

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【A 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）。

1. 字符串反转：编写一个名为 `reverseString` 函数，实现对输入字符串的反转。

代码要求：在 First 文件夹下创建名为 `string.cpp` 的文件，其中描述并使用指定函数；请实现以下五个字符串的反转：helloworld、tomboy、gidle、drama、qwertyuiop 【每次只反转一个字符串即可，字符串字符长度： $0 \leq \text{length} \leq 10$ 】。

输入样例：

helloworld

输出样例：

dlorwolleh

评分标准：

基础验收：通过题目要求的 5 个字符串验收，可得基础分 12 分，未全部通过可得 6 分，其他两项将不再验收；

程序鲁棒性验收：通过评分人的附加验收（字符串长度未知、输入的内容也未知）可得附加分 3 分；

代码逻辑验收：使用非暴力方式完成字符串反转，如通过指针操作字符数组、使用动态分配内存或合理使用栈空间、代码逻辑与结构清晰可得到满分。

2.斐波那契数列：编写一个函数，生成**指定长度**【 $0 \leq \text{length} \leq 100$ 】的斐波那契数列。

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

(1) **指定函数签名为：** `std::vector<int> generateFibonacci(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；然后是 `generateFibonacci` 是函数的名称，`(int n)` 是参数。

使用示例：

```
int n = 5; // n 就是斐波那契数列的长度

std::vector<int> fib = generateFibonacci(n);

// 输出应为[0, 1, 1, 2, 3];
```

(2) 终端需要输出类似于以下内容**为蓝/紫色字体**：

```
Fibonacci sequence of length 5: 0 1 1 2 3
```

提示：使用 `cin` 输入 `n` 的值，然后输出要求的内容。

输入样例：

5

输出样例：

Fibonacci sequence of length 5: 0 1 1 2 3

评分标准：

基础验收：正确全部通过 **n=3, 5, 10, 12, 15 这五项** 的验收，可得基础分 10 分【输出正确 + 一个黄色的终端输出 1 分】，未全部通过的其他两项将不再验收；

鲁棒性验收：通过评分人的附加验收（`n` 的值未知、输入的内容也未知）可得附加分 6 分；

代码逻辑验收：使用指针实现斐波那契数列生成（如通过指针操作动态分配的数组）、优化内存管理（如使用动态内存分配和释放）、代码结构清晰可得到满分。

3. 素数判断：编写一个名为 `isPrime` 的 `bool` 函数，判断一个 1000 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；请实现对以下五个数字的素数判断：17、18、19、20、21。函数返回值类型为 `bool`，并在终端输出结果。

输入样例：

17

输出样例：

`true/True/TRUE/1` 【这些输出都可以】

评分标准：

基础验收：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：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：代码结构清晰、判断全面可得满分。

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）

1.复写 0：编写一个 zero.cpp 的文件，完成数字 0 复写和移动。

代码要求：给你一个整数数组 arr，请你将该数组中出现的每个零都复写一遍，并将其余的元素向右平移。注意：请不要在超过该数组长度的位置写入元素【意思就是样例 1 中把每个 0 都写一遍后，数组长度变长了，你需要保持和原先数组一样的长度，把超出部分舍去】。

示例 1：

输入：1,0,2,3,0,4,5,0

输出：1,0,0,2,3,0,0,4

示例 2：

输入：1,0,2,3

输出：1,0,0,2

评分标准：

基础验收：正确全部通过题干中两示例的验收，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：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：使用一些高级方法（提示：逆向遍历、双指针法、原地修改）、代码结构清晰可得满分。

三、面向对象的编程与项目构建。(20 分)

“21”点纸牌游戏：设计一个程序来模拟经典的 21 点扑克牌游戏。

游戏规则如下：

- 1.游戏参与者包括庄家和 1 到 7 个玩家。
- 2.每位参与者初始时有两张牌，玩家可以看到所有牌的总分，但庄家有一张牌是隐藏的。
- 3.玩家按顺序决定是否继续要牌（“Hit”）或停止（“Stand”）。若玩家的总分超过 21，则该玩家“爆牌”，即输了比赛。
- 4.当所有玩家完成操作后，庄家揭示隐藏的牌。若庄家总分为 16 或更少，必须继续要牌，直到分数超过 16 或爆牌。
- 5.如果庄家爆牌，则没有爆牌的所有玩家获胜；爆牌的玩家与庄家平局。若庄家未爆牌，则与玩家比较分数，分数高者获胜，相同分数为平局。

输入格式：

用户输入游戏参与的玩家数量（范围 1 至 7）。

玩家在每轮询问时输入 “h”表示要牌，“s”表示停止。

输出格式：

显示初始牌面和得分（庄家有一张牌隐藏）。

玩家操作后显示更新的牌面和得分。

显示庄家操作后的牌面和最终得分。

输出每个玩家的结果（胜利、平局、或输）。

规则说明

牌面分值：2 到 10 取其面值，J/Q/K 为 10 分，A 可为 1 分或 11 分（自动调整以避免爆牌）。

庄家在总分 17 或以上时必须停止要牌。

示例输入：

输入玩家数量：3

Player 1 输入 (h/s): h

Player 1 输入 (h/s): s

Player 2 输入 (h/s): s

Player 3 输入 (h/s): h

...

示例输出：

庄家牌面： [Hidden], 7 of Spades

Player 1 牌面： 9 of Hearts, 5 of Diamonds (总分: 14)

Player 2 牌面： J of Clubs, 6 of Hearts (总分: 16)

Player 3 牌面： A of Spades, 4 of Clubs (总分: 15)

...

庄家揭示牌面： 7 of Spades, 10 of Hearts (总分: 17)

Player 1 爆牌，输。

Player 2 输于庄家。

Player 3 赢！

提示：

使用面向对象编程设计 Card、Deck、Player 类，确保程序结构清晰。

游戏中的输入和输出要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。

评分标准：

基础验收：能够实现当前要求的内容（无论是否使用 cmake），可得基础分 20 分，未实现当前内容但代码能够体现完整的思路可得 8 分。

代码逻辑验收：代码结构整齐、算法高级可得满分。

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

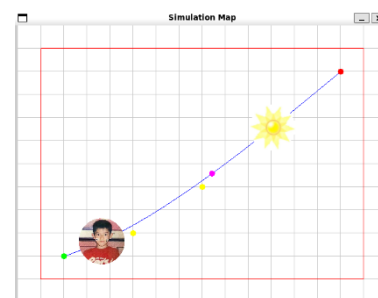
一个 C++ 游戏项目通常会包含以下结构：

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
```

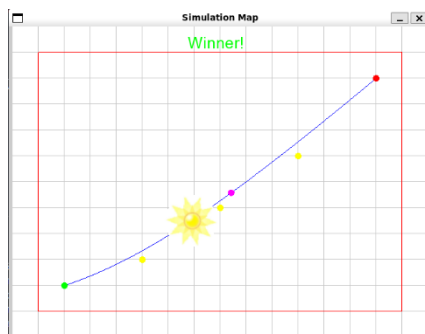
解压 `sim.tar` 压缩包，阅读这个文件夹内的所有项目并进行改编代码：

- 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，垂直向上为 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。