深蓝学院
shenlanxueyuan.com

第三章作业思路讲解

主讲人 苏涛

纲要

深蓝学院
shenlanxueyuan.com

- 第一部分：题目分析
- 第二部分：思路讲解
- 第三部分：常见问题与建议

题目要求

- 输出10道简单的百以内加减法练习题，**每输出一道，等待用户输入答案**，再进行下一题；
- 使用循环实现，且题目是随机的；
- **需要使用if语句检查用户输入的正确性**；
- **10道题目全部回答完毕后**，输出答对、答错的数目，并**一次性输出答错的题目和正确答案**，可以在此制定赋分规则，给用户打分；
- 扩展一：添加简单交互，设置题目数量、难度（数的范围，**乘除法**）
- 扩展二：统计每道题的用时，并向用户输出平均用时、最短用时。

纲要

深蓝学院
shenlanxueyuan.com

- 第一部分：题目分析
- **第二部分：思路讲解**
- 第三部分：常见问题与建议

思路讲解

- 主要分为以下几个步骤
  - 交互获取题目数量
  - 交互获取难度系数
  - 以当前时间设置随机数种子
  - 根据题目总数进行循环
  - 生成题目，并输出给用户
  - 开启计时，接收用户输入答案，并更新总计算时间，单道题最短时间
  - 计算正确值
  - 判断结果是否正确，正确则更新正确题数，否则保存题目，在程序最后一次性输出
  - 输出正确题数，错误题数，得分，错误题目和正确答案，单题最短用时，所有题目平均用时

思路讲解

深蓝学院 shenlanxueyuan.com

- 交互获取题目数量--处理用户无效输入的问题
  - 接收的数据越界
    - 重新接收
  - 接收的数据无法转换成对应的类型
    - 不会将stdin缓存区的数据接收，数据依旧保留，且std::cin的failbit会被置位，之后无法再进行io操作
    - 清零failbit--std::cin.clear();
    - 清空缓存区--fflush(stdin);

```
std::cout << "Please enter the number of questions : ";
while (!(std::cin >> questionNum) || questionNum <= 0) {
    std::cin.clear();
    fflush(stdin);
    std::cerr << "Your input does not meet the requirements, please re-enter : ";
}
```

思路讲解

深蓝学院 shenlanxueyuan.com

- 交互获取难度系数--减少设置难度系数时的条件分支
  - 根据难度系数的不同，我们会设计**数的范围、操作符的个数**，可以利用**建表**，然后用**索引**的方式减少条件分支
  - using DifficultTable = int[5][2]; // 5个难度系数，每一维保存{数的范围，运算符数目}
  - int difficulty = 0; // 定义难度系数，让用户输入，直接将其作为表的索引

```
constexpr DifficultTable difficultTable = {{100, 1}, {100, 2}, {200, 2}, {200, 4}, {500, 4}};
constexpr char opTable[] = "+-*/";
```

- 以当前时间设置随机数种子
  - 循环之前，设置一次即可
  - srand(time(NULL));

思路讲解

- 根据题目总数进行循环
  - 可以使用 for (int i = 1; i <= questionNum; ++i) {}
  - 根据难度系数获取两个操作数，一个操作符
  - 注意**如果是除法操作，除数不能为0**

```
for (int i = 1; i <= questionNum; ++i) {
    int num1 = rand() % difficultTable[difficulty][0]; // 第一个操作数
    int num2 = rand() % difficultTable[difficulty][0]; // 第二个操作数
    int op = rand() % difficultTable[difficulty][1]; // 操作符
    if (op == 3) {    // 防止除数为0
        while (num2 == 0) {
            num2 = rand() % difficultTable[difficulty][0];
        }
    }
```

- 使用字符串保存当前题目的信息，并输出给用户

```
std::string info = std::to_string(i) + ". " + std::to_string(num1) + opTable[op] + std::to_string(num2) + '=';
std::cout << info;
```

思路讲解

深蓝学院 shenlanxueyuan.com

- 开启计时，接收用户输入答案
  - 计时可以使用std::chrono::high_resolution_clock，方式很多
  - **接收用户输入时，同样需要进行判断**
- 计算正确值
  - **根据switch-case计算**
  - 可以预先计算+-*/四种运算结果保存到数组中，然后用索引方式获取
  - 使用函数指针，预先构建四种操作的函数，然后将函数指针保存到数组中，再用索引方式获取函数

```
float sysAns;
switch (op) {
case 0:
    sysAns = num1 + num2;
    break;
case 1:
    sysAns = num1 - num2;
    break;
case 2:
    sysAns = num1 * num2;
    break;
case 3:
    sysAns = static_cast<float>(num1) / num2;
    break;
}
```

思路讲解

深蓝学院 shenlanxueyuan.com

- 判断结果是否正确
  - 浮点数的判等方法：std::abs(sysAns - userAns) < epsilon
  - **可以选择保留两位小数**
    - 如果将sysAns已事先转换成保留两位小数，std::abs(sysAns - userAns) < 0.01
    - 如果sysAns未转换，std::abs(sysAns - userAns) < 0.005
    - 如sysAns=0.104和userAns=0.11，二者是不等的，使用0.005的阈值才成立
  - 结果正确就计数，++correctNum;
  - 结果错误，直接拼接字符串并保存
    - using OutputInfo = std::vector<std::string>; // 类型别名
    - OutputInfo outputInfo; // 用于汇总错题信息，最后输出
    - outputInfo.push_back(info + std::to_string(sysAns) + ", but your answer is " + std::to_string(userAns) + '.');

思路讲解

深蓝学院 shenlanxueyuan.com

- 最终的一次性输出
  - 输出正确题数，错误题数，得分，错误题目和正确答案，单题最短用时，所有题目平均用时

```
std::cout << "\nYou answered " << correctNum << " questions correctly and " << questionNum - correctNum << " questions incorrectly.\n";
std::cout << "Your score is : " << static_cast<float>(correctNum) / questionNum * 100 << '\n';

if (outputInfo.size()) {
    std::cout << "The wrong questions are :\n";
    for (const auto& info : outputInfo) {
        std::cout << "\t" << info << "\n";
    }
}

std::cout << "The shortest time you use is " << minTime << " ms.\n";
std::cout << "The average time you use is " << static_cast<double>(sumTime) / questionNum << " ms.\n";
```

纲要

深蓝学院
shenlanxueyuan.com

- 第一部分：题目分析
- 第三部分：思路讲解
- **第三部分：常见问题与建议**

常见问题

- 变长数组
  - 如果定义变量x，int b[x]（即可变长度的数组）是不被C++标准支持的，但是被C99标准支持
  - 编译器g++/clang++可以编译变长数组，但是这是它们自己的扩展，不是C++标准所支持的，所以不推荐使用这个功能，因为可能在其他编译器中无法使用
  - 推荐使用C++标准库中的动态数组std::vector
- 浮点数的判等方法
  - std::abs(sysAns - userAns) < epsilon

常见问题

- 除数为0情况
  - 很多同学未考虑到该情况
  - 不推荐除数为0时直接赋值为1，最恰当的方法还是重新生成随机数
- 引入除法操作符后的处理
  - 直接丢弃小数部分是不合适的
  - 在生成第二个操作数时，如果能被第一个操作数整除才保留
    - 该方式是很好的避免小数部分处理的方法
    - 如果先生成倍数，与第一个操作数相乘后再作为第二个操作数，可以减少生成第二个操作数的时间

常见问题

深蓝学院
shenlanxueyuan.com

- 逻辑不够清晰
  - 有些同学的条件分支比较混乱，建议可以画个程序流程图梳理思路
- 代码规范
  - 尽量多些注释
  - 后续用户交互比较少的作业中，建议直接使用cpp compiler

深蓝学院
shenlanxueyuan.com

感谢各位聆听！

Thanks for Listening