***Отчёт по лабораторной работе №1***

***Выполнил студент группы 1-41 XX***

***Прохоров Максим***

**Содержание**

[Условие задачи 3](#_Toc114556503)

[Математическая модель 3](#_Toc114556504)

[Алгоритм решения задачи 4](#_Toc114556505)

[Тестовые примеры 5](#_Toc114556506)

# Условие задачи

*26. Быстрое возведение матрицы в степень.* Дано натуральное число *k* и действительная квадратная матрица ***A*** порядка *n*. Составить программу быстрого вычисления ***Ak***. Перемножение матриц организовать в подпрограмме.

# Математическая модель

Чтобы перемножить матрицы нужно поочередно перемножать элементы строки одной матрицы на элементы столбца второй матрицы. При четном k быстрое возведение в степень можно представить как Ak = (A2)k/2, а при нечетном как Ak = A(A2)(k-1)/2.

# Программа

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <iostream>

#include <Windows.h>

using namespace std;

float\*\* matrix\_multiplication(float\*\* A, float\*\* B, int n) {

float\*\* C = new float\* [n];

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

C[i] = new float[n];

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

float summ = 0;

for (int l = 0; l < n; l++)

summ += A[i][l] \* B[l][j];

C[i][j] = summ;

}

}

return C;

}

int main()

{

SetConsoleOutputCP(1251);

int n, k;

printf("Введите n = ");

scanf("%d", &n);

printf("Введите элементы матрицы:\n");

float\*\* A = new float\* [n], \*\* E = new float\* [n];

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

A[i] = new float[n];

E[i] = new float[n];

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

scanf("%f", &A[i][j]);

if (i == j)

E[i][j] = 1;

else

E[i][j] = 0;

}

}

printf("Введите k = ");

scanf("%d", &k);

float\*\* R = E, \*\* T = A;

while (k) {

if (k % 2 == 0) {

k /= 2;

T = matrix\_multiplication(T, T, n);

}

else {

k--;

R = matrix\_multiplication(R, T, n);

}

}

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

for (int j = 0; j < n; j++)

printf("%6.2f ", R[i][j]);

printf("\n");

}

}

# Алгоритм решения задачи

# Тестовые примеры

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 2  1 2  3 4  3 | 37.00 54.00  81.00 118.00 |
| 2  1 0  4 1  48 | 1.00 0.00  192.00 1.00 |
| 4  1 2 3 4  4 5 6 8  3 4 5 6  5 4 5 6  3 | 658.00 724.00 914.00 1144.00  1474.00 1617.00 2040.00 2552.00  1158.00 1270.00 1602.00 2004.00  1234.00 1350.00 1702.00 2128.00 |