Пензенский государственный университет

Кафедра «Вычислительная техника»

**ОТЧЕТ**

по лабораторной работе №2

# по курсу «Логика и основы алгоритмизации в инженерных задачах»

на тему «Оценка времени выполнения программ»

**Выполнили:**

студенты группы 20ВВ3

Кузин Д.Д.

**Приняли:**

Митрохин М.А.

Пенза 2022

**Цель работы** – изучение библиотеки time.h для оценки времени выполнения программы.

# **Лабораторное задание**

**Задание 1:**

1. Вычислить порядок сложности программы (О-символику).
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.
3. Построить график зависимости времени выполнения программы от размера матриц и сравнить полученный результат с теоретической оценкой.

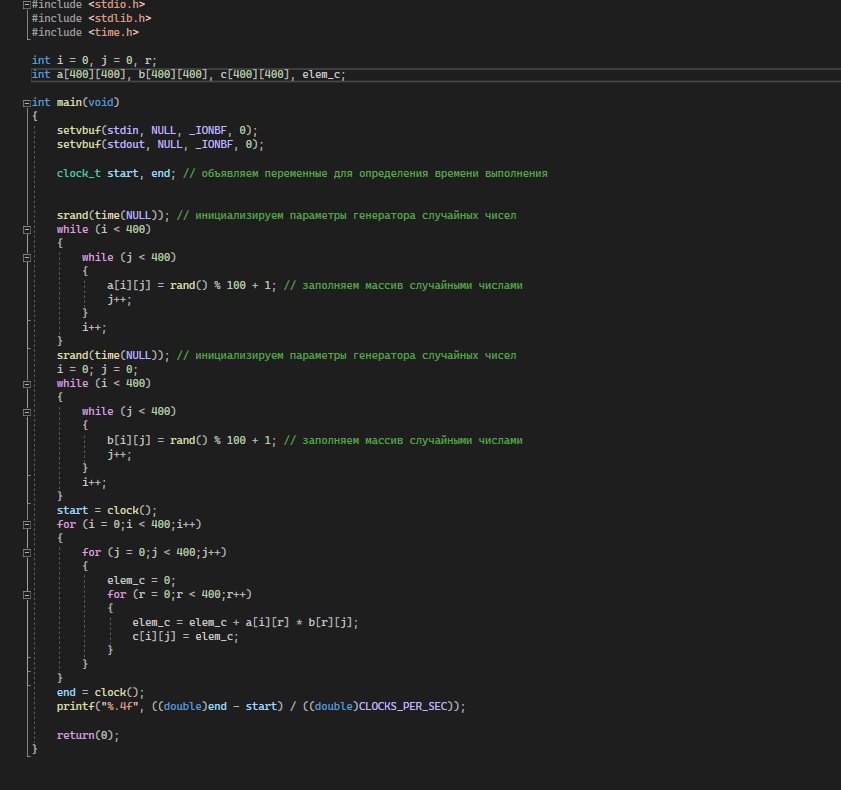
**Задание 2:**

1. Оценить время работы каждого из реализованных алгоритмов на случайном наборе значений массива.
2. Оценить время работы каждого из реализованных алгоритмов на массиве, представляющем собой возрастающую последовательность чисел.
3. Оценить время работы каждого из реализованных алгоритмов на массиве, представляющем собой убывающую последовательность чисел.
4. Оценить время работы каждого из реализованных алгоритмов на массиве, одна половина которого представляет собой возрастающую последовательность чисел, а вторая, – убывающую.

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

**Ход работы**

**Задание 1:**

****

**Листинг**

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

int i = 0, j = 0, r;

int a[400][400], b[400][400], c[400][400], elem\_c;

int main(void)

{

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

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

clock\_t start, end; // объявляем переменные для определения времени выполнения

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

while (i < 400)

{

while (j < 400)

{

a[i][j] = rand() % 100 + 1; // заполняем массив случайными числами

j++;

}

i++;

}

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

i = 0; j = 0;

while (i < 400)

{

while (j < 400)

{

b[i][j] = rand() % 100 + 1; // заполняем массив случайными числами

j++;

}

i++;

}

start = clock();

for (i = 0;i < 400;i++)

{

for (j = 0;j < 400;j++)

{

elem\_c = 0;

for (r = 0;r < 400;r++)

{

elem\_c = elem\_c + a[i][r] \* b[r][j];

c[i][j] = elem\_c;

}

}

}

end = clock();

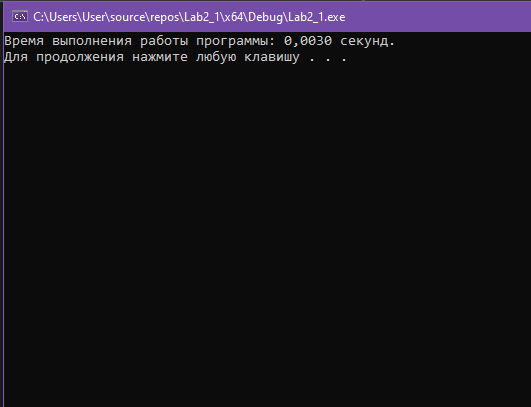
printf("%.4f", ((double)end - start) / ((double)CLOCKS\_PER\_SEC));

return(0);

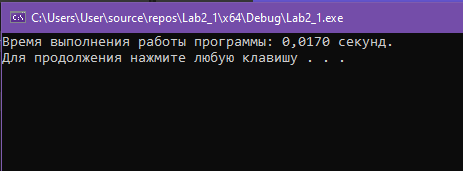
}

**Результат работы программы**

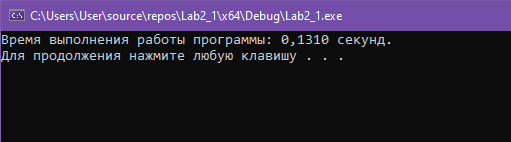
Матрица размером – 100:

****

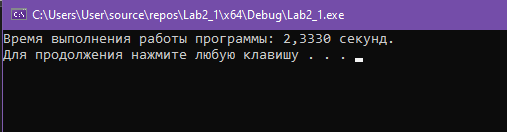
Матрица размером – 200:

****

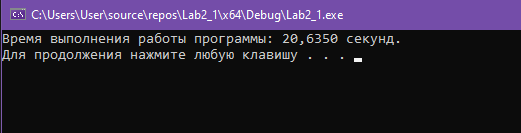
Матрица размером – 400:

****

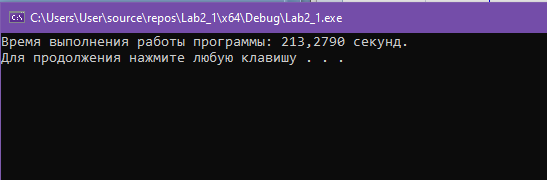
Матрица размером – 1000:



Матрица размером – 2000:

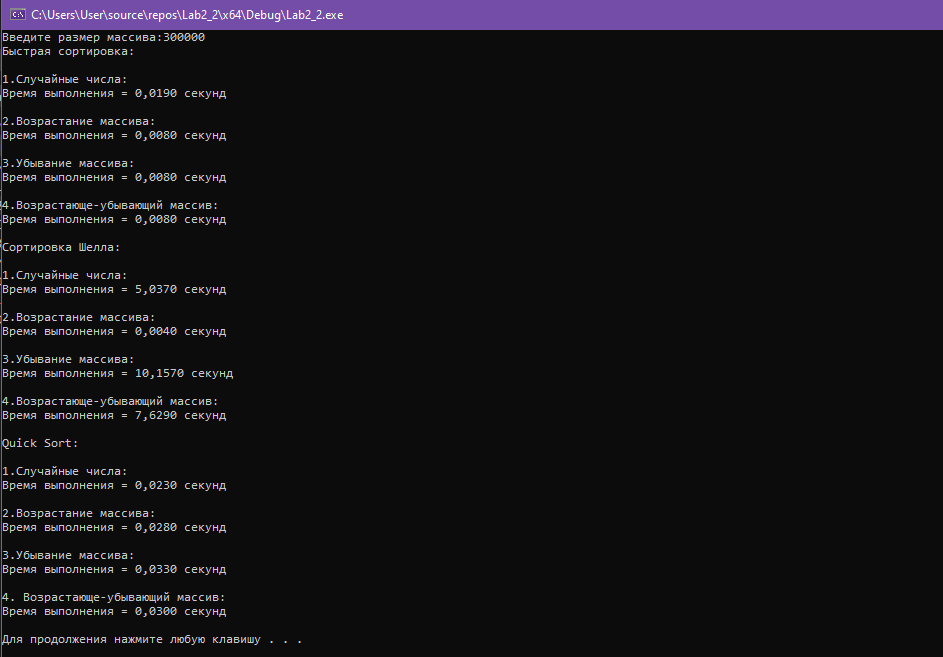
****

Матрица размером – 4000:



**График зависимости времени выполнения программы от размера матриц**

**Задание 2:**

******

**Листинг:**#include <stdio.h>

#include <stdlib.h>

#include <time.h>

#include <locale.h>

void shell(int\* items, int count)

{

int i, j, gap, k;

int x, a[5];

a[0] = 9; a[1] = 5; a[2] = 3; a[3] = 2; a[4] = 1;

for (k = 0; k < 5; 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;

}

}

}

void qs(int\* array1, int left, int right)

{

int i, j;

int x, y;

i = left; j = right;

/\* выбор компаранда \*/

x = array1[(left + right) / 2];

do {

while ((array1[i] < x) && (i < right)) i++;

while ((x < array1[j]) && (j > left)) j--;

if (i <= j) {

y = array1[i];

array1[i] = array1[j];

array1[j] = y;

i++; j--;

}

} while (i <= j);

if (left < j) qs(array1, left, j);

if (i < right) qs(array1, i, right);

}

int comp(const void\* a, const void\* b)

{

return \*(int\*)a - \*(int\*)b;

}

int main(void)

{

setlocale(LC\_ALL,"Russian");

clock\_t start, end; // объявляем переменные для определения времени выполнения

start = clock();

int i = 0;

int size;

printf("Введите размер массива:");

scanf\_s("%d", &size);

int\* array1 = (int\*)malloc(size \* sizeof(int));

srand(time(NULL));

printf("Быстрая сортировка:\n");

printf("\n");

printf("1.Случайные числа:\n");

for (int i = 0; i < size; i++)

{

array1[i] = rand() % 100;

}

start = clock();

qs(array1, 0, size - 1);

end = clock();

printf("Время выполнения = %.4f секунд\n", ((double)end - start) / ((double)CLOCKS\_PER\_SEC));

printf("\n");

printf("2.Возрастание массива:\n");

for (int i = 0; i < size; i++)

{

array1[i] = i;

}

start = clock();

qs(array1, 0, size - 1);

end = clock();

printf("Время выполнения = %.4f секунд\n", ((double)end - start) / ((double)CLOCKS\_PER\_SEC));

printf("\n");

printf("3.Убывание массива:\n");

for (int i = 0; i < size; i++)

{

array1[i] = 10-i;

}

start = clock();

qs(array1, 0, size - 1);

end = clock();

printf("Время выполнения = %.4f секунд\n", ((double)end - start) / ((double)CLOCKS\_PER\_SEC));

printf("\n");

printf("4.Возрастающе-убывающий массив:\n");

for (int i = 0; i < size; i++)

{

if (i < size / 2)

array1[i] = i;

if (i >= size / 2)

array1[i] = 10 - i;

}

start = clock();

qs(array1, 0, size - 1);

end = clock();

printf("Время выполнения = %.4f секунд\n", ((double)end - start) / ((double)CLOCKS\_PER\_SEC));

printf("\n");

printf("Сортировка Шелла:\n");

printf("\n");

printf("1.Случайные числа:\n");

for (int i = 0; i < size; i++)

{

array1[i] = rand() % 100;

}

start = clock();

shell(array1, size);

end = clock();

printf("Время выполнения = %.4f секунд\n", ((double)end - start) / ((double)CLOCKS\_PER\_SEC));

printf("\n");

printf("2.Возрастание массива:\n");

for (int i = 0; i < size; i++)

{

array1[i] = i;

}

start = clock();

shell(array1, size);

end = clock();

printf("Время выполнения = %.4f секунд\n", ((double)end - start) / ((double)CLOCKS\_PER\_SEC));

printf("\n");

printf("3.Убывание массива:\n");

for (int i = 0; i < size; i++)

{

array1[i] = 10 - i;

}

start = clock();

shell(array1, size);

end = clock();

printf("Время выполнения = %.4f секунд\n", ((double)end - start) / ((double)CLOCKS\_PER\_SEC));

printf("\n");

printf("4.Возрастающе-убывающий массив:\n");

for (int i = 0; i < size; i++)

{

if (i < size / 2)

array1[i] = i;

if (i >= size / 2)

array1[i] = 10 - i;

}

start = clock();

shell(array1, size);

end = clock();

printf("Время выполнения = %.4f секунд\n", ((double)end - start) / ((double)CLOCKS\_PER\_SEC));

printf("\n");

printf("Quick Sort:\n");

printf("\n");

printf("1.Случайные числа:\n");

for (int i = 0; i < size; i++)

{

array1[i] = rand() % 100;

}

start = clock();

qsort(array1, size, sizeof(int), comp);

end = clock();

printf("Время выполнения = %.4f секунд\n", ((double)end - start) / ((double)CLOCKS\_PER\_SEC));

printf("\n");

printf("2.Возрастание массива:\n");

for (int i = 0; i < size; i++)

{

array1[i] = i;

}

start = clock();

qsort(array1, size, sizeof(int), comp);

end = clock();

printf("Время выполнения = %.4f секунд\n", ((double)end - start) / ((double)CLOCKS\_PER\_SEC));

printf("\n");

printf("3.Убывание массива:\n");

for (int i = 0; i < size; i++)

{

array1[i] = 10-i;

}

start = clock();

qsort(array1, size, sizeof(int), comp);

end = clock();

printf("Время выполнения = %.4f секунд\n", ((double)end - start) / ((double)CLOCKS\_PER\_SEC));

printf("\n");

printf("4. Возрастающе-убывающий массив:\n");

for (int i = 0; i < size; i++)

{

if (i < size / 2)

array1[i] = i;

if (i >= size / 2)

array1[i] = 10 - i;

}

start = clock();

qsort(array1, size, sizeof(int),comp);

end = clock();

printf("Время выполнения = %.4f секунд\n", ((double)end - start) / ((double)CLOCKS\_PER\_SEC));

printf("\n");

system("PAUSE");

return(0);

}

**Вывод:** изучение библиотеки time.h для оценки времени выполнения программы.