作为编译环境，如果选手使用的不同操作系统，并且引用了操作系统相关的头文件和函数，那么在 Linux 下无法编译通过。
3. CMakeLists.txt 被修改，由于平台编译的时候会用原版 SDK 的 cmake 文件替换，因此通过修改 cmake 附加的一些编译选项和链接库均无法生效，请确保在原版 cmake 文件下是可正常编译的。
4. 目录结构问题，请关注编译运行环境说明，例如 C/C++要求源码和 CMakeLists.txt 必须在压缩包的根目录下，即解压后能直接看到而不是在一个子目录中。
5. C/C++提交压缩包缺少 CMakeLists.txt 文件。该文件被用于判定是不是一个 C/C++源码目录，因此压缩包中必须包含一个 CMakeLists.txt。
6. 对于 Java，请关注平台使用的 JDK 版本为：1.8，并且在编译构建环境中没有网络可用。
7. Java 源代码的入口文件 Main.java 所在包路径及 Main.java 文件名不可修改，否则会编译失败；

Q：为什么本地运行正常，提交上去显示运行错误？

A：可能有以下几种原因

1. 写了日志文件，由于运行环境没有任何写文件的权限，故所有的写入文件均会失败。
2. 内存超限（3.8G），会被系统杀死。
3. 运行中尝试引用了代码包中的其他文件，例如数据表、so 库、可执行程序等。由于运行时只保留最终编译的可执行程序，因此所有对代码包中其他文件的运行时引用都会失败。
4. 调用了特权操作，运行环境无 root 权限，所有特权操作均会失败。
5. 对于 python，请关注 python 的版本，以及是否使用了系统默认支持以外的库。
6. 有访问网络的行为，运行环境无网络环境。

Q：已经输出了控制指令，为什么我的机器人一直不动？

A：大多数情况下是选手没有在输出 OK 之后 flush 标准输出，导致数据留存在缓冲区，没有真正送给判题器。对于 C/C++，可以使用 fflush(stdout)或 cout << flush，对于 java,可以使用 System.out.flush()，对于 python，可以使用 sys.stdout.flush()

Q：为什么 CMake 文件不让选手自己修改？提交修改过的 CMake 文件会怎么样？

A：这是基于安全性和维护比赛公平的考虑。cmake 过于灵活，可以在其中执行任意命令，这可能会让某些选手通过执行一些恶意命令来获取一些竞争优势，或者是绕过代码审查的机制（比如不经编译直接生成上传的执行程序），从而影响比赛公平性。故我们将 cmake 文件锁定了不让修改。提交已修改的 CMake 文件不会生效，系统依然会使用原版 CMake 文件进行构建。

Q：直接提交 SDK 的默认代码也有分数吗？

A：是的，会获得初始分值。

Q：平台会编译出 Release 版本用于运行吗？
A：会

Q：正式赛的题目会变吗？
A：正式赛和练习赛的题目一样，但是所使用的地图不同。故大家可以提前在练习赛的时候优化好代码，但是不要针对练习赛的地图做拟合设计，因为正式赛会换掉所有地图。建议选手在练习赛期间自己多造图，以提升代码在不同地图下的表现能力。

Q：正式赛的地图也会公开吗？
A：会在正式赛开始的时候公开。

Q：复赛和决赛的题目会变吗？
A：复赛将在初赛已有内容的基础上加入新的业务难题，同样，决赛也将在复赛的基础上加入新的业务难题。

Q：今年的比赛，性能是否是一个比较重要的考察点？对 Python 友好吗？
A：今年不直接考察性能，但是更高的性能可以让你处理更高的帧率，以此获得更精细的控制，从而间接提高你的机器人表现能力。如果你已经达到了最高帧率（50FPS），那么继续提升性能没有收益。但请大家注意的是，**在复赛和决赛中会引入新的业务难题，这可能会需要比初赛更高的算力支持**。我们的出题组已经有专家实现了 Python 版 DEMO，验证了 Python 可以顺利完成这道题。
最后，考虑在大多数的真实业务场景中，性能其实都是一个重要的业务指标，因此，性能作为一个间接影响结果的指标，我们认为是合理的。

Q：怎么判断自己的程序是否出现了跳帧？
A：1. 可以通过判题器的日志：player skipped frames:xxx 获取这一信息。2. 可以通过观看回放，点击播放控制的眼睛图标按钮显示出调试信息，即可在金额的底下显示跳帧信息，该方法同样适用于提交平台之后。3. 可以在代码中通过读入的帧 ID 进行判定，如果帧 ID 一直连续，说明没有跳帧，如果断断续续，说明跳帧了。

Q：为什么本地运行结果和提交平台的结果有较大差异？
A：1. 平台使用的随机种子不同，由于该随机种子只影响万分之一的机器人参数，故对选手的算法影响非常小。随机性较大的原因通常是因为选手的算法本身呈现出较大的随机性，例如，在选手自己没有考虑避障算法之前，碰撞就是一个随机事件，而这一随机事件将直接影响结果。再例如，选手在做出工作台任务调度的时候比较随意，只根据最近距离的选，那么这种工作台的任务选择也会成为一个随机事件，从而对结果造成较大浮动。总的来说，当选手的结果越接近最优解的时候，随机浮动会越小。
2. 平台使用的硬件和本地有差异，有可能本地运行没有跳帧的程序，在服务器运行时出现了跳帧，从而导致了结果差异。请检查回放文件是否出现了跳帧（参考前一条 QA），以此判断是否要做出一定的性能冗余。

Q：同一份代码提交到系统后结果会是相同的吗？

A：由于平台使用了固定的随机种子，因此一般来说，同一份代码会得到相同的提交结果，如果不同，可能有以下原因：1.出现了跳帧，因为跳帧可能随机跳在不同位置，所以会得到不同结果，请从回放中进行确认。2.代码自身包含了随机因子，每次的输出指令序列不同。

Q：是否能提供精确的物理预测算法？

A：不提供，因为在真实的业务场景中，有太多可变的环境因素，并不存在可以精确预测机器人运动的方法。所以本次赛题中，我们也不提供任何有关环境的因素，诸如阻力、弹性系数等。

Q：日志显示 Read inf('INF') from player... 是什么意思？

A：说明选手程序输出了不合法的浮点数：INF 或 NAN，请检查输出。这通常是由于选手代码的浮点计算错误引起的，例如出现了除 0，或者传递了不合法范围数值给三角函数等，也有可能是由于输出了未初始化变量。

Q：如何从平台下载比赛回放？

A：由于该需求提出时间较晚，开发小哥表示这个需求来不及实现，故页面上没有提供下载回放的功能，但是可以通过 F12 打开浏览器开发者工具，在网络那一栏中开启抓包，然后点击查看视频，通过抓包的方式抓取查看视频时的回放文件下载链接。

Q：不同地图之间物理引擎是否一致，如不一致是哪里不一致，最多会差多少

A：不同地图的物理引擎完全一致，仅有随机种子不同，如题目所说，随机种子影响万分之一左右的机器人参数信息，因此相同的指令序列在不同的地图下会有极细微的表现差异，这个差异会随着时间的累加而逐步扩大（例如角度只要稍微偏一点，随着行进路线变长，差异就会越来越大）。注意地图数据也会参与随机种子计算，故相同的随机种子参数在不同的地图上也会表现略有差异。

Q：为什么要引入随机种子的设定？

A：随机种子的设定目的旨在不影响选手正常算法的前提下，防止有选手直接打表提交指令序列，故这个随机种子的影响范围被设定的很小，选手在设计算法的时候可以完全忽略它。如果没有这个设定，那么选手很容易在本地得到和服务器完全相同的结果，那么就只需要针对地图做指令序列优化就好了，可以通过 DEBUG 模式花一个通宵跑一个最优指令序列来直接提交，也可以直接提交别人生成的指令序列来绕开代码查重，这些都会成为影响比赛公平公正的因素，故我们加入了这一个设定来维护比赛的公平公正。

Q：有没有可能通过控制回放文件中的表现来慢慢返回环境中的获取信息，以此得到官方随机种子？

A：设计中已经考虑了这个问题，故我们在实现时把这部分算法做了两版不同的实现，也就是平台版和用户版的随机算法实现不同，但它们都保证随机结果在万分之一以内的范围内。故，选手理论上不可能构造出和官方相同行为的随机种子。

Q：提交到线上运行时的日志有没有办法获取

A：没有办法获取，这是基于安全和维护比赛公平的考虑，不让选手通过打日志的方式返回一些正式比赛的环境信息（例如官方随机种子甚至是官方判题器等）以影响比赛的公平性。

Q：C/C++如何使用 SSE/AVX/BMI 等指令集？

A：使用此类指令集需要编译选项的支持，由于 CMake 无法修改，故可以通过在代码中添加：#pragma GCC target("avx") 的方式来支持。