

Задание

Проверка простоты числа с использованием *SSE*

Алгоритм

Входные данные

Число a (*uint64_t*), над которым производится проверка.

Выходные данные

- Результат проверки;
- Потенциальный делитель
- Затраченное время (мс)

Алгоритм решения

Последовательный перебор делителей в отрезке $[2, \sqrt{a}]$ и проверка наличия остатка от деления. В случае, если остаток найден – алгоритм останавливается, дальнейшая проверка не требуется.

Тесты

- Enter number: 1125899839733759
1125899839733759 is prime!
Time: 0.071

- Enter number: 1125899839733767
1125899839733767 is not prime!
Time: 0
- Enter number: 1125899839733791
1125899839733791 is not prime!
Time: 0

Замечание

Использование целочисленного деления приводит к ограничению на размер проверяемого числа: его величина должна укладываться в мантиссу *double* 2^{52}

Программный код

```
#include <immintrin.h>
#include <iostream>
#include <sstream>
#include <ctime>

bool is_prime_avx(uint64_t n)
{
    if (n < 2) return false;
    if (n == 2 || n == 3) return true;
    if (n % 2 == 0) return false;

    uint64_t sqrtnum = static_cast<double>(sqrt(n)) + 1;
    __m256d numvec = _mm256_set1_pd(n);
    __m256d divvec = _mm256_set_pd(6, 5, 4, 3);
    __m256d tvec = _mm256_set1_pd(4);

    for (uint64_t i = 3; i <= sqrtnum - 4; i += 4)
    {
        __m256d qutvec = _mm256_floor_pd(_mm256_div_pd(numvec, divvec));
        __m256d resvec = _mm256_mul_pd(qutvec, divvec);
        __m256d cmp = _mm256_cmp_pd(numvec, resvec, _CMP_EQ_OQ);
        divvec = _mm256_add_pd(divvec, tvec);

        int mask = _mm256_movemask_pd(cmp);
        if (mask) {
            return false;
        }
    }
}
```

```

    }
}

return true;
}

int main(int argc, char *argv[])
{
    uint64_t number = 0;
    if (argc > 1) {
        std::stringstream sstream;
        sstream << argv[1];
        sstream >> number;
    }
    else {
        std::cout << "Enter number: ";
        std::cin >> number;
    }

    auto timeBeg = clock();
    auto result = is_prime_avx(number);
    auto timeEnd = clock();

    if (result) {
        std::cout << number << " is prime!\n";
    }
    else {
        std::cout << number << " is not prime!\n";
    }
    std::cout << "Time: " << double(timeEnd - timeBeg) / CLOCKS_PER_SEC;

    return 0;
}

```