Пензенский государственный университет

Кафедра ”Вычислительной техники ”

**Отчет**

по лабораторной работе №2

по дисциплине: “ Логика и основы алгоритмизации в инженерных задачах ”

## по теме: “Оценка времени выполнения программы”

## *Выполнил студент группы 19ВВ1:*

## Васильев Л.Р

## *Приняли:*

## Митрохин М.А

ПЕНЗА 2020

**Общие сведения.**

Для оценки времени выполнения программ языка Си или их частей могут

использоваться средства, предоставляемые библиотекой time.h. Данная библиотека

содержит описания типов и прототипы функций для работы с датой и временем.

Типы данных:

1. clock\_t - возвращается функцией clock(). Обычно определён как int или long int.

2. time\_t - возвращается функцией time(). Обычно определён как int или long int.

3. struct tm - нелинейное, дискретное календарное представление времени.

Основные функции:

1. clock\_t clock(void) - возвращает время, измеряемое процессором в тактах от

начала выполнения программы, или −1, если оно не известно. Пересчет этого времени в

секунды выполняется по формуле:

clock() / CLOCKS\_PER\_SEC

где CLOCKS\_PER\_SEC – константа, определяющая количество тактов системных

часов в секунду.

2. time\_t time(time\_t \*tp)

Возвращает текущее календарное время или −1, если это время не известно. Если

указатель tp не равен NULL, то возвращаемое значение записывается также и в \*tp.

3. double difftime(time\_t time2,time\_t time1)

Возвращает разность time2-time1, выраженную в секундах.

Задание 1)

Вычислить порядок сложности программы (О-символику).

while(i<200){

while(i<200){

a[i][j] = rand()%100+1;

j++;

}

i++;

}

**Сложность O(n^2)**

while(i<200){

while(i<200){

b[i][j] = rand()%100+1;

j++;

}

i++;

}

**Сложность O(n^2)**

for(int i=0;i<200;i++){

for(int j=0;j<200;j++){

elem\_c = 0;

for(int r =0;r<200;r++){

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

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

}

}

}

**Сложность O(n^3)**

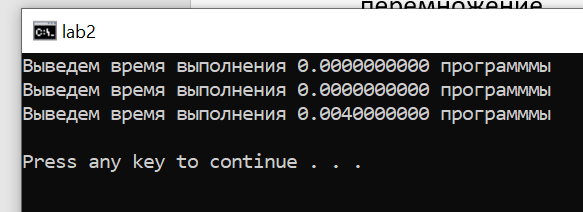
Задание 2)

Оценить время выполнения программы и кода, выполняющего перемножение

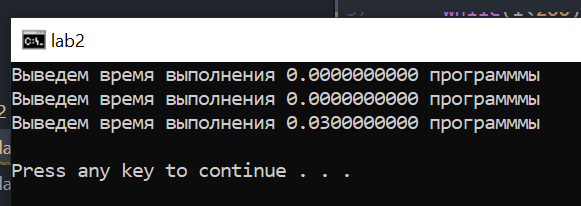
матриц, используя функции библиотеки time.h для матриц размерами от 100, 200,

400, 1000, 2000, 4000, 10000.

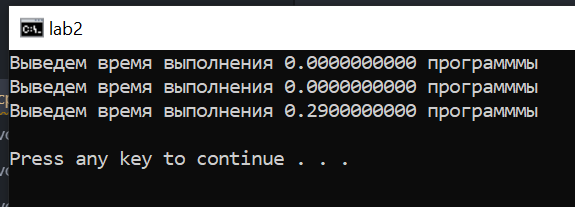
Размер 100



Размер 200



Размер 400



Задание 3)

Построить график зависимости времени выполнения программы от размера матриц

и сравнить полученный результат с теоретической оценкой.