

Пензенский государственный университет

Кафедра «Вычислительная техника»

ОТЧЕТ

по лабораторной работе №2

по дисциплине: «Логика и основы алгоритмизации в инженерных задачах.»

на тему: «Оценка времени выполнения программ.»

Выполнили:

студенты группы 21ВВ4

Колокольцева У. А.

Нагорная Д. А.

Принял:

Акифьев И. В.

Юрова О. В.

Пенза, 2022

Цель работы: научиться применять средства, предоставляемые библиотекой time.h.

Лабораторная работа:

Задание 1:

Дана программа, вычисляющая произведение двух квадратных матриц:

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <iostream>
#include <locale>

using namespace std;

const unsigned int n = 100;
int a[n][n], b[n][n], c[n][n];

int main() {

    setlocale(LC_ALL, "rus");

    setvbuf(stdin, NULL, _IONBF, 0);
    setvbuf(stdout, NULL, _IONBF, 0);

    time_t start, end; // объявляем переменные для определения времени выполнения

    int i = 0, j = 0, r;

    start = time(NULL);
    srand(time(NULL)); // инициализируем параметры генератора случайных чисел

    printf("Матрица A:\n\n");

    for (i = 0; i < n; i++)
    {
        for (j = 0; j < n; j++)
        {
            a[i][j] = rand() % 10 + 1; // заполняем массив случайными числами
            printf("%5d", a[i][j]);
        }
        printf("\n");
        printf("\n");
    }

    start = time(NULL);
    srand(time(NULL)); // инициализируем параметры генератора случайных чисел

    printf("Матрица B:\n\n");

    for (i = 0; i < n; i++)
    {
        for (j = 0; j < n; j++)
        {
            b[i][j] = rand() % 10 + 5; // заполняем массив случайными числами
            printf("%5d", b[i][j]);
        }
        printf("\n");
        printf("\n");
    }
}
```

```

printf("Матрица C = A * B:\n\n");

for (i = 0; i < n; i++)
{
    for (j = 0; j < n; j++)
    {
        c[i][j] = 0;
        for (r = 0; r < n; r++)
        {
            c[i][j] += a[i][r] * b[r][j];
        }
        printf("%5d", c[i][j]); // Вывод значения элемента на экран:
    }
    printf("\n");
    printf("\n");
}

end = time(NULL);
printf("Время работы программы: %lf секунд.\n", clock() / 1000.0);

return 0;
}

```

1. Вычислить порядок сложности программы (O -символику).

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <iostream>
#include <locale>

using namespace std;

o(1) { const unsigned int n = 100;
      int a[n][n], b[n][n], c[n][n];

      int main() {

          setlocale(LC_ALL, "rus");

          setvbuf(stdin, NULL, _IONBF, 0);
          setvbuf(stdout, NULL, _IONBF, 0);

          time_t start, end;

o(1) { int i = 0, j = 0, r;

          start = time(NULL);
          srand(time(NULL));

o(1) { printf("Матрица A:\n\n");

          o(n2) {
              for (i = 0; i < n; i++)
              {
                  o(n) {
                      o(1) {
                          for (j = 0; j < n; j++)
                          {
                              o(1) { a[i][j] = rand() % 10 + 1;
                                      printf("%5d", a[i][j]);
                              }
                          }
                      o(1) { printf("\n");
                              printf("\n");
                      }
                  }
              }

          start = time(NULL);
          srand(time(NULL));

o(1) { printf("Матрица B:\n\n");

          o(n2) {
              for (i = 0; i < n; i++)
              {
                  o(n) {
                      o(1) {
                          for (j = 0; j < n; j++)
                          {
                              o(1) { b[i][j] = rand() % 10 + 5;
                                      printf("%5d", b[i][j]);
                              }
                          }
                      o(1) { printf("\n");
                              printf("\n");
                      }
                  }
              }

o(1) { printf("Матрица C = A * B:\n\n");

          o(n3) {
              for (i = 0; i < n; i++)
              {
                  o(n2) {
                      for (j = 0; j < n; j++)
                      {
                          o(1) { c[i][j] = 0;
                                  for (r = 0; r < n; r++)
                                  {
                                      o(1) { c[i][j] += a[i][r] * b[r][j];
                                      }
                                  o(1) { printf("%5d", c[i][j]);
                                          }
                              }
                          o(1) { printf("\n");
                                  printf("\n");
                          }
                      }
                  }
              }

          end = time(NULL);
o(1) { printf("Время работы программы: %lf секунд.\n", clock() / 1000.0);

          return 0;

      }
  }
```

Сложность алгоритма равна $O(n^3)$.

2. Оценить время выполнения программы и кода, выполняющего перемножение матриц, используя функции библиотеки time.h для матриц размерами от 100, 200, 400, 1000, 2000, 4000, 10000.

Количество элементов	100	200	400	1000	2000	4000	10000
Время, с	8,362	26,494	78,945	729,282	2534,775	7378,439	21036,245

Матрицы А и В размером 100 × 100:

```
Консоль отладки Microsoft Visual Studio

4787 5122 5177 5415 5235 4770 5038 5055 4849 5288 4626 4826 4819 5004 4765 4805 5091 4986 4898 4915 5188 5257 5034 4890
4981 5074 4919 5252 4641 4757 5016 5094 5349 5172 5619 5125 5305 5052 5276 5068 5132 5387 5408 4814 4893 4873 4947 5209
5133 5221 4808 4861 5037 5229 4792 5122 4775 4957 4926 5121 5030 4952 4967 5145 4910 5073 4885 5106 5057 4977 4923 4877
5290 4971 4644 4779

5536 5709 6091 5689 5503 5477 5843 6153 5502 5624 5762 5694 5786 5839 5687 5719 5773 5556 5785 5598 5738 5451 5600 5912
5370 5867 5719 5984 5822 5493 5425 5716 5415 5857 5276 5464 5467 5659 5367 5558 5691 5772 5437 5686 5868 5666 5494 5529
5715 5837 5610 5965 5458 5478 5635 5764 5767 5757 6098 5631 5912 5575 5993 5564 5726 6024 5941 5386 5606 5383 5743 5942
5730 5810 5536 5663 5512 5816 5618 5674 5456 5538 5643 5841 5626 5446 5593 5800 5457 5720 5424 5818 5940 5544 5492 5428
5826 5496 5169 5707

4463 4853 5039 4867 4654 4729 4845 5040 4544 4525 4767 4662 4927 4900 4638 4802 4876 4661 4882 4676 4907 4494 4890 5023
4651 4961 4882 4980 4823 4674 4597 4686 4460 4873 4375 4826 4516 4757 4636 4670 4855 4877 4673 4714 4894 4883 4739 4910
4801 5013 4597 4959 4576 4680 4780 4775 4999 4930 5299 4665 4915 4926 4887 4909 4779 5088 5022 4707 4723 4615 4740 4929
4787 4898 4765 4811 4762 4891 4560 4755 4843 4605 4849 4925 4904 4657 4669 4914 4556 4715 4576 4868 5055 4797 4723 4498
4878 4631 4441 4800

4749 4783 5110 4963 4847 4773 4912 5039 4772 4809 5066 4667 4862 4843 5035 4926 5045 4731 4938 4780 4918 4791 5004 5108
4668 5011 5043 5299 5131 4531 4853 4870 4706 5110 4610 4841 4728 4888 4715 4838 4926 4899 4772 4828 5058 5031 4916 4847
4663 4897 4800 5262 4530 4718 4961 4863 5044 4826 5436 4995 5120 4924 4937 4869 4931 5357 5273 4934 4826 4675 4897 5077
4936 5096 4673 4834 4872 4881 4809 4861 4801 4753 4755 4989 5029 4848 4774 4980 4704 4938 4817 4947 5001 4898 4731 4768
5089 4755 4635 4969

Прошло: 8,362000 секунд.

C:\Users\Дарья\Desktop\code\2лиоавизддляб2\Debug\2лиоавизддляб2.exe (процесс 22072) завершил работу с кодом 0.
Чтобы автоматически закрывать консоль при остановке отладки, включите параметр "Сервис" ->"Параметры" ->"Отладка" -> "Автоматически закрыть консоль при остановке отладки".
Нажмите любую клавишу, чтобы закрыть это окно...
```

Матрицы А и В размером 200 × 200:

```
Консоль отладки Microsoft Visual Studio

993310338 9963107701009410061104501022010301105901007310366102311053310146106551059610718 99971051510526106051034810029
988810635100281030010735106671011810137

9591 998010144 9693 998110300 9914 995610076101361018110164 963310610 992610068 9822 995910052100221041510019 9766 9724
10275 994310226 972110039 995110155100381036310140101351022310123 997310068104641013810309 9986 995810018100011012310007
1044510062 9961101291011110008104311028610765101541005010300 984310268 9905101101056110138100111008510161100311025110212
1018110265100351014810227 9947 9822 97311025510461102801031610689 9938 98401034910049101211030510119101111047010333 9926
9992 986810176 957810220 996310446 984910042 9564103691012210314 995010193100181012910081100181026010117 9540 999210331
10001 9965 9772 99211022410437101031018510098 986310280 9864100461034010506101571022010060 9970 9911 9837 9811 9962 9879
1011210360101131015510377 997610272 9889 9856100161012510168 98061021610006 9868100441033010401 997510446 99241034710427
9695 9936 975410557100531016210179 99481005010548 981710106 99801015510032104661030710587 9811103721061310448 9885 9818
972310297 984410219102871038910063 9934

99411082310748104491035410751102951042910457106161081510571102091115210298103491069610595103491074710876106641039710326
103731005410532102231019110298106061063311097108631078210452105651066010608108341058910874104561070510363105501041710662
109191082410653107151077110325105481101810918106671048510865102391056910292107061106510873103471055410489105301083710674
104311074410459107261068910368103451001210468109891060710991112231018910001106481083410550104851038710867110161075110547
107081039710578102091079810437110641040510465102821091510823107331050010657103431069310821106211083610572102651062010717
105731035610226104641090410681106451065210408105251064010118105661062210832104751064910892106931055710099105171037610239
106091093510541105871041610547107721060210465106141055910690104191062510380103391032610806107761039610785106601067010993
101041056010360108961062910244108381055410458109461021510309104571042110799109141088011050103631059211087106431052910231
1003310577106151053310570109511000610377

Прошло: 26,494000 секунд.

C:\Users\Дарья\Desktop\code\2лиоавизддляб2\Debug\2лиоавизддляб2.exe (процесс 3304) завершил работу с кодом 0.
Чтобы автоматически закрывать консоль при остановке отладки, включите параметр "Сервис" ->"Параметры" ->"Отладка" -> "Автоматически закрыть консоль при остановке отладки".
Нажмите любую клавишу, чтобы закрыть это окно...
```

Матрицы А и В размером 400 × 400:

```
Консоль отладки Microsoft Visual Studio

213922125320988213592103821322214892041321201210432130220492210992114120917217632132621501215122063521430204902134321015
211172118920893209512137721006209622068120838215002074320598211232158721654211552112320979203822092620511210902103720888
21067212702123620943215372093421034205222098520904209842115121501215602097321326207721456205272126621952204492098721461
21082212272191821211211302082421526214152145722036208632069721347215282117121540

203452029320423197701990120865200182079520328204482052520483209062056420395198721998520726205512028019922204771983020495
210302008319898199212068320483203081967920154206912053820008206782083320836203032036320352204381996120227206492068120537
20166197152038720903200322035620538205320404202272049020506204352018220843206501973320441205002048319465205612098021205
202582076520714205322002420442210972068120080207452046620888200912028220764210232080420462200342047019908198822022820305
209962084820751202122010720852202772077220736204971993920902204582036620352205822094020846205342072820414204692003021077
206122075120277202812033120566203812050620193208532029520803204722041220879206242030920356202592065720767202772025120562
20872203020181206112050121116208102052920007201522050220220207581980221419210871963520256206452095420783205732035220383
203471940220897205141989120445202481990220802202772089620473205902094620939202762105020294203892071720726206822039820265
206432044120371201852036520986202682119020313201812045119944207942023919982203692048020570206632042020226201432040920391
201622135920749200632105320634205052050720929202712080720162206212023321062206582040621128200302016920352198172014620678
200532052420406203732023520518203762047320512206982065120594203292071820029212072005420849204482067520407203142025920481
2030720685201251973220113202522012920756206172023620331209222073020944206142022220321200972029221065200181211852044820885
197442065120173200072010020090200272067819890205291998421022200542072920716204012007419976196952012419778200632056119908
20455203012027620232204282071020663197782058120010204662018120529203582018120607204892063120794203020872199312021029562
20487208382023820302041920071203961999220075207441982220328202872057320870205722070620560199132048219871202732037020375
202252031520600200872076920226205111986820326202212000220584207852088920131208462026620878200362077921188195862024520701
20440203992131420684202251943120430206052114421113204412011220485207322045220798

Прошло: 78,945000 секунд.

С:\Users\Дарья\Desktop\code\2лиоавизддля62\Debug\2лиоавизддля62.exe (процесс 10144) завершил работу с кодом 0.
Чтобы автоматически закрывать консоль при остановке отладки, включите параметр "Сервис" ->"Параметры" ->"Отладка" -> "Ав
томатически закрыть консоль при остановке отладки".
Нажмите любую клавишу, чтобы закрыть это окно...
```

Матрицы А и В размером 1000 × 1000:

```
Консоль отладки Microsoft Visual Studio

5105252581518055111552371512244523395180850824523065116851576515385214051842519305270250787519305217452221514335087852350
515175118951080518775187652535506625194152668522635129051733518055193751969521225192251811510425097651585513155193251513
516635155052432505905211151639522715206151848524915134251305516195212251919525845226052356522795228051642519945206552217
524205038251559519245128752929532685107251903518515193152182502885139451179518705057951492513825092851935515045154551494
514275195252278516935130551393518385176451860516215308951263513515103151760526335176052568511945096251123516705273951462
523265218950757518555273751655515975232251411513525247551967515595226552053516755224852226514105196251906522475226851715
51970517845318151358508965237751849517125124952132520251828515955150853148512545179852661516665224852129512475169852466
5172351231514451522735127552216528775185851846522985044252348514955264051244517705124351405519615144350821524845147051883
518595080952122529435163552851521755156751055523565147651161526345245352288526405193651647513345177851654513885141351986
517065158651633508305223950758519115246452461518905231852893519885189152678523755150851973513545097851577523465125452038
514085223952392513985212652916516255210551100525195261451819514715199151282517365209551761513245163951137519295088151188
526535230551527513385148751057508705129451353511915174551667513405220851550508835124952303511895127051477521515139651620
521855135152178513585328151107525045175552060521415137351650522915274852352522665185651997519565517251195533425274551920
51863515035229652200518765189652802518305120851685521565096515305167451896523785224150960524265112052590517925190751051
519765218450760513635129951524516855131451841518925214553273520515197851734529125226451822525485133151204513445140051656
514795225152045522695139352222516175280751219524255197052262515825131152313519705185151583517045248552269509275050452423
502985070451605519055085051288510595117752356519975170752105212635196351903531155147251771521325283351674520815057351476
519775212951116516705197651916517975186452565512325164851458513195180352440520265264752970519635171052612524105159551872
50304510305242051846519315186751913510625160051633511605114051204508705131752139518115231152134523075173520195082651887
517455190652074519205249651592524455264650502522675191651999523505209252053526745139151713516315210251390513275168351094
521375255352307512225231252292525465143451402524395215852098523415250251928520465182252311511675179551358504255207651351
52219518275208151721509185152751994522925200050938513885169851997520545199951182

Прошло: 729,282000 секунд.

С:\Users\Дарья\Desktop\code\2лиоавизддля62\Debug\2лиоавизддля62.exe (процесс 2796) завершил работу с кодом 0.
Чтобы автоматически закрывать консоль при остановке отладки, включите параметр "Сервис" ->"Параметры" ->"Отладка" -> "Ав
томатически закрыть консоль при остановке отладки".
Нажмите любую клавишу, чтобы закрыть это окно...
```

Матрицы А и В размером 2000 × 2000:

```
Консоль отладки Microsoft Visual Studio

102974105807104407104989105009105279104163103968104052104902104527103639103942104875103796104290104988102489103247105091
104989104559103092103663104857104362104314103791103096104444103877104623104215105075104225104283104169104371044663105009
10340710412510431010520610389010461010408710480310378710403710552410393610401403105022104117104021105160103601104712104278
104748105131305403104582104505105277103908104639104479103355103385103819105426102643103594105130103776103631105099103301
104500104239103732106705103550104158104752103884103550105041104429103425103931104067103805103760105614104883104681104419
105157105661104580104675105966104786103566104370103739105509104441106065104645103832103215103461104421104099105607103889
10491610437410486910484810581710443610347610439810440610288410609010402210409310471210509210425210449710604410373410343
104034105027104847105042105079103507105278105613105630103544103669105788104170103205105085104456104878103870104807104472
105445103684103806105493104293103593104150104772104854103042104807104879104710104330104377104864104366104949104457104871
105473105395105281103666104739104600104249104374104889105287102310103034104038104202104779105013105079104441104433103487
102735105971035651105781104013104319105380104700105288104371104487105196104266104002104422104807104642105277104126104332
103034105715104903105053103752105896105089104947105368105123104390103811105130105211104636103024104686104686104483103707
1045001044641038621038001047411027481039001059411053701035616105390103765103235104437102860104819104266104333104285104418
105170104565103057105428104639104608103914103318104186105514104044105578104953104412103938104430102864103573103651105672
10374910367910476910409710454910440210416610308710423210426103914104608103431104822103444103860103859105560103991104435
103347104931104769105259104363104112104612104086104421102356105728104504104095103966104544104759103966105368104026103109
1040911042671055571043321034221039881049721051661052751038511056112105912104763103796105088105737105005105057103637104445
1038391053081034010476310428910363310381210444110364104603103915104366104822105054104465104282103689104522103929104509
104249104047104213104661104573105151105294102889105472104765104441104843104943105092103443103522104450105375104431104243
1046281056811040381037831048171048031052291058701050783842105079104097104180103224104144104540104062104656104211105842
1044181053411036011045910407910375610443510336010591103532102708104683104885104945104035104208103689104399105364103433
104108103953104406103751104506104056103595104372105232106314104185104746104580104091105392104566102858106182104636106510

Прошло: 2534,775000 секунд.

С:\Users\Дарья\Desktop\code\2лиоавизддля62\Debug\2лиоавизддля62.exe (процесс 20664) завершил работу с кодом 0.
Чтобы автоматически закрывать консоль при остановке отладки, включите параметр "Сервис" ->"Параметры" ->"Отладка" -> "Ав
томатически закрыть консоль при остановке отладки".
Нажмите любую клавишу, чтобы закрыть это окно...
```


Матрицы А и В размером 4000 × 4000:

```
Консоль отладки Microsoft Visual Studio
633976415764862635976369964780644856328564982633756472864766643646424664872635786625364168642676525763994648016346864225
655516377963281637786327563315633386560063646649356389663262645956343764380643536585263132649526503865376645296385763706
655706430263814636036438263335638856444762982640816501863374643076388464579645926593264596632596418664528653306468564601
645916481164182638616400964159632606427963713644736375464245659646508264907648386559965026639006406064339642516478564524
641676489263831651006458563649646686457863553654036367764400648936516463824642506429264018640886486764951633356530665302
637556533765434643756394664681636306516464008648086444964289646166491564477647546472963137636836477064534635156473564323
647996348663729643996470764953661426387864190652116363963465629196441664447628426445063380647586345564858645616426765238
639646378564159652316411265017642016440764881643626460864749637896425663494644606485963260634646341863718648906469564768
638076484764322656766415563717644716398464385644456398064206657516483064475640606415164017650286591664628659976448763646
6336364633635655766437064907645256437764680650246397164214649676284363398640016312563660656136486364528640006311764611
647516471163948642376488464229636886441863044640556372563562636296411864247651366517264237637496510164184640496512763416
6523564443640926352864697643046488965262633786335563660649586452964418648986391563694643606530263919644363642396559164681
651016318964258649226429764413656206495264321643716326163928636706497964353645326481764811647566396965365638366448364853
633496461063963647796495764099645486402763200642216545264464647076417963963633686289064143648726409064946642206404964678
640926497763797641496371864810644626365864390648566563764385646266402465027636716456863939645736445964615650786463164692
646976530665106631386440064741648046411964204652886493564177646616308964351651806316964652644286562564287640786411665898
645386345664722636026397064305650316434564209644536476864014639306401363931647926469264850649926450164070644686354164522
642506326764007644036458464443649276508363251639006431063685634816395064732646536345664842641596392965761648016452664312
642166339363541647676447563640648176487664661640656463864137634086400564935644006484063682639166440764230638156390064615
640016318964258649226429764413656206495264321643716326163928636706497964353645326481764811647566396965365638366448364853
639856365564921650726428963988652746425563540650726492765337645306424663884633256488463370648576510664703637516404564489
646276324862795641646378164287640816443164505646016449064373642016459664076640716508064998645486435164375641396513964548

Прошло: 7378,439000 секунд.

C:\Users\Дарья\Desktop\code\2\лиоавиздлляб2\Debug\2\лиоавиздлляб2.exe (процесс 12536) завершил работу с кодом 0.
Чтобы автоматически закрывать консоль при остановке отладки, включите параметр "Сервис" -> "Параметры" -> "Отладка" -> "Автоматически закрывать консоль при остановке отладки".
Нажмите любую клавишу, чтобы закрыть это окно...
```

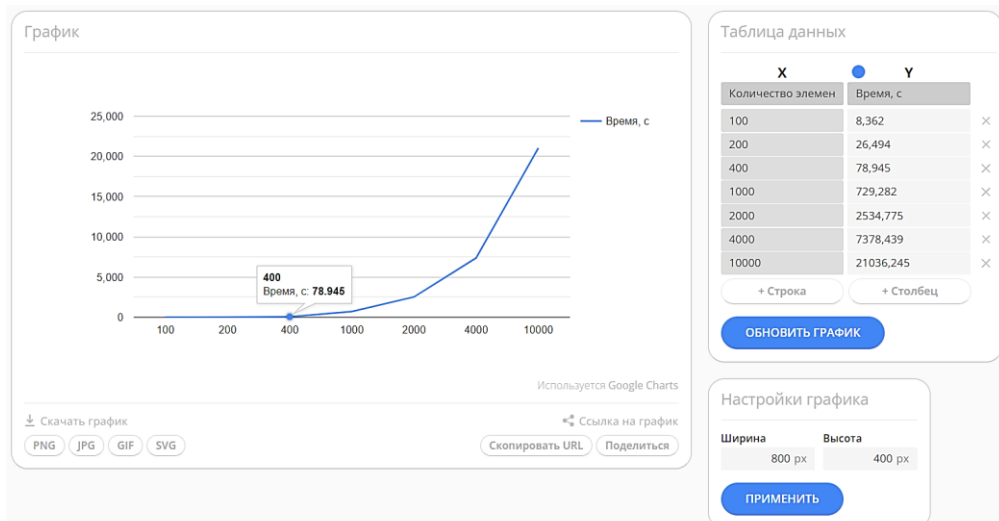
Матрицы А и В размером 10000 × 10000:

```
Консоль отладки Microsoft Visual Studio
633976415764862635976369964780644856328564982633756472864766643646424664872635786625364168642676525763994648016346864225
655516377963281637786327563315633386560063646649356389663262645956343764380643536585263132649526503865376645296385763706
655706430263814636036438263335638856444762982640816501863374643076388464579645926593264596632596418664528653306468564601
645916481164182638616400964159632606427963713644736375464245659646508264907648386559965026639006406064339642516478564524
641676489263831651006458563649646686457863553654036367764400648936516463824642506429264018640886486764951633356530665302
637556533765434643756394664681636306516464008648086444964289646166491564477647546472963137636836477064534635156473564323
647996348663729643996470764953661426387864190652116363963465629196441664447628426445063380647586345564858645616426765238
639646378564159652316411265017642016440764881643626460864749637896425663494644606485963260634646341863718648906469564768
638076484764322656766415563717644716398464385644456398064206657516483064475640606415164017650286591664628659976448763646
6336364633635655766437064907645256437764680650246397164214649676284363398640016312563660656136486364528640006311764611
647516471163948642376488464229636886441863044640556372563562636296411864247651366517264237637496510164184640496512763416
6523564443640926352864697643046488965262633786335563660649586452964418648986391563694643606530263919644363642396559164681
651016318964258649226429764413656206495264321643716326163928636706497964353645326481764811647566396965365638366448364853
633496461063963647796495764099645486402763200642216545264464647076417963963633686289064143648726409064946642206404964678
640926497763797641496371864810644626365864390648566563764385646266402465027636716456863939645736445964615650786463164692
646976530665106631386440064741648046411964204652886493564177646616308964351651806316964652644286562564287640786411665898
645386345664722636026397064305650316434564209644536476864014639306401363931647926469264850649926450164070644686354164522
642506326764007644036458464443649276508363251639006431063685634816395064732646536345664842641596392965761648016452664312
642166339363541647676447563640648176487664661640656463864137634086400564935644006484063682639166440764230638156390064615
640016318964258649226429764413656206495264321643716326163928636706497964353645326481764811647566396965365638366448364853
639856365564921650726428963988652746425563540650726492765337645306424663884633256488463370648576510664703637516404564489
646276324862795641646378164287640816443164505646016449064373642016459664076640716508064998645486435164375641396513964548

Прошло: 21036,245000 секунд.

C:\Users\Дарья\Desktop\code\2\лиоавиздлляб2\Debug\2\лиоавиздлляб2.exe (процесс 12536) завершил работу с кодом 0.
Чтобы автоматически закрывать консоль при остановке отладки, включите параметр "Сервис" -> "Параметры" -> "Отладка" -> "Автоматически закрывать консоль при остановке отладки".
Нажмите любую клавишу, чтобы закрыть это окно...
```

3. Построить график зависимости времени выполнения программы от размера матриц и сравнить полученный результат с теоретической оценкой.



Задание 2:

Сортировка Шелла:

```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <iostream>
#include <locale>

// Функция сортировки Шелла
void shell(int* items, int count)
{
    int x, a[8];
    int i, j, gap, k;
    a[0] = 3; a[1] = 7; a[2] = 19; a[3] = 35; a[4] = 64; a[5] = 91; a[6] = 128; a[7] = 547;

    for (k = 0; k < 8; k++) {
        gap = a[k];
        for (i = gap; i < count; ++i) {
            x = items[i];
            for (j = i - gap; (x < items[j]) && (j >= 0); j = j - gap)
                items[j + gap] = items[j];
            items[j + gap] = x;
        }
    }
}

int main()
{
    setlocale(LC_ALL, "rus");

    setvbuf(stdin, NULL, _IONBF, 0);
    setvbuf(stdout, NULL, _IONBF, 0);

    time_t start, end;

    int a[8];
    a[0] = 3; a[1] = 7; a[2] = 19; a[3] = 35; a[4] = 64; a[5] = 91; a[6] = 128; a[7] = 547;

    start = time(NULL);
    printf("Исходный массив: ");
    for (int i = 0; i < 8; i++)
    {
        printf("%2d ", a[i]);
    }

    shell(a, 8);

    printf("\nПолученный массив: ");
    for (int i = 0; i < 8; i++)
    {
        printf("%2d ", a[i]);
    }
    end = time(NULL);
    printf("\nВремя работы программы: %lf секунд.\n", clock() / 1000.0);

    return 0;
}
```


Быстрая сортировка:

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <iostream>
#include <locale>

using namespace std;

void qs(int* items, int left, int right) //вызов функции: qs(items, 0, count-1);
{
    int i, j;
    int x, y;

    i = left; j = right;

    /* выбор компаранда */
    x = items[(left + right) / 2];

    do {
        while ((items[i] < x) && (i < right)) i++;
        while ((x < items[j]) && (j > left)) j--;

        if (i <= j) {
            y = items[i];
            items[i] = items[j];
            items[j] = y;
            i++; j--;
        }
    } while (i <= j);

    if (left < j) qs(items, left, j);
    if (i < right) qs(items, i, right);
}

int main()
{
    setlocale(LC_ALL, "rus");
    srand(time(NULL));

    int count = 8;
    int a[] = {7, 35, 91, 547, 128, 64, 19, 3};

    printf("Исходный массив:\n");
    for (int i = 0; i < count; i++)
    {
        printf("%2d ", a[i]);
    }

    int first = 0, last = count - 1;
    qs(a, first, last);

    printf("\n\nРезультат:\n");
    for (int i = 0; i < count; i++)
    {
        printf("%2d ", a[i]);
    }
    printf("\n\nВремя работы программы: %lf секунд.\n", clock() / 1000.0);

    return 0;
}
```

1. Оценить время работы каждого из реализованных алгоритмов на случайном наборе значений массива.

Сортировка Шелла:

Консоль отладки Microsoft Visual Studio

```
Исходный массив: 367 291 38 112 64 3 0 279
Полученный массив: 0 64 3 112 279 38 367 291
Время работы программы: 0,064000 секунд.
```

Быстрая сортировка:

Консоль отладки Microsoft Visual Studio

```
Исходный массив:
367 291 38 112 64 3 0 279

Результат:
0 3 38 64 112 279 291 367

Время работы программы: 0,132000 секунд.
```

2. Оценить время работы каждого из реализованных алгоритмов на массиве, представляющем собой возрастающую последовательность чисел.

Сортировка Шелла:

Консоль отладки Microsoft Visual Studio

```
Исходный массив: 3 7 19 35 64 91 128 547
Полученный массив: 3 7 19 35 64 91 128 547
Время работы программы: 0,083000 секунд.
```

Быстрая сортировка:

Консоль отладки Microsoft Visual Studio

```
Исходный массив:
3 7 19 35 64 91 128 547

Результат:
3 7 19 35 64 91 128 547

Время работы программы: 0,198000 секунд.
```

3. Оценить время работы каждого из реализованных алгоритмов на массиве, представляющем собой убывающую последовательность чисел.

Сортировка Шелла:

```
Консоль отладки Microsoft Visual Studio
Исходный массив: 547 128 91 64 35 19 7 3
Полученный массив: 3 35 19 7 128 91 64 547
Время работы программы: 0,078000 секунд.
```

Быстрая сортировка:

```
Консоль отладки Microsoft Visual Studio
Исходный массив:
547 128 91 64 35 19 7 3

Результат:
3 7 19 35 64 91 128 547

Время работы программы: 0,201000 секунд.
```

4. Оценить время работы каждого из реализованных алгоритмов на массиве, одна половина которого представляет собой возрастающую последовательность чисел, а вторая, – убывающую.

Сортировка Шелла:

```
Консоль отладки Microsoft Visual Studio
Исходный массив: 7 35 91 547 128 64 19 3
Полученный массив: 3 7 64 19 35 91 547 128
Время работы программы: 0,081000 секунд.
```

Быстрая сортировка:

```
Консоль отладки Microsoft Visual Studio
Исходный массив:
7 35 91 547 128 64 19 3

Результат:
3 7 19 35 64 91 128 547

Время работы программы: 0,097000 секунд.
```

5. Оценить время работы стандартной функции qsort, реализующей алгоритм быстрой сортировки на выше указанных наборах данных.

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <iostream>
#include <locale>

using namespace std;

int a[] = { 3,7,19,35,64,91,128,547 };
int compare(const void* x1, const void* x2) // функция сравнения элементов массива
{
    return (*(int*)x1 - *(int*)x2);
}
int main()
{
    setlocale(LC_ALL, "rus");
    srand(time(NULL)); // инициализируем параметры генератора случайных чисел

    int count = 8;

    printf("Исходный массив: ");
    for (int i = 0; i < count; i++)
    {
        printf("%2d ", a[i]);
    }

    printf("\n\nРезультат: ");
    int first = 0, last = count - 1;
    qsort(a, count, sizeof(int), compare);
    for (int i = 0; i < count; i++)
    {
        printf("%2d ", a[i]);
    }
    printf("\n\nВремя работы программы: %lf секунд.\n", clock() / 1000.0);

    return 0;
}
```

Возрастающая последовательность чисел:

```
cs Консоль отладки Microsoft Visual Studio
Исходный массив: 3 7 19 35 64 91 128 547
Результат: 3 7 19 35 64 91 128 547
Время работы программы: 0,064000 секунд.
```

Убывающая последовательность чисел:

```
cs Консоль отладки Microsoft Visual Studio
Исходный массив: 547 128 91 64 35 19 7 3
Результат: 3 7 19 35 64 91 128 547
Время работы программы: 0,067000 секунд.
```

Треугольная последовательность чисел:

```
cs Консоль отладки Microsoft Visual Studio
Исходный массив: 7 35 91 547 128 64 19 3
Результат: 3 7 19 35 64 91 128 547
Время работы программы: 0,066000 секунд.
```


Запись результатов в файл в виде таблицы:

```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <iostream>
#include <locale>

// ФУНКЦИЯ СОРТИРОВКИ ШЕЛЛА
void shell(int* items1, int count1)
{
    int x1;
    int i1, j1, gap, k;
    int a[8];
    a[0] = 3; a[1] = 7; a[2] = 19; a[3] = 35; a[4] = 64; a[5] = 91; a[6] = 128; a[7] = 547;
    int a1[8];
    a1[0] = 547; a1[1] = 128; a1[2] = 91; a1[3] = 64; a1[4] = 35; a1[5] = 19; a1[6] = 7; a1[7]
= 3;
    int a2[8];
    a2[0] = 7; a2[1] = 35; a2[2] = 91; a2[3] = 547; a2[4] = 128; a2[5] = 64; a2[6] = 19; a2[7]
= 3;

    for (k = 0; k < 8; k++) {
        gap = a[k];
        for (i1 = gap; i1 < count1; ++i1) {
            x1 = items1[i1];
            for (j1 = i1 - gap; (x1 < items1[j1]) && (j1 >= 0); j1 = j1 - gap)
                items1[j1 + gap] = items1[j1];
            items1[j1 + gap] = x1;
        }
    }
}

// ФУНКЦИЯ БЫСТРОЙ СОРТИРОВКИ
void qs(int* items2, int left, int right) //вызов функции: qs(items, 0, count-1);
{
    int a[8];
    a[0] = 3; a[1] = 7; a[2] = 19; a[3] = 35; a[4] = 64; a[5] = 91; a[6] = 128; a[7] = 547;
    int a1[8];
    a1[0] = 547; a1[1] = 128; a1[2] = 91; a1[3] = 64; a1[4] = 35; a1[5] = 19; a1[6] = 7; a1[7]
= 3;
    int a2[8];
    a2[0] = 7; a2[1] = 35; a2[2] = 91; a2[3] = 547; a2[4] = 128; a2[5] = 64; a2[6] = 19; a2[7]
= 3;

    int i2, j2;
    int x2, y2;

    i2 = left; j2 = right;

    /* выбор компаранда */
    x2 = items2[(left + right) / 2];

    do {
        while ((items2[i2] < x2) && (i2 < right)) i2++;
        while ((x2 < items2[j2]) && (j2 > left)) j2--;

        if (i2 <= j2) {
            y2 = items2[i2];
            items2[i2] = items2[j2];
            items2[j2] = y2;
            i2++; j2--;
        }
    } while (i2 < j2);
}
```

```

    }
} while (i2 <= j2);

if (left < j2) qs(items2, left, j2);
if (i2 < right) qs(items2, i2, right);
}

// ФУНКЦИЯ QSORT
int aq[] = { 3,7,19,35,64,91,128,547 };
int a1q[] = { 547,128,91,64,35,19,7,3 };
int a2q[] = { 7,35,91,547,128,64,19,3 };
int compare(const void* x3_1, const void* x3_2) // функция сравнения элементов массива
{
    return (*(int*)x3_1 - *(int*)x3_2);
}

int main()
{
    setlocale(LC_ALL, "rus");

    setvbuf(stdin, NULL, _IONBF, 0);
    setvbuf(stdout, NULL, _IONBF, 0);

    char inputFile[255];
    FILE* input;

    int a[8];
    a[0] = 3; a[1] = 7; a[2] = 19; a[3] = 35; a[4] = 64; a[5] = 91; a[6] = 128; a[7] = 547;
    int a1[8];
    a1[0] = 547; a1[1] = 128; a1[2] = 91; a1[3] = 64; a1[4] = 35; a1[5] = 19; a1[6] = 7; a1[7]
= 3;
    int a2[8];
    a2[0] = 7; a2[1] = 35; a2[2] = 91; a2[3] = 547; a2[4] = 128; a2[5] = 64; a2[6] = 19; a2[7]
= 3;

    /*-----*/
    do {
        printf("Введите имя файла, в который необходимо записать результаты сравнения времени
работы всех сортировок: ");
        scanf("%s", inputFile);

        input = fopen(inputFile, "w");

        if (input != NULL)
            break;

        printf("Ошибка! Не удалось открыть файл! Повторите ввод\n\n");
    } while (true);
    /*-----*/

    //-----
    printf("\n\n1.1 Сортировка Шелла:\n");
    clock_t start11, end11;
    start11 = clock();
    printf("Исходный массив при возрастающей последовательности: ");
    for (int i = 0; i < 8; i++)
    {
        printf("%2d ", a[i]);
    }
    shell(a, 8);
    printf("\nПолученный массив: ");
    for (int i = 0; i < 8; i++)
    {
        printf("%2d ", a[i]);
    }

```

```

}
end11 = clock();
double extime11 = (double)(end11 - start11) / clock();
printf("\n\tВремя работы программы: %lf секунд.\n\n", extime11);
//-----

//-----
printf("\n\n1.2 Сортировка Шелла:\n");
clock_t start12, end12;
start12 = clock();
printf("Исходный массив при убывающей последовательности: ");
for (int i = 0; i < 8; i++)
{
    printf("%2d ", a1[i]);
}
shell(a1, 8);
printf("\nПолученный массив: ");
for (int i = 0; i < 8; i++)
{
    printf("%2d ", a[i]);
}
end12 = clock();
double extime12 = (double)(end12 - start12) / clock();
printf("\n\tВремя работы программы: %lf секунд.\n\n", extime12);
//-----

//-----
printf("\n\n1.3 Сортировка Шелла:\n");
clock_t start13, end13;
start13 = clock();
printf("Исходный массив при треугольной последовательности: ");
for (int i = 0; i < 8; i++)
{
    printf("%2d ", a2[i]);
}
shell(a2, 8);
printf("\nПолученный массив: ");
for (int i = 0; i < 8; i++)
{
    printf("%2d ", a[i]);
}
end13 = clock();
double extime13 = (double)(end13 - start13) / clock();
printf("\n\tВремя работы программы: %lf секунд.\n\n", extime13);
//-----

int first1 = 0, last1 = 8 - 1;
//-----
printf("\n\n2.1 Быстрая сортировка:\n");
clock_t start21, end21;
start21 = clock();
printf("Исходный массив при возрастающей последовательности: : ");
for (int i = 0; i < 8; i++)
{
    printf("%2d ", a[i]);
}
qs(a, first1, last1);
printf("\nПолученный массив: ");
for (int i = 0; i < 8; i++)
{
    printf("%2d ", a[i]);
}
end21 = clock();
double extime21 = (double)(end21 - start21) / clock();
printf("\n\tВремя работы программы: %lf секунд.\n\n", extime21);

```

```

//-----

//-----
printf("\n\n2.2 Быстрая сортировка:\n");
clock_t start22, end22;
start22 = clock();
printf("Исходный массив при убывающей последовательности: ");
for (int i = 0; i < 8; i++)
{
    printf("%2d ", a1[i]);
}
qs(a1, first1, last1);
printf("\nПолученный массив: ");
for (int i = 0; i < 8; i++)
{
    printf("%2d ", a1[i]);
}
end22 = clock();
double extime22 = (double)(end22 - start22) / clock();
printf("\n\tВремя работы программы: %lf секунд.\n\n", extime22);
//-----

//-----
printf("\n\n2.3 Быстрая сортировка: \n");
clock_t start23, end23;
start23 = clock();
printf("Исходный массив при треугольной последовательности: ");
for (int i = 0; i < 8; i++)
{
    printf("%2d ", a2[i]);
}
qs(a2, first1, last1);
printf("\nПолученный массив: ");
for (int i = 0; i < 8; i++)
{
    printf("%2d ", a2[i]);
}
end23 = clock();
double extime23 = (double)(end23 - start23) / clock();
printf("\n\tВремя работы программы: %lf секунд.\n\n", extime23);
//-----

int count2 = 8;
int first2 = 0, last2 = count2 - 1;
//-----
printf("\n\n3.1 Сортировка с использованием функции qsort:\n");
clock_t start31, end31;
start31 = clock();
printf("Исходный массив при возрастающей последовательности: ");
for (int i = 0; i < 8; i++)
{
    printf("%2d ", aq[i]);
}
qsort(aq, count2, sizeof(int), compare);
printf("\nПолученный массив: ");
for (int i = 0; i < 8; i++)
{
    printf("%2d ", aq[i]);
}
end31 = clock();
double extime31 = (double)(end31 - start31) / clock();
printf("\n\tВремя работы программы: %lf секунд.\n\n", extime31);
//-----

```

```

//-----
printf("\n\n3.2 Сортировка с использованием функции qsort:\n");
clock_t start32, end32;
start32 = clock();
printf("Исходный массив при убывающей последовательности: ");
for (int i = 0; i < 8; i++)
{
    printf("%2d ", a1q[i]);
}
qsort(a1q, count2, sizeof(int), compare);
printf("\nПолученный массив: ");
for (int i = 0; i < 8; i++)
{
    printf("%2d ", a1q[i]);
}
end32 = clock();
double extime32 = (double)(end32 - start32) / clock();
printf("\n\tВремя работы программы: %lf секунд.\n\n", extime32);
//-----

//-----
printf("\n\n3.3 Сортировка с использованием функции qsort:\n");
clock_t start33, end33;
start33 = clock();
printf("Исходный массив при треугольной последовательности: ");
for (int i = 0; i < 8; i++)
{
    printf("%2d ", a2q[i]);
}
qsort(a2q, count2, sizeof(int), compare);
printf("\nПолученный массив: ");
for (int i = 0; i < 8; i++)
{
    printf("%2d ", a2q[i]);
}
end33 = clock();
double extime33 = (double)(end33 - start33) / clock();
printf("\n\tВремя работы программы: %lf секунд.\n\n", extime33);
//-----

fprintf(input, "Результат:\n\n");

fprintf(input, " -----
-----\n");
fprintf(input, " |                | Сортировка Шелла | Быстрая Сортировка | Функция
qsort  |\n");
fprintf(input, " |-----|-----|-----|-----|
-----|\n");
fprintf(input, " | Возраст. послед. |      %lf с      |      %lf с      |      %lf с      |\n",
extime11, extime21, extime31);
fprintf(input, " |-----|-----|-----|-----|
-----|\n");
fprintf(input, " | Убывающ. послед. |      %lf с      |      %lf с      |      %lf с      |\n",
extime12, extime22, extime32);
fprintf(input, " |-----|-----|-----|-----|
-----|\n");
fprintf(input, " | Треугол. послед. |      %lf с      |      %lf с      |      %lf с      |\n",
extime13, extime23, extime33);
fprintf(input, " -----
-----\n");

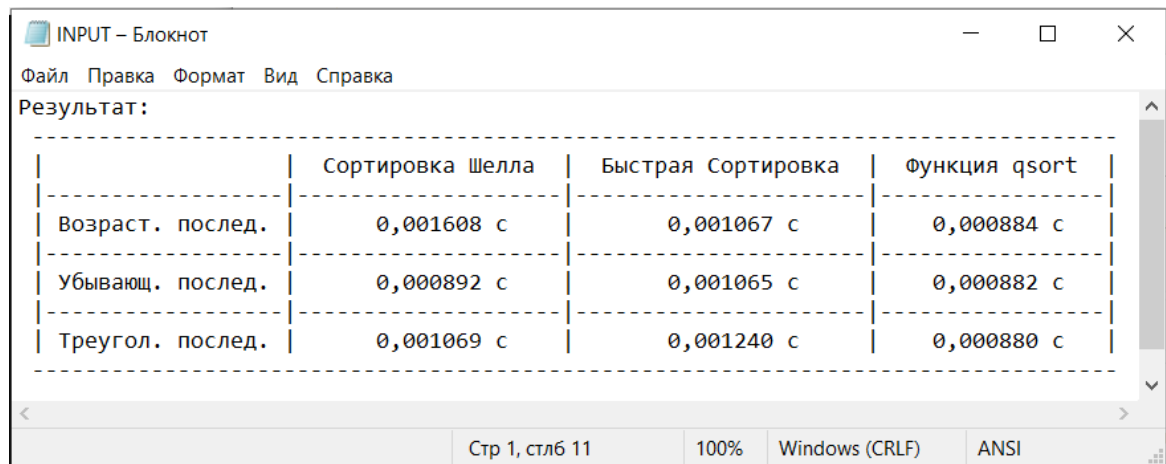
fclose(input);

return 0;
}

```


Приложение к заданию 2:

	Сортировка Шелла	Быстрая сортировка	Функция qsort
Возрастающая	0,083 с	0,198 с	0,064 с
Убывающая	0,078 с	0,201 с	0,067 с
Треугольная	0,081 с	0,097 с	0,066 с



INPUT – Блокнот

Файл Правка Формат Вид Справка

Результат:

	Сортировка Шелла	Быстрая Сортировка	Функция qsort
Возраст. послед.	0,001608 с	0,001067 с	0,000884 с
Убывающ. послед.	0,000892 с	0,001065 с	0,000882 с
Треугол. послед.	0,001069 с	0,001240 с	0,000880 с

Стр 1, столб 11 100% Windows (CRLF) ANSI

Вывод: мы научились применять средства, предоставляемые библиотекой `time.h`. Научились определять сложность алгоритма, рассмотрели зависимость времени от количества элементов в массиве с помощью графика. Пронаблюдали, что алгоритм быстрой сортировки работает медленнее, чем алгоритм Шелла, но самым быстрым в выполнении показала функция `qsort`.