2024 年智能车实验室地平线智慧医疗组考核（一）

C++语言考核题目【B 卷】

（满分 110 + 10 分，共两部分）

在我们的比赛中，我们经常使用 C++和 Python 两种编程语言。基于实际测试和性能分析，C++在处理相似逻辑的代码时表现出显著更高的响应速度和性能优势。因此，我们决定将 C++作为代码构建的主要语言。为了确保代码的高效性、稳定性和可靠性，对 C++进行深入考核是必不可少的。这不仅有助于优化我们的代码结构，还能提高整体系统的运行效率，满足高性能计算的需求。

第一部分 C++基础

该部分评估您使用 C++处理基本问题的技能，请根据要求在 ubuntu 上编写相应的 C++代码。

请先建立一个名叫“First”的文件夹，在该文件夹下进行第一部分代码的编写。

一、基础编程题（每题 20 分）。

小试牛刀——素数判断

编写一个名为 isPrime 的布尔函数，判断 500 以内的整数是否为素数。

代码要求：在 First 文件夹下创建并编写一个名为 prime.cpp 的文件，其中描述并使用指定函数；请实现对以下五组数字的判断：11 12、15 19、17 200、217 310、259 499【每次输入两个数字（用逗号隔开）并在终端输出结果】。

输入样例：

11 12

输出样例：

The first number 11 is true

The second number 12 is false

评分标准：

基础验收【12 分】：正确全部通过题干中五项的验收，可得基础分 12 分，未通过则其他两项将不再验收；

鲁棒性验收【6 分】：通过评分人的附加验收（输入的样例未知）可得附加分 6 分；

代码逻辑验收【2 分】：代码结构清晰、判断全面可得满分。

大试牛刀——字符串元音提取与重组

编写一个名为 addVowels 的函数，针对输入字符串进行以下处理：

从字符串中提取出所有元音字母（a, e, i, o, u/A, E, I, O, U）。

按原顺序将元音字母组合成一个新的字符串。

返回一个新的字符串，格式为：原始字符串+提取的元音字母。

例如，给定输入 HelloWorld，提取出的元音为 eoo，生成的新字符串为 HelloWorldeoo。

代码要求:在 First 文件夹下创建一个名为 vowels.cpp 的文件，并在其中实现 addVowels 函数。

函数要求：遍历字符串并提取元音,返回拼接后的字符串。

请分别处理以下字符串：

umbrella

significant

developer

parallel

mountain

输入示例

developer

输出示例

developereeoe

评分标准：

基础验收【12 分】：函数能正确提取元音并生成新字符串，5 个字符串全部通过验收则得 12 分（若部分通过则得 6 分，其它两项将不再验收）。

程序鲁棒性验收【3 分】：针对评分人的附加测试（如字符串包含特殊字符等情况）能正确处理的可得附加分 3 分。

代码逻辑验收【5 分】：合理使用指针和字符串操作、逻辑结构清晰，命名规范，能获得满分。

老试牛刀——数字分解为平方数之和

编写一个名为 sumOfSquares 的函数，将一个给定的正整数 n 表示为若干个平方数之和。例如，17 可以表示为 16 + 1（即 4^2 + 1^2），而 18 可以表示为 9 + 9 或 16 + 1 + 1。请使完成分解，返回结果中较大的平方数优先显示。

代码要求：在 First 文件夹下创建并编写名为 squares.cpp 的文件，内容符合以下要求：

（1）指定函数签名为：std::vector<int> sumOfSquares (int n);

TIPS：为大家讲解一下这个函数签名是什么意思：

首先是函数的返回值 std::vector<int>表示一个整数类型的动态数组（也称为向量），可以存储多个 int 类型的值，其中包含分解结果中的每个平方数【例如，对于 n = 17，返回 {16, 1}】，然后是 sumOfSquares 是函数的名称，(int n)是参数。

使用示例：

```
int n = 17;
std::vector<int> squares = sumOfSquares (n);
// 输出应为[16，1];
```

（2）终端需要输出类似于以下内容为蓝/紫色字体：

```
Decomposition of 17 into squares: 16 1
```

提示：使用 cin 输入 n 的值，然后输出要求的内容。

输入样例：

17

输出样例：

Decomposition of 17 into squares: 16 1

评分标准

基础验收【10 分】：程序能够正确分解并输出 17、18、23、50、99 的平方数之和，若 5 个数字全部通过验收可得 10 分，若部分通过则得 5 分【终端输出不是蓝色扣 2 分】。

鲁棒性验收【5 分】：对未知正整数 n 的附加测试，能够正确分解成平方数之和可得附加分 5 分。

代码逻辑验收【5 分】：使用递归或者更高级的方法进行分解，结构清晰，命名规范，符合最佳实践。

第一部分到此结束，做完的同学可以提前交卷进行下一部分的考核。

第二部分 C++技能使用

该部分评估您使用 C++处理实际问题的技能，请根据要求在 ubuntu 上编写相应的 C++代码。

请先建立一个名叫“Second”的文件夹，在该文件夹下进行第二部分代码的编写。

二、数据处理题。（20 分）

不试牛刀——订单整理

在线订单数据由一个整数数组 arr 表示，其中每个元素的值表示订单的状态：1 表示订单已完成，0 表示订单未完成。请将数组中所有未完成的订单移到数组前端，已完成的订单移到数组末尾，并保持每个部分的相对顺序不变。

代码要求： 在 Second 文件夹下创建一个名为 orders.cpp 的文件，并在其中实现订单整理功能，具体要求如下：

该程序应能够读取一个整数数组 arr 作为输入，将其中的未完成订单（0）移到数组前端，已完成订单（1）移到数组末端。

示例 1：

输入：1, 0, 1, 0, 1, 0

输出：0, 0, 0, 1, 1, 1

示例 2：

输入：0, 1, 1, 0, 1

输出：0, 0, 1, 1, 1

评分标准：

基础验收【10 分】：正确通过题干中的两项示例验收，可得基础分 10 分，未通过则其他两项将不再验收。

鲁棒性验收【4 分】：通过评分人的附加验收（数组长度和内容未知、输入的内容也未知）可得附加分 4 分。

代码逻辑验收【6 分】：使用**原地修改的**方法实现订单整理，代码逻辑简洁清晰，符合最佳实践可得满分。

三、面向对象的编程与项目构建——地下矿洞探险。（30 分）

题目描述：设计一个程序来模拟地下矿洞探险游戏。每位玩家需要在矿洞中收集宝石，并在矿洞塌陷前尽量带回营地。玩家需要在收集宝石和管理体力之间做出合理选择，并及时返回营地以确保获得的宝石安全无损。

游戏规则：

玩家数量：游戏参与者为 1 到 4 名玩家。

初始资源：

每位玩家初始体力为 100 点，用于支撑在矿洞中的探险活动。

矿洞区域：

游戏设定三种不同的矿洞区域，每个区域的宝石和体力消耗不同：

浅层矿洞：每次进入消耗 10 点体力，可能获得 5 到 10 颗宝石，几乎无风险。

中层矿洞：每次进入消耗 20 点体力，可能获得 10 到 25 颗宝石，有 20%概率遇到塌方导致失去已收集的宝石。

深层矿洞：每次进入消耗 30 点体力，可能获得 20 到 50 颗宝石，有 40%概率遇到塌方导致失去所有已收集的宝石。

玩家操作：

每位玩家在回合中可以选择以下操作之一：

探险：选择进入浅层、中层或深层矿洞，收集宝石并扣除相应体力。

返回营地：将已收集的宝石带回营地以确保安全，体力恢复 20 点。

一旦体力耗尽，玩家必须返回营地。

游戏结束：

当所有玩家都选择返回营地，或全部体力耗尽后，游戏结束。

在游戏结束时，拥有最多宝石的玩家获胜。

输入格式：

用户输入参与的玩家数量（范围 1 至 4）。

每位玩家在每轮选择进入的矿洞区域，输入 "S" 表示浅层、"M" 表示中层、"D" 表示深层，或输入 "B" 表示返回营地。

输出格式：

显示玩家当前的体力、已收集的宝石数量以及营地中的宝石总数。

每次行动显示探险结果，包括宝石数量、体力消耗情况以及是否遭遇塌方。

游戏结束时，显示每位玩家的最终宝石数量，并宣布获胜者。

示例输入：

输入玩家数量：2

Player 1 输入 (S/M/D/B): S

Player 2 输入 (S/M/D/B): D

...

示例输出：

Player 1 在浅层矿洞中获得 8 颗宝石。

Player 2 在深层矿洞中获得 30 颗宝石，但遭遇塌方失去所有宝石。

回合结束：

Player 1 - 体力: 90, 宝石: 8

Player 2 - 体力: 70, 宝石: 0

...

游戏结束。

Player 1 总宝石：50 颗

Player 2 总宝石：30 颗

胜者：Player 1

提示：

使用面向对象编程设计 Player 和 Game 类，管理玩家的体力和宝石数量。游戏中的输入和输出应提供清晰的提示和反馈，让用户了解自己的当前状态和操作结果。

评分标准：

基础验收（20 分）：实现矿洞探险、体力管理和宝石收集等规则，正常执行游戏流程（相似即可）。

代码逻辑验收（10 分）：代码结构清晰，符合面向对象设计和游戏规则，使用 cmake 可得 5 分，能够合理处理随机事件（塌方）可得满分。

附加题：C++项目实战练习（15 分）

开发一个 C++游戏项目可以是一个令人兴奋的体验，尤其是对于想要深入理解计算机底层和编程逻辑的人来说。

1.为什么选择 C++开发游戏？

性能优越：C++允许开发者精确控制内存和处理器的使用，是许多性能要求高的 3D 游戏的首选。

广泛的库支持：C++有丰富的库和框架（如 SFML、SDL、OpenGL 等），提供图形、音效、输入处理等支持。

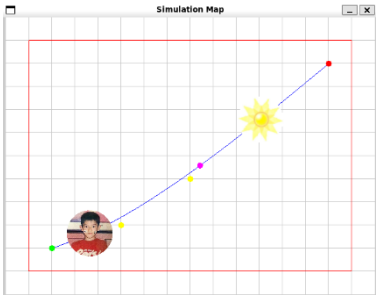
游戏引擎：许多主流的游戏引擎如 Unreal Engine 都使用 C++，学习 C++可以更好地利用这些强大的工具。

2.项目结构

一个 C++游戏项目通常会包含以下结构：

```
GameProject/
├── src/                    # 源代码文件夹
│   ├── main.cpp            # 游戏入口文件
│   ├── Game.cpp            # 游戏主类
│   ├── Player.cpp          # 玩家类
│   ├── Enemy.cpp           # 敌人类
├── include/                # 头文件文件夹
│   ├── Game.h
│   ├── Player.h
│   ├── Enemy.h
├── images/                 # 游戏资源（如图片、声音等）
├── build/                  # 编译输出目录
└── CMakeLists.txt          # CMake 文件，用于跨平台构建项目
```

解压 sim.tar 压缩包，阅读这个文件夹内的所有项目并进行改编代码：

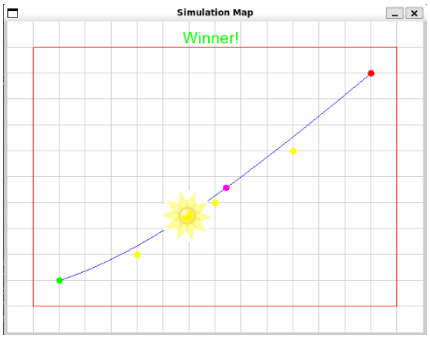


1.可以直接执行 build.sh 脚本进行编译运行，或者使用 cmake、make 命令手动编译运行，然后显示以下的画面【左上角为原点，水平向右为 x 轴正方向，水平向右为 y 轴的正方向】。

2.两端的端点分别为起点和终点，黄点为控制点（3 个），其中 main.cpp 中使用了样条插值的方式来拟合这个轨迹曲线。

3.有两只小精灵（红色叫 winner 黄色叫 loser）分别从起点和终点出发，沿着轨迹曲线以相同的线速度向着中点出发，先到中点的赢，但是作者想让 winner 胜利，然后篡改了这个代码。

首先验收的是你能不能成功配置 SFML、yaml、这三个库【环境问题不要问学长】，成功运行后**把游戏结束画面截图 first.jpg 放在 sim 文件夹下**（5 分），如图：



然后你需要做的是：找到拟合曲线的函数，修改它，让控制点保持在曲线的上方【不限任何方法，只要你能让三个控制点在曲线上方即可】，**把画面截图 second.jpg 放在 sim 文件夹下**（5 分）。

第二部分到此结束，做完的同学可以提前交卷结束考核。