**ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ**

**ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ**

***ЛУГАНСКОЙ НАРОДНОЙ РЕСПУБЛИКИ***

**«ЛУГАНСКИЙ НАЦИОНАЛЬНЫЙ УНИВЕРСИТЕТ ИМЕНИ ВЛАДИМИРА ДАЛЯ»**

Факультет компьютеризированных систем и информационных техно-логий

Кафедра информатики и программной инженерии

# ОТЧЁТ ПО ЛАБОРАТОРНЫМ РАБОТАМ

по дисциплине «**Введение в программную инженерию**»

Выполнил: студент 1 курса

группы ИT-691

Шило Даниил

Проверил(а): Чигрина А.Н.

Луганск, 2019 г.

Вариант 16

Задание 1:

1. #include <iostream>
2. #include <cmath>
3. #include <string>
4. **using** **namespace** std;
5. **int** main()
6. {
8. //Задание 1: В матрице X(5×5) определите произведение положительных элементов главной диагонали.
10. **double** matrice[5][5]; //Матрица
11. srand(time(NULL));    //Указывает время от которого создаются "случайные" числа
13. **for** (**int** i = 0; i < 5; i++) //Заполнение матрицы
14. {
15. **for** (**int** j = 0; j < 5; j++)
16. {
17. matrice[i][j] = rand() % 100;
18. }
19. }
20. //Главная диагональ - элементы с повторяющимся индексом (в квадратной матрице)
21. cout << "Вывод диагонали матрицы: \n";
22. **for** (**int** i = 0; i < 5; i++)
23. {
24. cout << string(i, '\t') << matrice[i][i] << endl; //string(int, char) - вставляет символы n-ное количество раз
25. }

Задание 2: Дан двумерный массив из пятнадцати строк и двух столбцов. Найти номера двух соседних строк, сумма элементов в которых минимальна.

1. //Задание 2: Дан двумерный массив из пятнадцати строк и двух столбцов. Найти номера двух соседних строк, сумма элементов в которых минимальна.
2. **int** matrice2[15][2];
3. **for** (**int** i = 0; i < 15; i++)
4. {
5. **for** (**int** k = 0; k < 2; k++)
6. {
7. matrice2[i][k] = rand() % 100;
8. }
9. }
11. **double** minimal\_sum = sqrt(pow(matrice2[0][0] + matrice2[0][1], 2) + pow(matrice2[1][0] + matrice2[1][1], 2)); //Сумма длин векторов первых двух строк
12. **int** numberofstring[2];
13. **for** (**int** i = 1; i < 15; i++)
14. {
15. **int** sum[2] = {matrice2[i][0] + matrice2[i][1], matrice2[i - 1][0] + matrice2[i - 1][1]}; //Сумма данной строки и предыдущей (начинаем со 2 строки)
16. **double** absofvec = sqrt(pow(sum[0], 2) + pow(sum[1], 2));
17. **if** (absofvec < minimal\_sum)
18. {
19. minimal\_sum = absofvec;
20. **int** numberofstring[2] = {i - 1, i};
21. }
22. }
23. cout << "Сумма длин векторов двух соседних строк: " << minimal\_sum << "\n";
24. cout << "Номера соседних строк с минимальным абсолютным значением суммы: " << numberofstring[0] << "," << numberofstring[1] << endl;
25. }

Задание 3:

1. #include <iostream>
3. **using** **namespace** std;
5. **int** main()
6. {
7. srand(time(NULL));
8. //Задание 3: Дан двумерный массив размером 7х7, заполняется случайными элементами. Найти наименьшее из значений элементов, расположенных в местах, где находятся нули так, как показано на рисунке
9. **int** matrice3[7][7];
10. //Заполнение случайными числами
11. **for** (**int** i = 0; i < 7; i++)
12. {
13. **for** (**int** j = 0; j < 7; j++)
14. {
15. matrice3[i][j] = rand() % 100;
16. }
17. }
18. //Цикл который будет искать числа вне равнобедренного треугольника, который построен внизу матрицы
19. **int** minimal\_num = matrice3[5][0];
20. **for** (**int** i = 6; i >= 0; i--)
21. {
22. **switch** (i)
23. {
24. **case** 6:
25. **if** (matrice3[i][0] < minimal\_num)
26. minimal\_num = matrice3[i][0];
27. **else** **if** (matrice3[i][-1] < minimal\_num)
28. minimal\_num = matrice3[i][-1];
29. **break**;
30. **case** 5:
31. **if** (matrice3[i][0] < minimal\_num)
32. minimal\_num = matrice3[i][0];
33. **else** **if** (matrice3[i][-1] < minimal\_num)
34. minimal\_num = matrice3[i][-1];
35. **else** **if** (matrice3[i][1] < minimal\_num)
36. minimal\_num = matrice3[i][1];
37. **else** **if** (matrice3[i][-1] < minimal\_num)
38. minimal\_num = matrice3[i][-2];
39. **break**;
40. **case** 4:
41. **for** (**int** k = 0; k < 7; k++)
42. {
43. **if** (k == 4)
44. **continue**;
45. **if** (matrice3[i][k] < minimal\_num)
46. minimal\_num = matrice3[i][k];
47. }
49. **default**:
50. **for** (**int** k = 0; k < 7; k++)
51. {
52. **if** (matrice3[i][k] < minimal\_num)
53. {
54. minimal\_num = matrice3[i][k];
55. }
56. }
57. **break**;
58. }
59. }
60. cout << "Наименьшее число в этом массиве: " << minimal\_num << "\n";
61. }