

Описание результатов

После запуска программы вы получите два времени выполнения — одно для последовательного доступа и другое для случайного. Обычно последовательный доступ будет значительно быстрее, чем случайный, из-за более эффективного использования кэш-памяти.

Заключение

В этом примере видно, что последовательный доступ к массиву использует кэш-память гораздо более эффективно, чем случайный доступ. Это связано с тем, что при последовательном доступе данные загружаются в кэш блоками, что уменьшает количество обращений к медленной оперативной памяти.

Пример работы программы:

Sequential Sum: 4999999.999999999, Time: 0.123456 seconds

Random Sum: 4999999.999999999, Time: 1.234567 seconds