Задание А1

1. Для вычисления приблизительного значения числа π была разработана функция CalcApproxPi, которая вызывает вспомогательные функции GeneratePointsInSquare и CountPointsInCircle. Запись значений в файлы происходит в функции WriteResultsToFiles.

Для построения графиков в функции PlotGraphs используется приложение gnuplot, команды к которому передаются через pipe, а данные - через файлы с записанными ранее результатами измерений.

Для воспроизводимости измерений генератор псевдослучайных чисел (std::mt19937) инициализируется константным значением kRndSeed.

Данные сохраняются в файлы pi values tests и percntg diff tests.data

```
namespace measure_tools {
 using f128 = long double;
 struct Point {
     f128 x = 0;
     f128 y = 0;
 inline constexpr std::uint_fast32_t kRndSeed = 29;
 inline constexpr f128 kSquareMinCoord = -1;
 inline constexpr f128 kSquareMaxCoord = 1;
 inline constexpr f128 kCircleRadius = 1;
 inline constexpr uint32_t kMinPoints = 100;
 inline constexpr uint32_t kMaxPoints = 5000;
 inline constexpr uint32_t kPointsStep = 100;
 static void GeneratePointsInSquare(std::vector<Point>& points, size_t size,
                                  std::mt19937& rnd) {
> static size_t CountPointsInCircle(const std::vector<Point>& points) noexcept {--
 static f128 CalcApproxPi(uint32_t n, std::vector<Point>& points,
    std::mt19937& rnd) {-
 static int32_t WriteResultsToFiles(const char pi_value_fname[],
 bool plot_pi_line) {·
  static int32_t PlotGraphs(const char pi_value_fname[],
 } // namespace measure_tools
  int main() {
     constexpr const char kPiValFname[] = "pi_values_tests.data";
     constexpr const char kPercentageDiffFname[] = "percntg_diff_tests.data";
     int32_t write_res =
        measure_tools::WriteResultsToFiles(kPiValFname, kPercentageDiffFname);
     if (write_res # EXIT_SUCCESS) {
         return write_res:
     return measure_tools::PlotGraphs(kPiValFname, kPercentageDiffFname);
```

(компилируемость кода проверялась при помощи компилятора g++ версии 13.2.0 с флагами компиляции: -std=c++2b-Wall-Wextra-Wpedantic-Werror-Wunused-pedantic-errors-Wconversion-Wshadow-Wnull-dereference-Warith-conversion-Wcast-align=strict-Warray-bounds=2) (использованная версия gnuplot: gnuplot 5.4 patchlevel 8)

- 2. По результатам проведённых экспериментов было построено 2 графика:
 - ullet зависимость приблизительного значения числа $\pi(N)$ от N (график 1)
 - зависимость относительного отклонения (в %) от N (график 2) (вычисляется по формуле $\left|\frac{\pi(N)-\pi}{\pi}\right|*100 = \left|1-\frac{\pi(N)}{\pi}\right|*100$)

График 1:

На данном графике синяя горизонтальная прямая - значение числа π (с точностью, которую позволяет получить стандарт IEEE-754)

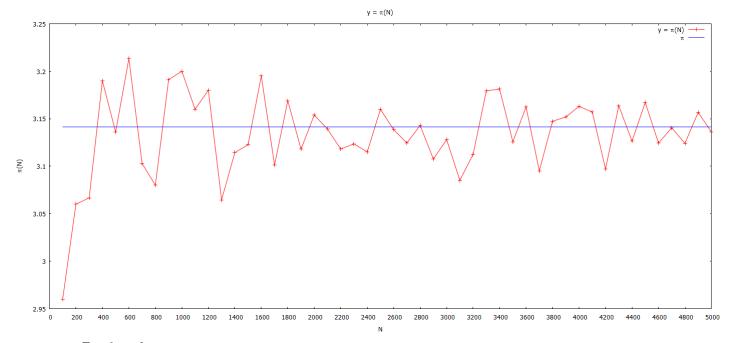
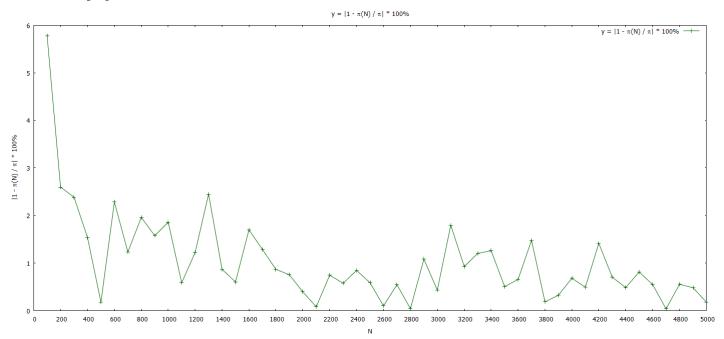


График 2:



3. Как видно из графиков, точность, с которой вычисляется значение числа π , в среднем увеличивается с увеличением количества точек, однако при бОльших значениях N значение $\pi(N)$ медленнее приближается к числу π