

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](mailto:mvminets@edu.hse.ru)

Москва 2020

## Оглавление

<b>1. Задание .....</b>	<b>3</b>
<b>2. Решение задачи .....</b>	<b>4</b>
2.1 Формулировка задания.....	4
2.2 Решение задачи .....	4
2.3 Формат ввода данных .....	4
2.4 Формат ввода данных .....	4
<b>3. Тестирование программы .....</b>	<b>7</b>
3.1 Тест № 1.....	7
3.2 Тест № 2.....	7
3.3 Тест № 3.....	8
3.4 Тест № 4.....	10
3.5 Тест № 5.....	11
<b>4. Текст программы .....</b>	<b>13</b>
<b>5. Список использованной литературы.....</b>	<b>15</b>

## 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 два аргумента: число  $n$  (больше единицы и меньше десяти – ограничения заданы в условии задачи) и количество потоков (больше нуля и меньше тысячи – данные границы установлены на усмотрение разработчика). Пример входных данных продемонстрирован на рис. 1.

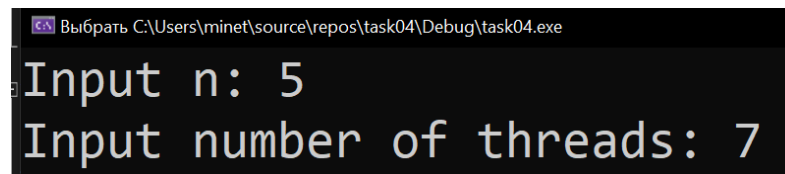


рис. 1

### 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 и рис. 3.

```
Консоль отладки Microsoft Visual Studio
Input n: 4
Input number of threads: 3
Program has started its work.

num = 1282      n*num = 5128      current thread = 0
num = 8222      n*num = 32888    current thread = 2
num = 2282      n*num = 9128     current thread = 0
num = 5882      n*num = 23528    current thread = 1
num = 8282      n*num = 33128    current thread = 2
num = 2812      n*num = 11248     current thread = 0
num = 6662      n*num = 26648    current thread = 1
num = 2821      n*num = 11284     current thread = 0
num = 2822      n*num = 11288     current thread = 0
num = 6666      n*num = 26664    current thread = 1
num = 9399      n*num = 37596     current thread = 2
num = 3331      n*num = 13324     current thread = 0
num = 6676      n*num = 26704     current thread = 1
num = 3333      n*num = 13332     current thread = 0
num = 9993      n*num = 39972     current thread = 2
num = 6677      n*num = 26708     current thread = 1
num = 3364      n*num = 13456     current thread = 0
num = 9999      n*num = 39996     current thread = 2
num = 6682      n*num = 26728     current thread = 1
num = 3366      n*num = 13464     current thread = 0
num = 6687      n*num = 26748     current thread = 1
num = 3399      n*num = 13596     current thread = 0
num = 6699      n*num = 26796     current thread = 1
```

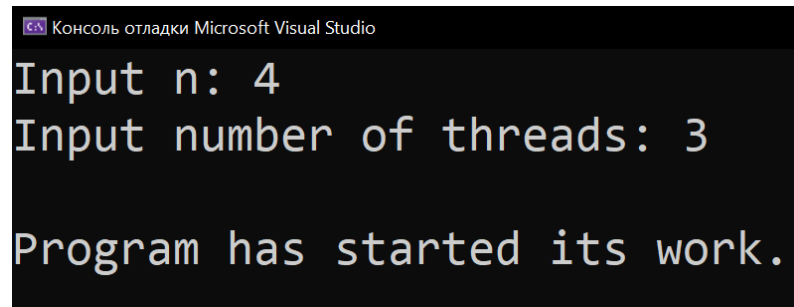
рис. 2

```
Консоль отладки Microsoft Visual Studio
Input n: 5
Input number of threads: 50
Program has started its work.

num = 4992      n*num = 24960    current thread = 22
num = 2499      n*num = 12495    current thread = 8
num = 5225      n*num = 26125    current thread = 23
num = 1225      n*num = 6125     current thread = 1
num = 9499      n*num = 47495    current thread = 47
num = 6939      n*num = 34695    current thread = 32
num = 4499      n*num = 22495    current thread = 19
num = 2225      n*num = 11125    current thread = 6
num = 4099      n*num = 20495    current thread = 17
num = 2005      n*num = 10025    current thread = 5
num = 4994      n*num = 24970    current thread = 22
num = 9994      n*num = 49970    current thread = 49
num = 2511      n*num = 12555    current thread = 8
num = 2251      n*num = 11255    current thread = 6
num = 2025      n*num = 10125    current thread = 5
num = 4999      n*num = 24995    current thread = 22
num = 9999      n*num = 49995    current thread = 49
num = 2512      n*num = 12560    current thread = 8
num = 2050      n*num = 10250    current thread = 5
num = 2515      n*num = 12575    current thread = 8
num = 2519      n*num = 12595    current thread = 8
```

рис. 3

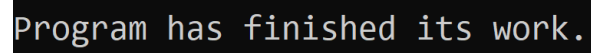
Также программа уведомляет о начале (см. рис. 4) и о завершении (см. рис. 5) своей работы.



Консоль отладки Microsoft Visual Studio

```
Input n: 4  
Input number of threads: 3  
  
Program has started its work.
```

рис. 4



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

рис. 5

### 3. Тестирование программы

#### 3.1 Тест № 1

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): 1005
Incorrect input. Try again (number should be > 0 and < 1000): 5000
Incorrect input. Try again (number should be > 0 and < 1000): -50
Incorrect input. Try again (number should be > 0 and < 1000): 4

Program has started its work.
```

#### 3.2 Тест № 2

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).

```
C:\Users\minet\source\repos\task04\Debug\task04.exe
Input n: -1
Incorrect input. Try again (number should be > 0 and < 10): 0
Incorrect input. Try again (number should be > 0 and < 10): 12
Incorrect input. Try again (number should be > 0 and < 10): 100
Incorrect input. Try again (number should be > 0 and < 10): -500
Incorrect input. Try again (number should be > 0 and < 10): 2000
Incorrect input. Try again (number should be > 0 and < 10): -15
Incorrect input. Try again (number should be > 0 and < 10): 6
Input number of threads:
```

### 3.3 Тест № 3

Работа программы при корректных данных. Число  $n = 8$ , число потоков = 10.

```
Выбрать C:\Users\minet\source\repos\task04\Debug\task04.exe
Input n: 8
Input number of threads: 10

Program has started its work.

num = 60000608      n*num = 505037568      current thread = 6
num = 100001281     n*num = 800010248     current thread = 1
num = 1413          n*num = 11304        current thread = 0
num = 600000808     n*num = 505039168     current thread = 6
num = 100001413     n*num = 800011304     current thread = 1
num = 1572          n*num = 12576        current thread = 0
num = 600000999     n*num = 505040696     current thread = 6
num = 100001572     n*num = 800012576     current thread = 1
num = 1713          n*num = 13704        current thread = 0
num = 100001713     n*num = 800013704     current thread = 1
num = 100001717     n*num = 800013736     current thread = 1
num = 1717          n*num = 13736        current thread = 0
num = 600008608     n*num = 505101568     current thread = 6
num = 100001801     n*num = 800014408     current thread = 1
num = 1811          n*num = 14488        current thread = 0
num = 600008618     n*num = 505101648     current thread = 6
num = 100001810     n*num = 800014480     current thread = 1
num = 1841          n*num = 14728        current thread = 0
num = 1848          n*num = 14784        current thread = 0
num = 100001811     n*num = 800014488     current thread = 1
num = 100001841     n*num = 800014728     current thread = 1
num = 1991          n*num = 15928        current thread = 0
num = 1995          n*num = 15960        current thread = 0
num = 100001848     n*num = 800014784     current thread = 1
num = 600008808     n*num = 505103168     current thread = 6
num = 1999          n*num = 15992        current thread = 0
num = 100001881     n*num = 800015048     current thread = 1
num = 600008833     n*num = 505103368     current thread = 6
num = 2826          n*num = 22608        current thread = 0
num = 100001990     n*num = 800015920     current thread = 1
num = 100001991     n*num = 800015928     current thread = 1
num = 2962          n*num = 23696        current thread = 0
num = 600008858     n*num = 505103568     current thread = 6
num = 100001995     n*num = 800015960     current thread = 1
num = 2992          n*num = 23936        current thread = 0
```

(тут еще много циферок)

(продолжение теста см. на следующий странице)



```

num = 999799579      n*num = -591537960      current thread = 8
num = 999799587      n*num = -591537896      current thread = 8
num = 999226362      n*num = -596123696      current thread = 2
num = 999799975      n*num = -591534792      current thread = 8
num = 999559575      n*num = -593457992      current thread = 5
num = 999959575      n*num = -590257992      current thread = 9
num = 999559577      n*num = -593457976      current thread = 5
num = 999959577      n*num = -590257976      current thread = 9
num = 999675975      n*num = -592526792      current thread = 6
num = 999779975      n*num = -591694792      current thread = 7
num = 999559579      n*num = -593457960      current thread = 5
num = 999959579      n*num = -590257960      current thread = 9
num = 999559587      n*num = -593457896      current thread = 5
num = 999959587      n*num = -590257896      current thread = 9
num = 999789587      n*num = -591617896      current thread = 7
num = 999795975      n*num = -591566792      current thread = 7
num = 999797975      n*num = -591550792      current thread = 7
num = 999859587      n*num = -591057896      current thread = 8
num = 999995555      n*num = -589970152      current thread = 9
num = 999995558      n*num = -589970128      current thread = 9
num = 999995559      n*num = -589970120      current thread = 9
num = 999995585      n*num = -589969912      current thread = 9
num = 999995588      n*num = -589969888      current thread = 9
num = 999995589      n*num = -589969880      current thread = 9
num = 999995595      n*num = -589969832      current thread = 9
num = 999995598      n*num = -589969808      current thread = 9
num = 999995599      n*num = -589969800      current thread = 9
num = 999995855      n*num = -589967752      current thread = 9
num = 999995858      n*num = -589967728      current thread = 9
num = 999995859      n*num = -589967720      current thread = 9
num = 999995885      n*num = -589967512      current thread = 9
num = 999995888      n*num = -589967488      current thread = 9
num = 999995889      n*num = -589967480      current thread = 9
num = 999995895      n*num = -589967432      current thread = 9

num = 999995898      n*num = -589967408      current thread = 9
num = 999995899      n*num = -589967400      current thread = 9
num = 999995955      n*num = -589966952      current thread = 9
num = 999995958      n*num = -589966928      current thread = 9
num = 999995959      n*num = -589966920      current thread = 9
num = 999995985      n*num = -589966712      current thread = 9
num = 999995988      n*num = -589966688      current thread = 9
num = 999995989      n*num = -589966680      current thread = 9
num = 999995995      n*num = -589966632      current thread = 9
num = 999995998      n*num = -589966608      current thread = 9
num = 999995999      n*num = -589966600      current thread = 9
num = 999998555      n*num = -589946152      current thread = 9
num = 999998558      n*num = -589946128      current thread = 9
num = 999998559      n*num = -589946120      current thread = 9
num = 999998585      n*num = -589945912      current thread = 9
num = 999998588      n*num = -589945888      current thread = 9
num = 999998589      n*num = -589945880      current thread = 9
num = 999998595      n*num = -589945832      current thread = 9
num = 999998598      n*num = -589945808      current thread = 9
num = 999998599      n*num = -589945800      current thread = 9
num = 999998855      n*num = -589943752      current thread = 9
num = 999998858      n*num = -589943728      current thread = 9
num = 999998859      n*num = -589943720      current thread = 9
num = 999998885      n*num = -589943512      current thread = 9
num = 999998888      n*num = -589943488      current thread = 9
num = 999998889      n*num = -589943480      current thread = 9
num = 999998895      n*num = -589943432      current thread = 9
num = 999998898      n*num = -589943408      current thread = 9
num = 999998899      n*num = -589943400      current thread = 9
num = 999998955      n*num = -589942952      current thread = 9
num = 999998958      n*num = -589942928      current thread = 9
num = 999998959      n*num = -589942920      current thread = 9
num = 999998985      n*num = -589942712      current thread = 9
num = 999998988      n*num = -589942688      current thread = 9
num = 999998989      n*num = -589942680      current thread = 9
num = 999998995      n*num = -589942632      current thread = 9
num = 999998998      n*num = -589942608      current thread = 9
num = 999998999      n*num = -589942600      current thread = 9

Program has finished its work.

```

Примечание к тесту № 3: (n\*num) является отрицательным из-за переполнения типа данных int.

### 3.4 Тест № 4

Работа программы при корректных данных. Число n = 9, число потоков = 100.

```
Выбрать C:\Users\minet\source\repos\task04\Debug\task04.exe
Input n: 9
Input number of threads: 100
Program has started its work.

num = 1125          n*num = 10125          current thread = 0
num = 959999099    n*num = 50057299    current thread = 96
num = 10001115     n*num = 90010035    current thread = 1
num = 10001125     n*num = 90010125    current thread = 1
num = 100001115    n*num = 900010035    current thread = 10
num = 120001125    n*num = 1080010125    current thread = 12
num = 669999999    n*num = 1735032695    current thread = 67
num = 899999444    n*num = -489939596    current thread = 90
num = 899999448    n*num = -489939560    current thread = 90
num = 899999449    n*num = -489939551    current thread = 90
num = 899999484    n*num = -489939236    current thread = 90
num = 949999977    n*num = -39934799    current thread = 95
num = 230001373    n*num = 2070012357    current thread = 23
num = 1251          n*num = 11259          current thread = 0
num = 1373          n*num = 12357          current thread = 0
num = 10001150      n*num = 90010350    current thread = 1
num = 10001151      n*num = 90010359    current thread = 1
num = 10001175      n*num = 90010575    current thread = 1
num = 590000095     n*num = 1015033559    current thread = 59
num = 590000099     n*num = 1015033595    current thread = 59
num = 590000305     n*num = 1015035449    current thread = 59
num = 590000335     n*num = 1015035719    current thread = 59
num = 899999488     n*num = -489939200    current thread = 90
num = 230002573     n*num = 2070023157    current thread = 23
num = 230002604     n*num = 2070023436    current thread = 23
num = 100001125     n*num = 900010125    current thread = 10
num = 100001150     n*num = 900010350    current thread = 10
num = 100001151     n*num = 900010359    current thread = 10
num = 100001175     n*num = 900010575    current thread = 10
num = 100001191     n*num = 900010719    current thread = 10
num = 100001199     n*num = 900010791    current thread = 10
num = 590000355     n*num = 1015035899    current thread = 59
num = 590000359     n*num = 1015035935    current thread = 59
num = 899999489     n*num = -489939191    current thread = 90
num = 899999494     n*num = -489939146    current thread = 90
```

...

```
num = 999000019    n*num = 401065579    current thread = 0
num = 999939809    n*num = 409523689    current thread = 94
num = 999959909    n*num = 409704589    current thread = 96
num = 999500009    n*num = 405565489    current thread = 50
num = 999019999    n*num = 401245399    current thread = 2
num = 999509500    n*num = 405650908    current thread = 51
num = 999709709    n*num = 407452789    current thread = 71
num = 999050019    n*num = 401515579    current thread = 5
num = 999770000    n*num = 407995408    current thread = 77
num = 999000099    n*num = 409247209    current thread = 91
num = 999099909    n*num = 401964589    current thread = 10
num = 999090009    n*num = 401875489    current thread = 9
num = 999079979    n*num = 401785219    current thread = 8
num = 999600699    n*num = 407011699    current thread = 66
num = 999839809    n*num = 408623689    current thread = 84
num = 999339809    n*num = 404123689    current thread = 34
num = 999030009    n*num = 401335489    current thread = 3
num = 999030010    n*num = 401335498    current thread = 3
num = 999559909    n*num = 406104589    current thread = 56
num = 999000699    n*num = 401611699    current thread = 6
num = 999170709    n*num = 402601789    current thread = 17
num = 999898009    n*num = 409973689    current thread = 99
num = 999200009    n*num = 402865489    current thread = 20
num = 999849909    n*num = 408714589    current thread = 85
num = 999981809    n*num = 409901689    current thread = 98
num = 999799709    n*num = 408262789    current thread = 80
num = 999209809    n*num = 402953689    current thread = 21
num = 999000999    n*num = 401155399    current thread = 1
num = 999449909    n*num = 405114589    current thread = 45
num = 999409699    n*num = 404752699    current thread = 41
num = 999041009    n*num = 401434489    current thread = 4
num = 999041010    n*num = 401434498    current thread = 4
num = 999859909    n*num = 408804589    current thread = 86
num = 999309699    n*num = 403852699    current thread = 31
num = 999309709    n*num = 403852789    current thread = 31
num = 999970709    n*num = 409801789    current thread = 97
num = 999970710    n*num = 409801798    current thread = 97
num = 999870709    n*num = 408901789    current thread = 87
num = 999409909    n*num = 401665489    current thread = 40
```

...

(продолжение теста см. на следующей странице)

...

```

num = 999022620      n*num = 401268988      current thread = 2
num = 999033919      n*num = 401370679      current thread = 3
num = 999019499      n*num = 401240899      current thread = 1
num = 999022621      n*num = 401268997      current thread = 2
num = 999033939      n*num = 401370859      current thread = 3
num = 999019629      n*num = 401242069      current thread = 1
num = 999022629      n*num = 401269069      current thread = 2
num = 999033979      n*num = 401371219      current thread = 3
num = 999019699      n*num = 401242699      current thread = 1
num = 999022669      n*num = 401269429      current thread = 2
num = 999033999      n*num = 401371399      current thread = 3
num = 999019709      n*num = 401242789      current thread = 1
num = 999019710      n*num = 401242798      current thread = 1
num = 999022699      n*num = 401269699      current thread = 2
num = 999019809      n*num = 401243689      current thread = 1
num = 999022779      n*num = 401270419      current thread = 2
num = 999019829      n*num = 401243869      current thread = 1
num = 999022799      n*num = 401270599      current thread = 2
num = 999019909      n*num = 401244589      current thread = 1
num = 999038709      n*num = 401413789      current thread = 3
num = 999022809      n*num = 401270689      current thread = 2
num = 999019910      n*num = 401244598      current thread = 1
num = 999022829      n*num = 401270869      current thread = 2
num = 999039809      n*num = 401423689      current thread = 3
num = 999019919      n*num = 401244679      current thread = 1
num = 999022909      n*num = 401271589      current thread = 2
num = 999019921      n*num = 401244697      current thread = 1
num = 999022910      n*num = 401271598      current thread = 2
num = 999019929      n*num = 401244769      current thread = 1
num = 999022919      n*num = 401271679      current thread = 2
num = 999019944      n*num = 401244984      current thread = 1
num = 999022921      n*num = 401271697      current thread = 2
num = 999019949      n*num = 401244949      current thread = 1
num = 999022929      n*num = 401271769      current thread = 2
num = 999022979      n*num = 401272219      current thread = 2
num = 999022999      n*num = 401272399      current thread = 2
num = 999026329      n*num = 401302369      current thread = 2

```

Program has finished its work.

### 3.5 Тест № 5

Работа программы при корректных данных, где число потоков будет приближено к 1000.  
Число n = 7, число потоков = 995.

```

C:\Users\minet\source\repos\task04\Debug\task04.exe
Input n: 7
Input number of threads: 995
Program has started its work.

num = 1166      n*num = 8162      current thread = 0
num = 901506855      n*num = 2015580689      current thread = 897
num = 888441722      n*num = 1924124758      current thread = 884
num = 901506995      n*num = 2015581669      current thread = 897
num = 901506999      n*num = 2015581697      current thread = 897
num = 901507005      n*num = 2015581739      current thread = 897
num = 555778874      n*num = -404515178      current thread = 553
num = 649246299      n*num = 249756797      current thread = 646
num = 977888819      n*num = -1744712859      current thread = 973
num = 919597575      n*num = 2142215729      current thread = 915
num = 455277577      n*num = -1108024257      current thread = 453
num = 748744868      n*num = 946246780      current thread = 745
num = 391961611      n*num = -1551236019      current thread = 390
num = 391961661      n*num = -1551235669      current thread = 390
num = 917589155      n*num = 2128156789      current thread = 913
num = 88443344      n*num = 619103408      current thread = 88
num = 666333333      n*num = 369366035      current thread = 663
num = 666333336      n*num = 369366056      current thread = 663
num = 666333363      n*num = 369366245      current thread = 663
num = 666333366      n*num = 369366266      current thread = 663
num = 666333367      n*num = 369366273      current thread = 663
num = 831155822      n*num = 1523123458      current thread = 827
num = 67337673      n*num = 471363711      current thread = 67
num = 67337676      n*num = 471363732      current thread = 67
num = 67337677      n*num = 471363739      current thread = 67
num = 67337683      n*num = 471363781      current thread = 67
num = 67337797      n*num = 471364579      current thread = 67
num = 67337967      n*num = 471365769      current thread = 67
num = 161811116      n*num = 1132677812      current thread = 161
num = 981911819      n*num = -1716551859      current thread = 977
num = 177891117      n*num = 1245237819      current thread = 177
num = 177891127      n*num = 1245237889      current thread = 177
num = 177891128      n*num = 1245237896      current thread = 177
num = 511559591      n*num = -714050159      current thread = 509
num = 437186174      n*num = -1234664078      current thread = 435

```

(продолжение теста см. на следующий странице)

```
Консоль отладки Microsoft Visual Studio

num = 999999199      n*num = -1589940199      current thread = 914
num = 999998589      n*num = -1589944469      current thread = 853
num = 999998999      n*num = -1589941599      current thread = 894
num = 999991159      n*num = -1589996479      current thread = 110
num = 999995189      n*num = -1589968269      current thread = 513
num = 999995919      n*num = -1589963159      current thread = 586
num = 999995599      n*num = -1589965399      current thread = 554
num = 999998199      n*num = -1589947199      current thread = 814
num = 999991559      n*num = -1589993679      current thread = 150
num = 999998119      n*num = -1589947759      current thread = 806
num = 999995518      n*num = -1589965966      current thread = 546
num = 999995118      n*num = -1589968766      current thread = 506
num = 999991119      n*num = -1589996759      current thread = 106
num = 999995889      n*num = -1589963369      current thread = 583
num = 999999519      n*num = -1589937959      current thread = 946
num = 999991899      n*num = -1589991299      current thread = 184
num = 999998189      n*num = -1589947269      current thread = 813
num = 999999589      n*num = -1589937469      current thread = 953
num = 999995899      n*num = -1589963299      current thread = 584
num = 999995589      n*num = -1589965469      current thread = 553
num = 999999959      n*num = -1589934879      current thread = 990
num = 999999989      n*num = -1589934669      current thread = 993
num = 999999999      n*num = -1589934599      current thread = 994
num = 999999859      n*num = -1589935579      current thread = 980
num = 999991598      n*num = -1589993406      current thread = 154
num = 999999189      n*num = -1589940269      current thread = 913
num = 999995819      n*num = -1589963859      current thread = 576
num = 999991519      n*num = -1589993959      current thread = 146
num = 999991859      n*num = -1589991579      current thread = 180
num = 999991199      n*num = -1589996199      current thread = 114
num = 999995859      n*num = -1589963579      current thread = 580
num = 999995999      n*num = -1589962599      current thread = 594
num = 999995519      n*num = -1589965959      current thread = 546
num = 999995119      n*num = -1589968759      current thread = 506
num = 999991599      n*num = -1589993399      current thread = 154

Program has finished its work.

C:\Users\minet\source\repos\task04\Debug\task04.exe (процесс 32320) завершил работу с кодом 0.
Нажмите любую клавишу, чтобы закрыть это окно...
```

Примечание к тесту № 5: (n\*num) является отрицательным из-за переполнения типа данных int.

## 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
// upperBound - максимальное девятизначное число
constexpr int lowerBound = 1000, 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 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) {
        // Устанавливаем количество потоков.
        omp_set_num_threads(t);
#pragma omp parallel for num_threads(t)
        for (int i = lowerBound; i <= upperBound; ++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 << "\nProgram has started its work.\n\n";

    process(n, threads);

    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/>