

程设第一次实验

PART1

在我的 gcc 编译器中，rand() 函数使用的是线性同余算法。在网上查询资料后，发现我的编译器执行的计算如（计算文件 1_1.cpp）

可以得出，三个系数分别为

a=214013

c=2531011

m=32767

这种算法可以使数字较为平均地分布在随机数区间里，同时因为算法简单，不需要提供很大的内存空间就能运行。（计算文件 1_1.cpp）

1. 生成的是伪随机数，如果我直接调用 rand 函数，那么每次输出的数列都是相同的，如果我使用 srand 函数设置初始值，那么在同一个初始值下，rand 输出的数列相同。（计算文件 1_2）
2. 生成随机数的范围可以由 RAND_MAX 函数得到，如果想设置一定范围内的随机数，只需要采用求比例的算法即可：rand()*a/RAND_MAX, 即可算 a 范围内的随机数。
3. 随机种子就是提供给随机数算法的初始值，使用 srand(int) 设置即可
4. 设置随机数种子会改变随机数的数列，可以由（计算文件 1_2）给出修改后的结果。
5. <random> 文件中包含了很多新的随机数生成算法（又称生成器），根据搜索，该头文件包含如下算法：

linear_congruential_engine（rand 使用的线性同余法）

mersenne_twister_engine（Mersenne Twister 生成算法）

subtract_with_carry_engine

discard_block_engine

independent_bits_engine

shuffle_order_engine

default_random_engine

minstd_rand

minstd_rand0

mt19937

mt19937_64

ranlux24_base

ranlux48_base

ranlux24

ranlux48

knuth_b

random_device

PART2

1. 在<iostream>头文件中，定义了 std::cerr（标准错误流，也用于在控制台上输出信息，但通常用于输出错误消息）与 std::clog（标准日志流，也用于在控制台上输出信息，通常用于输出程序的运行日志）。这些输出是为了区分输出的内容。

此外，在头文件<fstream>中定义了 std::ofstream 的输出方式，这个输出可以定向到文件里，通过指定文件名和打开模式来创建文件输出流，然后使用流操作符 (<<) 将数据写入文件。

2. 对于 C 语言的 printf 函数，修改输出格式是简单的，只需要修改输出的格式说明符即可。(计算文件 2_1)

对于 C++，因为 cout 的输出格式无需手动设置，因此，需引入头文件<iomanip>并使用其中的格式修改函数，完成对输出格式的设置。(计算文件 2_2)

3. 在 ANSI 转义码中，使用 \x1B 或 \033 开始一个转义序列，然后定义前景色和背景色，最后使用 m 结束序列。我写了一个简单的输出带颜色输出的程序。(计算文件 2_3)

PART3 (计算文件 3)

小编程思路：

首先，在堆里放一个随机数生成器，并给出三个分别提供 1-100, 1-10, 1-3 的随机数函数。1-100 函数负责提供加法和减法的计算数，1-10 负责提供乘法的计算数，1-3 负责选择模式。

之后，编写三个函数，分别实现加法，减法和乘法，这些函数的形参列表皆为空，在函数内生成随机数，并返回计算值（此处需要在函数内自动剔除不符合题目要求的计算结果）。

主函数思路：由模式选择随机数，每次随机生成模式，在不同的模式里调用不同的算式函数，返回答案。之后，读入用户输入，根据用户输入，返回相应的语句即可。当输入-1 时，跳出大循环，返回答题数和正确数。