

Правительство Российской Федерации
Национальный исследовательский университет
«Высшая школа экономики»

Факультет компьютерных наук
Департамент программной инженерии

Домашнее задание 4
Вариант 15

Выполнил
Студент группы БПИ193
Минец Максим

mvminets@edu.hse.ru

Москва 2020

Оглавление

1. Задание	3
2. Решение задачи	4
2.1 Формулировка задания.....	4
2.2 Решение задачи	4
2.3 Формат ввода данных	4
2.4 Формат ввода данных	4
3. Тестирование программы	6
3.1 Тест № 1.....	6
3.2 Тест № 2	6
3.3 Тест № 3	7
3.4 Тест № 4.....	7
3.5 Тест № 5.....	8
3.6 Тест № 6.....	8
3.7 Тест № 7	9
3.8 Тест № 8.....	10
3.9 Тест № 9.....	10
4. Текст программы	11
5. Список использованной литературы.....	14

1. Задание

Вывести список всех целых чисел, содержащих от 4 до 9 значащих цифр, которые после умножения на n , будут содержать все те же самые цифры в произвольной последовательности и в произвольном количестве. Входные данные: целое положительное число n , больше единицы и меньше десяти. Количество потоков является входным параметром.

2. Решение задачи

2.1 Формулировка задания

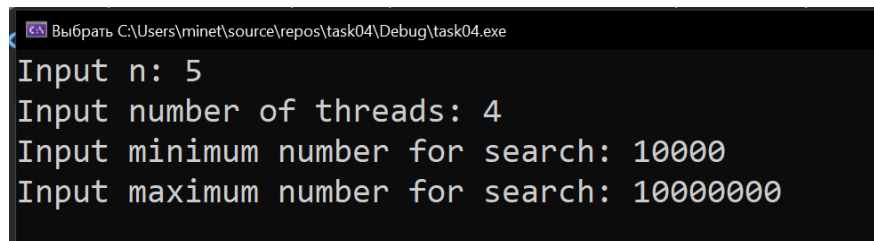
Другими словами, задача заключается в том, чтобы пройти все числа от 1000 до 999999999, которые при умножении на число n (лежащее в промежутке от 1 до 10), содержит все числа, лежащие в промежутке от 1000 до 999999999 и вывести эти числа в консоль.

2.2 Решение задачи

Прочитать входные данные (см. пункт 2.3), запустить метод выбора, подходящих по условию чисел. Весь массив (автоматически) равномерно распределяется между всеми потоками.

2.3 Формат ввода данных

Ввод входных данных осуществлен через консоль (не через командную строку). Для запуска необходимо ввести в консоль четыре аргумента: число n (больше единицы и меньше десяти – ограничения заданы в условии задачи), количество потоков (больше нуля и меньше тысячи – данные границы установлены на усмотрение разработчика), нижнюю границу для проверки чисел (больше 999 и меньше 1000000000) и верхнюю границу для проверки чисел (больше нижней границе и меньше 1000000000). Пример входных данных продемонстрирован на рис. 1.



```
Выбрать C:\Users\minet\source\repos\task04\Debug\task04.exe
Input n: 5
Input number of threads: 4
Input minimum number for search: 10000
Input maximum number for search: 10000000
```

рис. 1

2.4 Формат вывода данных

Результат программы – три числа:

- 1) num – число, которое при умножении на число n (лежащее в промежутке от 1 до 10), содержит все числа, лежащие в промежутке от 1000 до 999999999;
- 2) $n * num$ – число, которое является результатом произведения чисел num (см. пункт выше) и n .
- 3) $current\ thread$ – номер потока, который выполнил проверку для числа num .

Каждая итерация выводится с новой строки. Пример выходных данных продемонстрирован на рис. 2.

```
Консоль отладки Microsoft Visual Studio
Input n: 5
Input number of threads: 3
Input minimum number for search: 1000
Input maximum number for search: 10000

Program has started its work.

num = 4099      n*num = 20495      current thread = 1
num = 1225      n*num = 6125       current thread = 0
num = 4499      n*num = 22495      current thread = 1
num = 2005      n*num = 10025      current thread = 0
num = 4992      n*num = 24960      current thread = 1
num = 9499      n*num = 47495      current thread = 2
num = 2025      n*num = 10125      current thread = 0
num = 4994      n*num = 24970      current thread = 1
num = 9994      n*num = 49970      current thread = 2
num = 2050      n*num = 10250      current thread = 0
num = 4999      n*num = 24995      current thread = 1
num = 9999      n*num = 49995      current thread = 2
num = 2225      n*num = 11125      current thread = 0
num = 5225      n*num = 26125      current thread = 1
num = 2251      n*num = 11255      current thread = 0
num = 2499      n*num = 12495      current thread = 0
num = 2511      n*num = 12555      current thread = 0
num = 2512      n*num = 12560      current thread = 0
num = 2515      n*num = 12575      current thread = 0
num = 2519      n*num = 12595      current thread = 0
num = 6939      n*num = 34695      current thread = 1

Program has finished its work.
```

рис. 2

Также программа уведомляет о начале (см. рис. 3) и о завершении (см. рис. 4) своей работы.

```
Консоль отладки Microsoft Visual Studio
Input n: 5
Input number of threads: 3
Input minimum number for search: 1000
Input maximum number for search: 10000

Program has started its work.
```

рис. 3

```
Program has finished its work.
```

рис. 4

3. Тестирование программы

3.1 Тест № 1

При некорректном вводе числа n , программа предупреждает о некорректном вводе и просит пользователя ввести число еще раз. Пограничными значениями являются числа 1 и 10 (по условию задачи).

```
Консоль отладки Microsoft Visual Studio
Input n: -1
Incorrect input. Try again (number should be > 0 and < 10): 0
Incorrect input. Try again (number should be > 0 and < 10): 11
Incorrect input. Try again (number should be > 0 and < 10): 120
Incorrect input. Try again (number should be > 0 and < 10): 50
Incorrect input. Try again (number should be > 0 and < 10): 4
Input number of threads: 6
Input minimum number for search: 10000
Input maximum number for search: 100000

Program has started its work.
```

3.2 Тест № 2

При некорректном вводе количества потоков, программа предупреждает о некорректном вводе и просит пользователя ввести число еще раз. Максимальным числом потоков взято число 1000 (на усмотрение разработчика).

```
Консоль отладки Microsoft Visual Studio
Input n: 4
Input number of threads: -1
Incorrect input. Try again (number should be > 0 and < 1000): 0
Incorrect input. Try again (number should be > 0 and < 1000): 1000000
Incorrect input. Try again (number should be > 0 and < 1000): 2345
Incorrect input. Try again (number should be > 0 and < 1000): -234
Incorrect input. Try again (number should be > 0 and < 1000): 10
Input minimum number for search: 1000
Input maximum number for search: 100000

Program has started its work.
```

3.3 Тест № 3

При некорректном вводе минимальной границы для поиска чисел, программа предупреждает о некорректном вводе и просит пользователя ввести число еще раз. Минимальным числом для данной границы может быть число 1000, а максимальным 999999999 (по условию).

```
Консоль отладки Microsoft Visual Studio
Input n: 4
Input number of threads: 5
Input minimum number for search: -1
Incorrect input. Try again(number should be > 999 and < 1000000000): 5
Incorrect input. Try again(number should be > 999 and < 1000000000): 0
Incorrect input. Try again(number should be > 999 and < 1000000000): 999
Incorrect input. Try again(number should be > 999 and < 1000000000): -1005
Incorrect input. Try again(number should be > 999 and < 1000000000): 1001
Input maximum number for search: 56432

Program has started its work.
```

3.4 Тест № 4

При некорректном вводе максимальной границы для поиска чисел, программа предупреждает о некорректном вводе и просит пользователя ввести число еще раз. Минимальным числом для данной границы может быть минимальная граница (иначе поиск не сработает), а максимальным 999999999 (по условию).

```
Консоль отладки Microsoft Visual Studio
Input n: 5
Input number of threads: 4
Input minimum number for search: 5000
Input maximum number for search: 999
Incorrect input. Try again(number should be > 5000 and < 1000000000): 4999
Incorrect input. Try again(number should be > 5000 and < 1000000000): -1
Incorrect input. Try again(number should be > 5000 and < 1000000000): 0
Incorrect input. Try again(number should be > 5000 and < 1000000000): 10000

Program has started its work.

           num = 5225           n*num = 26125           current thread = 0
           num = 9499           n*num = 47495           current thread = 3
           num = 6939           n*num = 34695           current thread = 1
           num = 9994           n*num = 49970           current thread = 3
           num = 9999           n*num = 49995           current thread = 3

Program has finished its work.
```

3.5 Тест № 5

Работа программы при корректных данных: число $n = 5$, количество потоков = 10, минимальная граница = 1000, а максимальная = 5000.

```
Консоль отладки Microsoft Visual Studio
Input n: 5
Input number of threads: 10
Input minimum number for search: 1000
Input maximum number for search: 5000

Program has started its work.

      num = 2225      n*num = 11125      current thread = 3
      num = 2005      n*num = 10025      current thread = 2
      num = 1225      n*num = 6125       current thread = 7
      num = 4099      n*num = 20495      current thread = 0
      num = 2251      n*num = 11255      current thread = 3
      num = 4992      n*num = 24960      current thread = 9
      num = 4499      n*num = 22495      current thread = 8
      num = 2025      n*num = 10125      current thread = 2
      num = 2499      n*num = 12495      current thread = 3
      num = 4994      n*num = 24970      current thread = 9
      num = 2050      n*num = 10250      current thread = 2
      num = 2511      n*num = 12555      current thread = 3
      num = 4999      n*num = 24995      current thread = 9
      num = 2512      n*num = 12560      current thread = 3
      num = 2515      n*num = 12575      current thread = 3
      num = 2519      n*num = 12595      current thread = 3

Program has finished its work.
```

3.6 Тест № 6

Работа программы при корректных данных: число $n = 7$, количество потоков = 50, минимальная граница = 5000, а максимальная = 15000.

```
Консоль отладки Microsoft Visual Studio
Input n: 7
Input number of threads: 50
Input minimum number for search: 5000
Input maximum number for search: 15000

Program has started its work.

      num = 14497      n*num = 101479      current thread = 47
      num = 14499      n*num = 101493      current thread = 47
      num = 7331       n*num = 51317       current thread = 11
      num = 5797       n*num = 40579       current thread = 3
      num = 7797       n*num = 54579       current thread = 13
      num = 6664       n*num = 46648       current thread = 8
      num = 10015      n*num = 70105       current thread = 25
      num = 11661      n*num = 81627       current thread = 33
      num = 9677       n*num = 67739       current thread = 23
      num = 9996       n*num = 69972       current thread = 24
      num = 7497       n*num = 52479       current thread = 12
      num = 5335       n*num = 37345       current thread = 1
      num = 6666       n*num = 46662       current thread = 8
      num = 11666      n*num = 81662       current thread = 33
      num = 9687       n*num = 67809       current thread = 23
      num = 9999       n*num = 69993       current thread = 24
      num = 6674       n*num = 46718       current thread = 8
      num = 9697       n*num = 67879       current thread = 23
      num = 6676       n*num = 46732       current thread = 8
      num = 9699       n*num = 67893       current thread = 23
      num = 6677       n*num = 46739       current thread = 8
      num = 9797       n*num = 68579       current thread = 23
      num = 6684       n*num = 46788       current thread = 8
      num = 6686       n*num = 46802       current thread = 8
      num = 6688       n*num = 46816       current thread = 8
      num = 6699       n*num = 46893       current thread = 8

Program has finished its work.
```


3.7 Тест № 7

Работа программы при корректных данных: число $n = 9$, количество потоков = 150, минимальная граница = 100000, а максимальная = 120000.

```
Консоль отладки Microsoft Visual Studio
Input n: 9
Input number of threads: 150
Input minimum number for search: 100000
Input maximum number for search: 120000

Program has started its work.

      num = 111225      n*num = 1001025      current thread = 84
      num = 100015      n*num = 900135      current thread = 0
      num = 100150      n*num = 901350      current thread = 1
      num = 112491      n*num = 1012419      current thread = 93
      num = 111125      n*num = 1000125      current thread = 83
      num = 111250      n*num = 1001250      current thread = 84
      num = 100115      n*num = 901035      current thread = 0
      num = 100151      n*num = 901359      current thread = 1
      num = 112499      n*num = 1012491      current thread = 93
      num = 111251      n*num = 1001259      current thread = 84
      num = 100125      n*num = 901125      current thread = 0
      num = 100175      n*num = 901575      current thread = 1
      num = 112500      n*num = 1012500      current thread = 93
      num = 100191      n*num = 901719      current thread = 1
      num = 112501      n*num = 1012509      current thread = 93
      num = 100199      n*num = 901791      current thread = 1
      num = 112502      n*num = 1012518      current thread = 93
      num = 112505      n*num = 1012545      current thread = 93
      num = 112510      n*num = 1012590      current thread = 93
      num = 112511      n*num = 1012599      current thread = 93

Program has finished its work.
```

3.8 Тест № 8

Работа программы при корректных данных: число $n = 5$, количество потоков = 500, минимальная граница = 450000, а максимальная = 550000.

```
Консоль отладки Microsoft Visual Studio
Input n: 5
Input number of threads: 500
Input minimum number for search: 450000
Input maximum number for search: 550000

Program has started its work.

num = 464939      n*num = 2324695      current thread = 74
num = 494499      n*num = 2472495      current thread = 222
num = 487477      n*num = 2437385      current thread = 187
num = 499299      n*num = 2496495      current thread = 246
num = 499922      n*num = 2499610      current thread = 249
num = 492992      n*num = 2464960      current thread = 214
num = 494992      n*num = 2474960      current thread = 224
num = 462939      n*num = 2314695      current thread = 64
num = 492499      n*num = 2462495      current thread = 212
num = 522225      n*num = 2611125      current thread = 361
num = 499499      n*num = 2497495      current thread = 247
num = 466939      n*num = 2334695      current thread = 84
num = 469239      n*num = 2346195      current thread = 96
num = 499924      n*num = 2499620      current thread = 249
num = 492994      n*num = 2464970      current thread = 214
num = 494994      n*num = 2474970      current thread = 224
num = 469299      n*num = 2346495      current thread = 96
num = 499929      n*num = 2499645      current thread = 249
num = 492999      n*num = 2464995      current thread = 214
num = 494999      n*num = 2474995      current thread = 224
num = 469339      n*num = 2346695      current thread = 96
num = 499942      n*num = 2499710      current thread = 249
num = 469392      n*num = 2346960      current thread = 96
num = 499944      n*num = 2499720      current thread = 249
num = 469393      n*num = 2346965      current thread = 96
num = 499949      n*num = 2499745      current thread = 249
num = 469394      n*num = 2346970      current thread = 96
num = 499992      n*num = 2499960      current thread = 249
num = 469396      n*num = 2346980      current thread = 96
num = 499994      n*num = 2499970      current thread = 249
num = 469399      n*num = 2346995      current thread = 96
num = 499999      n*num = 2499995      current thread = 249

Program has finished its work.
```

3.9 Тест № 9

Работа программы при корректных данных, близких к пограничным максимальным значениям: число $n = 9$, количество потоков = 99, минимальная граница = 999998999, а максимальная = 999999999.

```
Консоль отладки Microsoft Visual Studio
Input n: 9
Input number of threads: 99
Input minimum number for search: 999998999
Input maximum number for search: 999999999

Program has started its work.

num = 999999669      n*num = 410062429      current thread = 65
num = 999999969      n*num = 410065129      current thread = 95
num = 999999699      n*num = 410062699      current thread = 68
num = 999999999      n*num = 410065399      current thread = 98

Program has finished its work.
```

4. Текст программы

```
#include <iostream>
#include "omp.h"
#include <iomanip>
#include <string>
#include <chrono>
#include <mutex>

/*
    Минец Максим
    БПИ-193

    Вариант 15
    Вывести список всех целых чисел, содержащих от 4 до 9 значащих цифр,
    которые после умножения на n, будут содержать все те же самые цифры
    в произвольной последовательности и в произвольном количестве. Входные
    данные: целое положительное число n, больше единицы и меньше десяти.
    Количество потоков является входным параметром.
*/

// Количество чисел от первого с 4 значащими цифрами по максимальное из 9 цифр.
constexpr int VALUES_AMOUNT = 999999000;
// lowerBound - минимальное четырехзначное число - 1;
// upperBound - максимальное девятизначное число.
constexpr int lowerBound = 999, upperBound = 999999999;
std::mutex mut;

/// <summary>
/// Пользователь вводит число n, на которое умножаются числа.
/// </summary>
/// <returns> Возвращает число n, на которое умножаются числа </returns>
int input_n()
{
    int num;

    do
    {
        std::cin >> num;
        if (num <= 1 || num >= 10)
            std::cout << "Incorrect input. Try again (number should be > 0 and <
10): ";
    } while (num <= 1 || num >= 10);

    return num;
}

/// <summary>
/// Пользователь вводит границы, в интервале которых ищутся числа,
/// подходящие в решении данной задачи.
/// </summary>
/// <param name="min"> Минимальная граница введенного числа </param>
/// <returns> Возвращает одну из границ, необходимую при работе программы </returns>
int inputBounds(int min)
{
    int num;

    do
    {
        std::cin >> num;
        if (num <= min || num > upperBound)
            std::cout << "Incorrect input. Try again(number should be > "
<< min << " and < " << upperBound + 1 << "): ";
    }
}
```

```

    } while (num <= min || num > upperBound);

    return num;
}

/// <summary>
/// Пользователь вводит число потоков.
/// </summary>
/// <returns> Возвращает число потоков </returns>
int inputNumberOfThreads()
{
    int num;
    // Поставлена верхняя граница для количества потоков (по усмотрению разработчика):
    100.
    do
    {
        std::cin >> num;
        if (num < 1 || num > 1000)
            std::cout << "Incorrect input. Try again (number should be > 0 and <
1000): ";
    } while (num < 1 || num > 1000);

    return num;
}

void process(int n, int t, int min, int max) {
    // Устанавливаем количество потоков.
    omp_set_num_threads(t);
#pragma omp parallel for num_threads(t)
    for (int i = min; i <= max; ++i) {
        // Текущая цифра.
        std::string number = std::to_string(i);
        // Текущая цифра, умноженная на n.
        std::string result = std::to_string(i * n);
        // Флаг, показывающий, соблюдает ли новое число необходимым условиям.
        bool flag = true;

        for (char num : number) {
            unsigned int lhs = 0;
            unsigned int rhs = result.length() - 1;
            while (lhs < rhs - 1) {
                unsigned int mid = lhs + (rhs - lhs) / 2;
                result[mid] < num ? lhs = mid : rhs = mid;
            }
            if (!(result[lhs] == num || result[rhs] == num)) {
                flag = false;
                break;
            }
        }

        // Вывод подходящих нам чисел.
        if (flag) {
            mut.lock();
            std::cout << std::setfill(' ') << std::setw(25) << "num = " << number
                << std::setfill(' ') << std::setw(25) << "n*num = " << result
                << std::setfill(' ') << std::setw(30) << "current thread = "
                << std::to_string(omp_get_thread_num()) << "\n";
            mut.unlock();
        }
    }
}

```

```

int main(int argc, char* argv[]) {
    std::cout << "Input n: ";
    int n = input_n();

    std::cout << "Input number of threads: ";
    int threads = inputnumberOfThreads();

    std::cout << "Input minimum number for search: ";
    int minBound = inputBounds(lowerBound);

    std::cout << "Input maximum number for search: ";
    int maxBound = inputBounds(minBound);

    std::cout << "\nProgram has started its work.\n\n";

    process(n, threads, minBound, maxBound);

    std::cout << "\nProgram has finished its work.\n";

    return 0;
}

```

5. Список использованной литературы

1. Википедия. OpenMP [Электронный ресурс] -<https://ru.wikipedia.org/wiki/OpenMP>
2. Грегори Р. Эндрюс. Основы многопоточного, параллельного и распределенного программирования. - М.: Издательский дом "Вильямс", 2003.
3. Легалов А. И. Учебно-методические материалы [Электронный ресурс] - <http://softcraft.ru/edu/comparch/>