*Федеральное государственное бюджетное образовательное учреждение* *высшего образования*

|  |  |
| --- | --- |
|  | ***«Московский государственный технический университет  имени Н.Э. Баумана»***  ***(национальный исследовательский университет)***  ***(МГТУ им. Н.Э. Баумана)*** |

ФАКУЛЬТЕТ \_\_\_\_ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ\_\_\_\_\_\_\_\_

КАФЕДРА \_\_\_\_\_\_Компьютерные Системы и сети (ИУ6)\_\_\_\_\_\_\_\_\_\_\_

**Отчет**

**по лабораторной работе № 2**

**Название лабораторной работы: Работа с массивами С++**

**Дисциплина: Объектно-ориентированное программирование**

Студент гр. ИУ6-22Б  **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А. П. Плютто**



(Подпись, дата) (И.О. Фамилия)



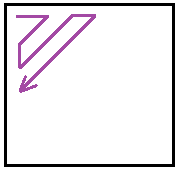
Преподаватель  **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ О. А. Веселовская**

(Подпись, дата) (И.О. Фамилия)

Москва, 2023

Задание

На главной диагонали матрицы D(*N,N*) найти значение и номер элемента, произведение которого с предыдущим максимально. Распечатать:

а) исходную матрицу;

б) значение найденного элемента;

в) столбец и строку, на пересечении которых расположен найденный элемент;

г) зигзагом вдоль главной диагонали все элемента, начиная с первого.

Решение

1. Создадим, заполним и выведем матрицу, найдем и выведем искомый элемент, а также столбец и строку, на пересечении которых расположен найденный элемент

1. int mas[n][n];

2. for (int i = 0; i < n; i++)

3. {

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

5. {

6. mas[i][j] = rand() % 1000;

7. cout << mas[i][j] << " ";

8. if (i != 0 and i == j and max < mas[i][j] \* mas[i - 1][j - 1])

9. {

10. max = mas[i][j] \* mas[i - 1][j - 1];

11. imax = i;

12. jmax = j;

13. }

14. }

15. cout << "\n";

16. }

17. cout << "max = " << max << " i = " << imax << " j = " << jmax;

18.

2. Зигзагом выведем все элементы

Отметим, что, чтобы вывести зигзагом матрицу, необходимо рассмотреть 4 случая:

* Вывод еще не дошел до главной диагонали
  + Вывод идет вниз
  + Вывод идет вверх
* Вывод дошел до главной диагонали
  + Вывод идет вниз
  + Вывод идет вверх

Все эти случаи были рассмотрены отдельно и реализованы с помощью системы флагов

1. int i = 0;

2. int j = 0;

3. bool vect = 1;

4. bool right = 0;

5. cout << "\n"

6. << mas[i][j] << " ";

7. while (i < n - 1 or j < n - 1)

8. {

9. if (((i == 0) and (j == (n - 1)) and n % 2 == 1) or

10. ((j == 0) and (i == (n - 1)) and n % 2 == 0))

11. {

12. right = 1;

13. vect = !vect;

14. }

15. if (!right)

16. {

17. if (vect)

18. {

19. j++;

20. cout << mas[i][j] << " ";

21. while (j != 0)

22. {

23. j--;

24. i++;

25. cout << mas[i][j] << " ";

26. };

27. }

28. else

29. {

30. i++;

31. cout << mas[i][j] << " ";

32. while (i != 0)

33. {

34. j++;

35. i--;

36. cout << mas[i][j] << " ";

37. };

38. };

39. }

40. else

41. {

42. if (vect)

43. {

44. j++;

45. cout << mas[i][j] << " ";

46. while (j != n - 1)

47. {

48. j++;

49. i--;

50. cout << mas[i][j] << " ";

51. };

52. }

53. else

54. {

55. i++;

56. cout << mas[i][j] << " ";

57. while (i != n - 1)

58. {

59. j--;

60. i++;

61. cout << mas[i][j] << " ";

62. };

63. };

64. };

65. vect = !vect;

66. };

67.

Весь получившийся код с отладочной программой и схема её алгоритма:

1. #include "iostream"

2. using std::cin, std::cout;

3. #define n 10

4.

5. int main()

6. {

7. int max = 0;

8. int jmax{};

9. int imax{};

10. int mas[n][n];

11. for (int i = 0; i < n; i++)

12. {

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

14. {

15. mas[i][j] = rand() % 1000;

16. cout << mas[i][j] << " ";

17. if (i != 0 and i == j and max < mas[i][j] \* mas[i - 1][j - 1])

18. {

19. max = mas[i][j] \* mas[i - 1][j - 1];

20. imax = i;

21. jmax = j;

22. }

23. }

24. cout << "\n";

25. }

26. cout << "max = " << max << " i = " << imax << " j = " << jmax;

27. int i = 0;

28. int j = 0;

29. bool vect = 1;

30. bool right = 0;

31. cout << "\n"

32. << mas[i][j] << " ";

33. while (i < n - 1 or j < n - 1)

34. {

35. if (((i == 0) and (j == (n - 1)) and n % 2 == 1) or

36. ((j == 0) and (i == (n - 1)) and n % 2 == 0))

37. {

38. right = 1;

39. vect = !vect;

40. }

41. if (!right)

42. {

43. if (vect)

44. {

45. j++;

46. cout << mas[i][j] << " ";

47. while (j != 0)

48. {

49. j--;

50. i++;

51. cout << mas[i][j] << " ";

52. };

53. }

54. else

55. {

56. i++;

57. cout << mas[i][j] << " ";

58. while (i != 0)

59. {

60. j++;

61. i--;

62. cout << mas[i][j] << " ";

63. };

64. };

65. }

66. else

67. {

68. if (vect)

69. {

70. j++;

71. cout << mas[i][j] << " ";

72. while (j != n - 1)

73. {

74. j++;

75. i--;

76. cout << mas[i][j] << " ";

77. };

78. }

79. else

80. {

81. i++;

82. cout << mas[i][j] << " ";

83. while (i != n - 1)

84. {

85. j--;

86. i++;

87. cout << mas[i][j] << " ";

88. };

89. };

90. };

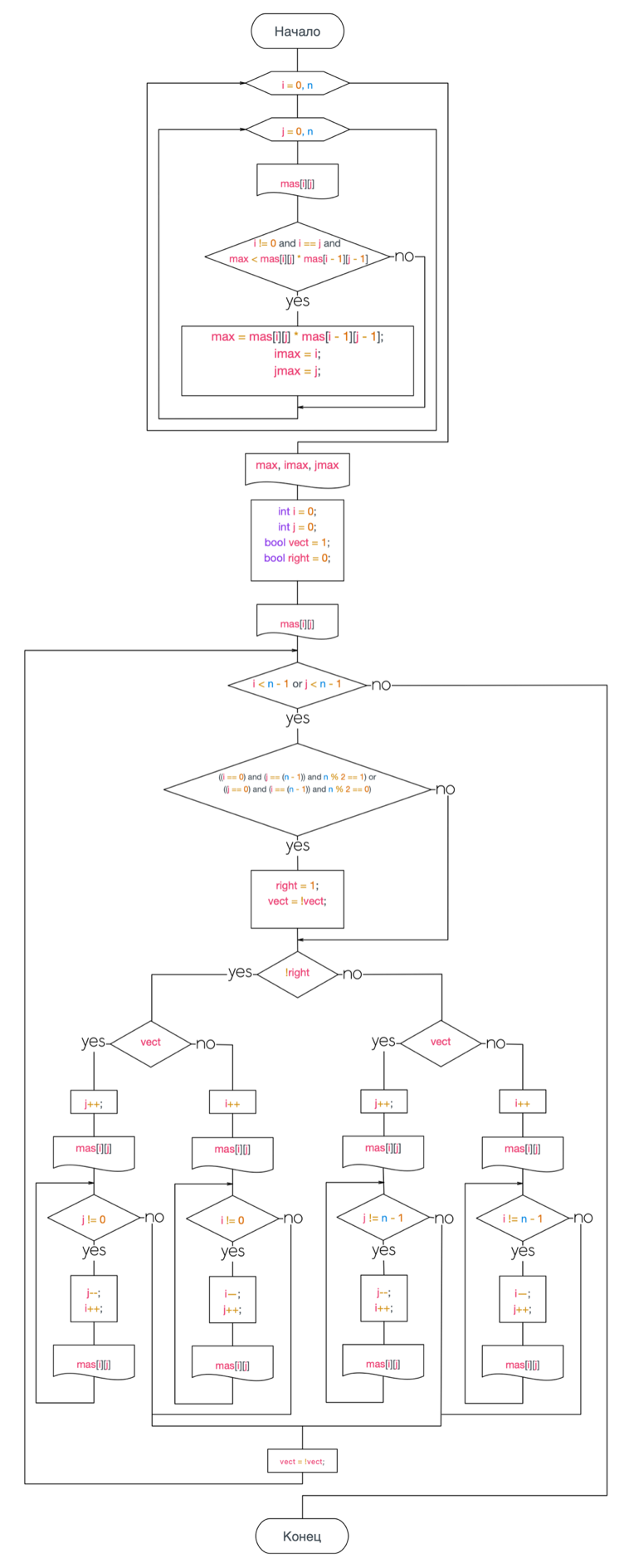
91. vect = !vect;

92. };

93. return 0;

94. }

95.



Вывод: В ходе выполнения работы были изучены и применены на практике массивы и их особенности в языке C++.

Были разработаны процедуры для инициализации матрицы случайными числами и вывода матрицы на экран, найдено значение и номер элемента, произведение которого с предыдущим максимально и выведена матрица зигзагом.