

Правительство Российской Федерации

Федеральное государственное автономное образовательное

учреждение высшего образования «Национальный

исследовательский университет «Высшая школа экономики»

Факультет компьютерных наук

Департамент программной инженерии

Отчет к домашнему заданию По дисциплине

«Архитектура вычислительных систем»

Работу выполнил:

Студент группы БПИ-194 Назмутдинов Р.Р.

Москва 2020

Содержание

Отчет к домашнему заданию По дисциплине.....	0
1. ЗАДАЧА.....	2
2. РЕШЕНИЕ.....	3
2.1. bool IsPrime(int num)	3
2.2. void GenerateArr(unsigned int* arr, int elemCount, int seed)	3
2.3. void checkPairs(std::vector<int> &resVec, unsigned int* arrA, unsigned int* arrB, int startInd, int endInd).....	3
2.4. void ReadNumber(int &num, int minValue, int maxValue = INT_MAX)	3
2.5. int main()	3
Основания функция программы, в ней происходит считывание входных данных, генерация массивов, создание потоков и выполнение задачи.	3
3. КОД ПРОГРАММЫ	4
4. ТЕСТИРОВАНИЕ.....	7
4.1. Корректные входные данные	7
4.2. Некорректные входные данные.....	8
Программа обрабатывает ввод некорректных входных данных и не завершает работу аварийно (см. рис 5).	8
СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ.....	10

1. ЗАДАЧА

Определить множество индексов i , для которых $(A[i] - B[i])$ или $(A[i] + B[i])$ являются простыми числами. Входные данные: массивы целых положительных чисел A и B , произвольной длины ≥ 1000 . Количество потоков является входным параметром.

2. РЕШЕНИЕ

Простое число – это число делителем которого является 1 и оно само. Опишем алгоритм решения данной задачи. Считывается количество элементов в массивах N и количество потоков M , использующихся в решении. Далее создаются массивы размера N , после чего созданные массивы заполняются случайными числами и создается M потоков каждый из которых отвечает за N/M элементов массивов. Потоки проходятся по своим кускам массивов и сохраняют все индексы при которых $(A[i] + B[i])$ или $(A[i] - B[i])$ являются простыми числами. При завершении работы все потоки закрываются и основной поток выводит все индексы при которых выполнялось описанное в задаче условие.

Таким образом в задаче используется итеративный параллелизм при котором используется несколько потоков, каждый из которых содержит в себе циклы.

При написании кода программы задача была разбита на несколько функций:

2.1. bool IsPrime(int num)

- num – проверяемое на простоту число;

Проверяет является ли переданное число простым. Если является, то возвращает True иначе False.

2.2. void GenerateArr(unsigned int* arr, int elemCount, int seed)

- arr – ссылка на заполняемый массив;
- elemCount – количество элементов в массиве;
- seed – ключ генерации;

Заполняет массив, переданный по ссылке, с количеством элементов elemCount случайными положительными числами.

2.3. void checkPairs(std::vector<int> &resVec, unsigned int* arrA, unsigned int* arrB, int startInd, int endInd)

- resVec – ссылка на вектор хранящий в себе подходящие индексы;
- arrA – ссылка на массив A;
- arrB – ссылка на массив B;
- startInd – начальный индекс проверки;
- endIndex – конечный индекс проверки;

Проверяет является ли $arrA[i] + arrB[i]$ или $arrA[i] - arrB[i]$ простым числом и если является, то записывает индекс в вектор result.

2.4. void ReadNumber(int &num, int minValue, int maxValue = INT_MAX)

- num – ссылка по которой будет записано значение;
 - minValue – минимальное значение, которое можно считать;
 - maxValue – максимальное значение, которое можно считать;
- Считывает число в отрезке $[minValue, maxValue]$

2.5. int main()

Основания функция программы, в ней происходит считывание входных данных, генерация массивов, создание потоков и выполнение задачи.

3. КОД ПРОГРАММЫ

```

#include <iostream>
#include <thread>
#include <mutex>
#include <cstdlib>
#include <ctime>
#include <vector>
#include <algorithm>

std::mutex mtx;

/*
 * Студент: Назмутдинов Роман Ренатович
 * Группа: БПИ-194
 * Вариант 13
 * Определить множество индексов i, для которых (A[i] - B[i]) или
 * (A[i] + B[i]) являются простыми числами. Входные данные: массивы целых
 * положительных чисел A и B, произвольной длины  $\geq 1000$ . Количество
 * потоков является входным параметром.
 */

/**
 * Проверяет является ли переданное число простым
 * @param num проверяемое число
 * @return
 */
bool IsPrime(int num) {
    for (int i = 2; i <= num / 2; ++i)
        if (num % i == 0)
            return false;
    return true;
}

/**
 * Заполняет массив случайными числами
 * @param arr ссылка на заполняемый массив
 * @param elemCount количество элементов в массиве
 * @param seed ключ генерации
 */
void GenerateArr(unsigned int* arr, int elemCount, int seed) {
    srand(seed);
    for (int i = 0; i < elemCount; ++i) {
        arr[i] = abs(rand());
    }
}

/**
 * Проверяет является ли arrA[i] + arrB[i] или arrA[i] - arrB[i]
 * простым числом и если является, то записывает индекс в вектор
 * result
 * @param resVec вектор с индексами
 * @param arrA
 * @param arrB
 * @param startInd начальный индекс проверки
 * @param endInd конечный индекс проверки
 */
void checkPairs(std::vector<int> &resVec, unsigned int* arrA, unsigned int* arrB, int
startInd, int endInd) {
    for (int i = startInd; i < endInd; ++i) {
        if (IsPrime(arrA[i] + arrB[i]) || IsPrime(arrA[i] - arrB[i])) { //проверка на
простоту
            mtx.lock(); //не даем потокам возможности записывать данные одновременно
            resVec.push_back(i);

```

```

        mtx.unlock();
    }
}

/**
 * Считывает число в отрезке [minValue, maxValue]
 * @param num ссылка по которой будет записано значение
 * @param minValue минимальное значение
 * @param maxValue максимальное значение
 */
void ReadNumber(int &num, int minValue, int maxValue = INT_MAX) {
    std::cin >> num;
    while (num < minValue || num > maxValue) {
        std::cout << "Incorrect input!" << std::endl;
        std::cout << "Enter number again:";
        std::cin >> num;
    }
}

int main() {
    srand(static_cast<int>(time(0))); //для генерации случайных чисел

    //считываем входные данные
    int size, threadCount;
    std::cout << "Enter size of arrays:";
    ReadNumber(size, 1000, 1000000);
    std::cout << "Enter count of threads:";
    ReadNumber(threadCount, 1, size);

    //для таймера
    std::clock_t t1, t2;
    t1 = std::clock();

    //Создаем массивы
    unsigned int* arrA = new unsigned int[size];
    GenerateArr(arrA, size, rand());
    unsigned int* arrB = new unsigned int[size];
    GenerateArr(arrB, size, rand());

    //Создаем вектор для записи результата
    std::vector<int> result;

    int elemsCountForThread = size / threadCount; //вычисляем количество элементов на
каждый поток
    std::thread threads[threadCount]; //создаем массив потоков
    for (int i = 0; i < threadCount; ++i) {
        int start = i * elemsCountForThread;
        int end = i < threadCount - 1 ? (i + 1) * elemsCountForThread : size;
        threads[i] = std::thread(checkPairs, std::ref(result), arrA, arrB, start, end);
    } //создаем поток

    for (int i = 0; i < threadCount; ++i) {
        threads[i].join(); //объединяем все потоки
    }

    t2 = std::clock(); //останавливаем таймер
    std::cout << std::endl << "time: " << (t2 - t1) / 1000.0 << " sec." << std::endl;
    //выводим время выполнения операции
    std::cout << std::endl << "Result:" << std::endl;
    std::sort(result.begin(), result.end()); //сортируем вектор с нужными индексами
    for (int i = 0; i < result.size(); ++i) {
        std::cout << result[i] << " "; //выводим вектор
    }
}

```

```
}  
  
//удаляем массивы  
delete[] arrA;  
delete[] arrB;  
return 0;  
}
```

4. ТЕСТИРОВАНИЕ

4.1. Корректные входные данные

При вводе корректных входных данных программа выполняет поставленную задачу без каких-либо проблем (см. рис. 1-3). Также можно заметить, что при большем количестве потоков, задача выполняется быстрее (несмотря на то, что количество потоков увеличилось в 4 раза, время уменьшилось в 2 раза так как у компьютера, на котором проводились тесты, двудерный процессор) (см. рис. 3-4)

```

Консоль отладки Microsoft Visual Studio
Enter size of arrays:1000
Enter count of threads:2

time: 0.022 sec.

Result:
2 3 6 8 10 13 14 15 17 18 19 21 23 24 25 27 29 30 36 37 38 44 45 48 51 53 56 57 58 59 60 61 62 65 68 69 7
0 71 72 73 74 77 78 79 81 82 84 85 86 90 91 92 93 94 96 97 98 100 101 102 105 106 107 108 109 111 112 113
115 116 118 119 122 123 125 126 128 129 131 133 134 138 139 140 144 145 146 147 149 150 155 159 160 161
162 163 164 165 166 167 169 170 171 172 173 174 175 177 181 184 185 186 188 189 190 191 192 193 194 195 1
96 197 198 199 201 202 204 205 206 207 208 209 210 212 213 214 219 220 221 227 229 230 231 234 235 237 23
8 239 241 242 243 246 247 249 253 255 256 258 260 264 265 266 267 268 269 270 271 273 275 277 280 281 284
286 288 289 291 293 295 296 304 307 308 309 311 312 313 316 320 321 322 323 324 326 327 328 333 336 337
338 340 342 343 344 348 349 350 351 353 356 357 361 363 364 365 366 369 370 375 376 379 382 383 384 386 3
87 388 389 390 391 396 397 399 400 401 402 403 404 406 407 408 409 410 412 413 414 415 416 417 418 419 42
1 422 423 424 425 426 427 428 430 432 434 435 436 437 438 440 443 445 446 447 450 453 454 455 457 458 460
462 463 465 466 467 469 472 473 475 476 478 479 480 482 484 485 489 490 491 492 493 494 496 498 500 501
503 505 506 508 510 511 512 513 514 515 516 517 521 523 525 526 528 529 531 534 535 537 538 539 540 541 5
44 545 550 555 556 557 559 560 563 564 565 566 567 568 569 570 571 572 573 574 575 577 584 585 586 587 59
0 591 592 595 596 597 598 599 600 602 604 607 609 613 614 615 617 618 620 625 627 630 632 634 636 638 639
642 643 646 647 648 649 650 651 653 654 655 657 658 659 660 662 663 664 666 667 668 669 670 672 674 675
677 679 681 682 683 686 687 688 689 690 691 693 694 695 698 699 700 701 702 704 705 706 707 709 710 711 7
13 714 715 717 719 720 723 724 726 730 731 733 734 735 736 737 738 739 741 742 746 747 749 754 758 762 76
5 766 767 768 769 772 774 775 776 778 779 780 781 786 787 790 792 794 796 797 798 799 801 803 804 805 809
811 812 813 814 817 820 821 822 823 824 827 828 830 831 832 835 837 838 839 840 841 842 844 845 847 849
  
```

Рисунок 1 – Работа программы при передаче корректных входных данных (1000, 2)

```

C:\Users\admin\source\repos\ConsoleApplication1\Debug\ConsoleApplication1.exe
Enter size of arrays:1000000
Enter count of threads:1

time: 8.704 sec.

Result:
2 3 4 5 6 7 8 9 15 17 20 21 24 25 26 28 30 31 33 36 37 40 41 45 46 49 52 54 59 61 62 63 66 67 68 71 72 73 74 76 77 79 80
81 83 84 85 87 88 89 90 91 93 94 95 98 99 103 104 105 106 107 108 110 111 112 113 115 117 118 120 123 125 132 134 136 1
39 140 141 142 144 146 148 149 150 151 152 153 157 159 160 161 163 167 168 170 171 172 174 175 176 178 179 182 183 184 1
85 186 187 190 191 192 194 195 196 200 201 202 203 204 205 207 209 210 212 213 217 220 221 229 230 231 232 234 236 238 2
41 242 243 244 245 247 248 249 251 253 254 255 258 259 260 262 263 265 266 268 269 270 271 273 274 275 277 281 282 283 2
87 288 289 291 292 293 294 295 296 298 299 300 302 303 306 307 308 309 310 311 313 315 318 319 320 321 324 325 327 331 3
32 333 334 335 337 339 340 341 342 345 346 347 348 350 352 353 355 357 358 360 362 363 366 367 368 369 370 371 372 373 3
74 378 379 381 382 383 384 387 389 390 391 392 393 394 396 397 399 401 402 404 406 407 408 409 410 412 413 414 416 417 4
40 421 422 423 424 427 428 429 431 435 436 438 439 441 442 443 444 445 446 447 448 451 452 453 456 458 459 460 461 463 4
66 468 470 474 476 477 478 479 480 481 482 484 485 486 489 490 491 492 494 495 496 498 499 504 509 510 512 514 518 519 5
21 526 527 528 529 532 533 534 535 536 537 540 541 543 544 545 547 548 550 554 556 557 558 559 560 561 563 565 566 567 5
69 572 573 574 575 577 578 579 583 585 587 588 589 590 591 592 595 597 598 600 602 603 604 605 607 609 610 612 614 615 6
18 619 622 624 625 626 627 630 631 632 633 634 635 636 637 638 640 641 642 644 645 647 652 653 654 655 657 658 662 663 6
64 665 666 667 669 670 671 672 673 674 677 681 682 684 687 688 690 691 692 693 694 695 696 697 698 700 701 703 706 707 7
08 713 714 715 718 719 723 724 726 727 728 730 732 733 734 737 739 740 741 742 743 744 745 746 747 748 752 755 756 757 7
58 760 762 763 766 767 769 770 771 773 774 775 778 779 782 783 785 786 788 790 791 792 793 794 795 796 797 798 799 800 8
01 803 804 807 809 811 813 814 815 816 817 818 820 823 824 825 826 827 828 829 830 831 834 835 836 837 839 840 841 842 8
43 845 846 847 848 851 852 853 854 855 857 859 863 866 867 869 870 871 872 873 874 875 876 877 878 880 882 883 885 887 8
89 890 891 895 897 898 900 901 905 908 910 911 913 915 916 917 918 921 923 924 926 927 928 931 932 933 934 937 938 939 9
40 943 945 946 948 949 950 951 952 953 954 957 959 960 961 966 967 969 970 971 972 976 979 980 983 984 986 987 989 990 9
92 994 997 998 999 1000 1002 1003 1004 1005 1008 1009 1011 1012 1013 1014 1015 1016 1017 1018 1019 1020 1021 1026 1027 1
028 1030 1031 1032 1036 1038 1040 1041 1042 1043 1045 1046 1047 1050 1052 1053 1056 1057 1059 1060 1062 1063 1064 1065 1
066 1067 1071 1073 1074 1075 1076 1078 1080 1081 1084 1085 1086 1087 1089 1090 1091 1092 1093 1094 1097 1098 1099 1100 1
101 1103 1105 1107 1109 1111 1113 1115 1119 1120 1122 1123 1125 1126 1127 1128 1129 1130 1131 1133 1135 1137 1138 1140 1
  
```

Рисунок 2 – Работа программы при передаче корректных входных данных (1000000, 1)


```

C:\Users\admin\source\repos\ConsoleApplication1\Debug\ConsoleApplication1.exe
Enter size of arrays:1000000
Enter count of threads:4

time: 4.566 sec.

Result:
1 2 6 8 10 13 16 19 20 21 22 24 28 32 33 34 35 36 37 39 41 42 43 44 45 47 48 50 51 52 53 55 56 57 58 59 60 61 65 66 68 6
9 70 71 72 73 76 77 78 79 83 84 85 86 88 91 93 94 101 102 104 108 110 115 118 119 122 123 124 125 126 128 129 132 135 13
8 139 141 142 143 144 146 148 149 150 152 153 156 161 164 165 167 172 174 175 176 178 180 181 182 183 187 190 191 193 19
4 196 197 199 201 202 205 206 207 208 209 210 211 212 213 216 219 221 222 223 224 225 227 228 231 232 235 236 237 238 24
0 241 242 243 245 246 247 249 250 252 253 254 255 256 257 259 261 264 267 268 269 270 271 274 275 277 278 279 280 281 28
2 283 285 286 287 289 291 293 294 295 296 298 299 300 302 303 304 305 306 307 310 314 316 317 320 323 324 325 326 328 32
9 330 331 332 338 339 340 345 346 347 348 349 350 352 354 355 357 358 360 362 364 366 368 369 370 371 375 378 380 381 38
3 387 389 390 392 396 397 398 401 402 404 406 407 410 412 415 416 417 419 420 421 422 423 425 427 430 431 432 434 436 43
7 440 441 442 443 444 446 449 451 452 453 454 455 456 457 459 460 461 462 465 466 467 470 471 473 475 478 479 480 481 48
4 485 487 488 489 490 493 494 495 496 497 498 499 503 505 507 509 512 515 520 526 528 530 531 534 535 537 538 539 541 54
2 544 545 547 548 549 550 551 552 554 555 556 557 559 561 562 563 566 567 570 572 574 575 576 577 578 579 581 582 586 58
7 591 592 593 598 601 603 604 606 607 608 610 611 612 615 616 618 619 620 623 626 628 632 634 636 638 639 640 642 644 64
8 650 651 652 654 656 660 662 663 665 667 668 669 672 681 682 684 688 691 692 693 694 695 696 697 698 702 704 705 707 70
8 709 710 711 712 714 717 719 720 721 725 726 727 728 729 730 731 732 734 736 737 741 743 744 745 746 748 749 750 751 75
3 754 755 758 759 761 763 766 767 771 774 775 776 777 783 784 785 788 789 792 793 794 797 798 799 800 804 805 807 808 80
9 810 811 812 814 816 817 819 820 822 824 825 826 827 828 829 830 831 834 835 836 837 838 839 841 842 843 849 851 854 85
5 857 858 860 861 862 867 868 869 871 873 874 877 878 879 882 883 884 885 886 887 888 889 891 893 898 899 900 906 907 90
8 909 912 913 916 917 918 919 923 925 927 930 931 934 937 938 939 940 941 942 945 946 947 949 951 952 955 956 958 960 96
2 963 964 965 966 967 969 972 973 974 975 977 978 979 980 981 982 983 984 985 986 989 991 993 994 995 996 997 998 999 10
00 1001 1004 1005 1006 1008 1009 1012 1013 1014 1015 1016 1017 1018 1020 1021 1023 1025 1026 1028 1030 1032 1033 1035 10
36 1037 1039 1040 1041 1042 1043 1046 1047 1048 1052 1053 1055 1057 1058 1061 1063 1064 1066 1069 1070 1071 1072 1073 10
75 1076 1077 1078 1080 1082 1084 1089 1091 1093 1095 1096 1097 1098 1101 1103 1106 1107 1109 1110 1111 1115 1116 1118 11
19 1121 1122 1123 1125 1126 1127 1128 1130 1131 1132 1134 1135 1136 1137 1139 1140 1141 1143 1147 1149 1151 1152 1153 11
54 1155 1156 1157 1158 1162 1165 1168 1170 1171 1172 1174 1175 1177 1178 1179 1180 1181 1182 1184 1185 1186 1188 1189 11

```

Рисунок 3 – Работа программы при передаче корректных входных данных (1000000, 4)

```

Консоль отладки Microsoft Visual Studio
Enter size of arrays:15000
Enter count of threads:11

time: 0.098 sec.

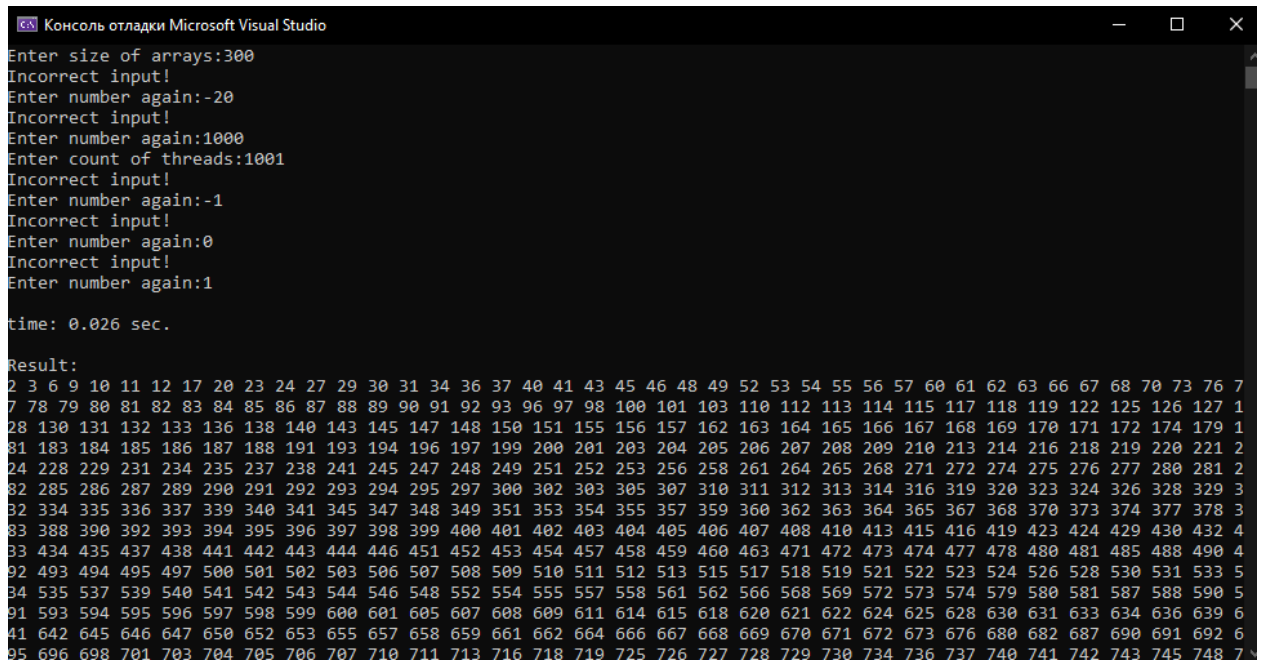
Result:
0 2 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 26 29 30 31 36 39 40 41 42 44 45 46 47 49 50 51 52 57 58 59 60 62 64 6
6 67 68 69 72 74 75 76 77 79 81 83 87 88 89 92 93 95 97 99 101 102 103 104 105 112 114 115 116 117 122 123 128 133 135 1
36 137 140 141 143 145 146 147 149 150 151 153 154 157 159 161 163 165 166 167 168 169 170 173 175 177 178 180 181 182 1
84 185 188 189 190 192 193 194 198 199 200 202 204 205 207 208 210 211 212 214 217 218 220 222 223 226 229 230 232 234 2
35 236 237 238 239 241 242 247 251 252 253 258 259 261 262 263 264 266 270 271 272 273 276 279 281 283 286 287 288 289 2
90 292 293 294 295 296 297 299 300 304 305 309 311 313 314 315 316 317 318 319 320 321 323 325 326 328 329 331 334 335 3
36 338 339 340 343 344 346 349 350 352 355 356 357 359 363 364 366 367 368 369 371 373 374 375 376 377 378 380 381 382 3
83 384 386 387 388 389 390 391 392 395 396 398 399 402 404 406 407 411 412 413 414 415 417 418 419 420 421 422 423 424 4
25 427 428 431 433 436 437 440 441 443 444 445 447 450 452 455 456 457 460 462 465 466 467 468 470 471 472 473 476 477 4
79 480 482 483 484 485 486 488 489 490 491 492 493 494 495 498 499 500 501 502 503 504 505 506 510 511 512 513 514 515 5
16 517 519 520 521 523 526 527 528 530 531 532 534 536 539 541 542 544 545 546 549 551 552 553 554 555 556 558 559 560 5
62 564 566 567 568 569 571 572 573 578 579 581 584 585 586 587 588 589 590 591 592 593 594 596 601 604 605 606 607 608 6
09 611 612 613 616 618 619 621 622 624 625 626 629 634 635 636 639 640 641 643 644 645 647 648 649 650 652 653 654 655 6
56 659 660 663 664 665 666 667 670 671 672 673 674 676 677 678 680 681 683 684 685 686 688 689 691 692 694 696 698 699 7
702 704 706 707 708 711 713 714 717 722 724 726 727 728 730 732 733 736 737 740 743 745 747 750 751 758 759 760 763 7
65 766 767 768 769 771 772 775 776 778 780 781 782 784 785 786 790 791 792 793 794 795 796 798 799 800 801 804 805 807 8
809 810 811 812 815 816 817 818 819 820 824 825 826 827 828 829 830 831 832 835 836 837 838 839 840 841 842 843 844 8
847 849 851 852 854 855 859 860 861 862 863 864 867 868 869 870 872 873 875 880 881 882 884 886 887 889 890 891 893 8
95 900 901 902 905 906 907 908 909 910 912 913 915 916 918 919 920 923 924 925 927 928 929 932 937 938 939 942 943 945 9
947 948 950 952 954 957 961 962 963 965 966 968 972 974 976 978 981 982 985 989 990 993 994 995 997 1001 1002 1004 10
07 1008 1009 1010 1012 1013 1015 1016 1017 1018 1019 1021 1022 1023 1024 1025 1026 1031 1033 1034 1035 1037 1038 1040 10
41 1043 1044 1045 1046 1047 1049 1050 1051 1052 1054 1056 1057 1058 1059 1061 1062 1063 1064 1067 1069 1073 1076 1078 10
79 1082 1083 1085 1086 1087 1088 1089 1091 1092 1093 1094 1095 1098 1100 1101 1103 1104 1105 1106 1110 1111 1112 1113 11
14 1117 1118 1119 1120 1121 1122 1123 1124 1126 1127 1128 1129 1133 1134 1136 1138 1140 1141 1143 1144 1145 1146 1147 11

```

Рисунок 4 – Работа программы при передаче корректных входных данных (15000, 11)

4.2. Некорректные входные данные

Программа обрабатывает ввод некорректных входных данных и не завершает работу аварийно (см. рис 5).



```

Консоль отладки Microsoft Visual Studio
Enter size of arrays:300
Incorrect input!
Enter number again:-20
Incorrect input!
Enter number again:1000
Enter count of threads:1001
Incorrect input!
Enter number again:-1
Incorrect input!
Enter number again:0
Incorrect input!
Enter number again:1

time: 0.026 sec.

Result:
2 3 6 9 10 11 12 17 20 23 24 27 29 30 31 34 36 37 40 41 43 45 46 48 49 52 53 54 55 56 57 60 61 62 63 66 67 68 70 73 76 7
7 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 96 97 98 100 101 103 110 112 113 114 115 117 118 119 122 125 126 127 1
28 130 131 132 133 136 138 140 143 145 147 148 150 151 155 156 157 162 163 164 165 166 167 168 169 170 171 172 174 179 1
81 183 184 185 186 187 188 191 193 194 196 197 199 200 201 203 204 205 206 207 208 209 210 213 214 216 218 219 220 221 2
24 228 229 231 234 235 237 238 241 245 247 248 249 251 252 253 256 258 261 264 265 268 271 272 274 275 276 277 280 281 2
82 285 286 287 289 290 291 292 293 294 295 297 300 302 303 305 307 310 311 312 313 314 316 319 320 323 324 326 328 329 3
32 334 335 336 337 339 340 341 345 347 348 349 351 353 354 355 357 359 360 362 363 364 365 367 368 370 373 374 377 378 3
83 388 390 392 393 394 395 396 397 398 399 400 401 402 403 404 405 406 407 408 410 413 415 416 419 423 424 429 430 432 4
33 434 435 437 438 441 442 443 444 446 451 452 453 454 457 458 459 460 463 471 472 473 474 477 478 480 481 485 488 490 4
92 493 494 495 497 500 501 502 503 506 507 508 509 510 511 512 513 515 517 518 519 521 522 523 524 526 528 530 531 533 5
34 535 537 539 540 541 542 543 544 546 548 552 554 555 557 558 561 562 566 568 569 572 573 574 579 580 581 587 588 590 5
91 593 594 595 596 597 598 599 600 601 605 607 608 609 611 614 615 618 620 621 622 624 625 628 630 631 633 634 636 639 6
41 642 645 646 647 650 652 653 655 657 658 659 661 662 664 666 667 668 669 670 671 672 673 676 680 682 687 690 691 692 6
95 696 698 701 703 704 705 706 707 710 711 713 716 718 719 725 726 727 728 729 730 734 736 737 740 741 742 743 745 748 7

```

Рисунок 5 – Работа программы при вводе некорректных входных данных

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Блог программиста «Парадигмы параллельного программирования» (<https://pro-prof.com/forums/topic/parallel-programming-paradigms>);
2. YouTube канал «#SimpleCode» плейлист «Многопоточное программирование» (https://www.youtube.com/watch?v=NawpxG81RRk&list=PLQOaTSbfxUtAc_RpyDiWCHq0YTzLtVSD0);
3. SoftCraft «Практические приемы построения многопоточных приложений» (<http://www.softcraft.ru/edu/comparch/tasks/t03/>);
4. Wikipedia «Простое число» (<https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D1%81%D1%82%D0%BE%D0%B5%D1%87%D0%B8%D1%81%D0%BB%D0%BE#:~:text=2%2C%203%2C%205%2C%207,%D1%87%D0%B8%D1%81%D0%BB%D0%B0%20%D0%B1%D1%8B%D1%82%D1%8C%20%D0%BF%D1%80%D0%BE%D1%81%D1%82%D1%8B%D0%BC%20%D0%BD%D0%B0%D0%B7%D1%8B%D0%B2%D0%B0%D0%B5%D1%82%D1%81%D1%8F%20%D0%BF%D1%80%D0%BE%D1%81%D1%82%D0%BE%D1%82%D0%BE%D0%B9.>);
5. CyberForum «Таймер C++» (<https://www.cyberforum.ru/cpp-beginners/thread1149197.html>);