

Міністерство освіти і науки України
Національний технічний університет України «Київський політехнічний
інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки

Кафедра інформатики та програмної інженерії

Звіт

з лабораторної роботи № 8 з дисципліни
«Основи програмування 1. Базові конструкції»

«Багатовимірні масиви»

Варіант №32

Виконав студент ІП-14 Шляхтун Денис Михайлович
(шифр, прізвище, ім'я, по батькові)

Перевірив Камінська Поліна Анатоліївна
(прізвище, ім'я, по батькові)

Київ 2021

Лабораторна робота №8

Тема: багатовимірні масиви.

Мета: опанувати технологію використання двовимірних масивів даних (матриць), навчитися розробляти алгоритми та програми із застосуванням матриць.

Хід роботи

Задача. Задана дійсна квадратна матриця розмірності $n \times n$. Замінити стовпці із нульовими значеннями елементами головної діагоналі.

Постановка задачі. Результатом розв'язку є масив, стовпці з нульовими значеннями якого будуть замінені елементами головної діагоналі.

Побудова математичної моделі. Складемо таблицю імен змінних

<i>Змінна</i>	<i>Тип</i>	<i>Ім'я</i>	<i>Призначення</i>
<i>Основні змінні</i>			
Розмірність масиву	Цілий	$n, j = n$	Початкове дане
Двовимірний масив	Дійсний	$arr[][]$	Початкове дане, результат
Масив діагональних елементів	Дійсний	$d[]$	Проміжне значення
Кількість цифр чисел у масиві	Цілий	an	Проміжне значення
Кількість цифр після коми	Цілий	pn	Проміжне значення
<i>Змінні підпрограми</i>			
Розмір масиву	Цілий	s	Проміжне значення
Лічильник	Цілий	i, k	Проміжне значення
Перевірка умови	Булевий	$check$	Проміжне значення

Випробування коду на C++.

Код:

```
1  #include <iostream>
2  #include <iomanip> //для форматованого виведення
3  using namespace std;
4
5  int inputSize(); //введення розміру масиву
6  float** arrayDecl(int, int); //оголошення двовимірного масиву
7  void arrayRand(float**, int, int, int, int); //генерація двовимірного масиву з випадкових значень
8  void arrayOutput(float*, int, int, int); //вивід одновимірного масиву
9  void arrayOutput(float**, int, int, int, int); //вивід двовимірного масиву
10 void arrayDiag(float**, float*, int); //наповнення одновимірного масиву діагональними елементами двовимірного
11 void arrayChange(float**, float*, int); //заміна стовбців з нульовими значеннями матриці елементами одновимірного масиву
12 void arrayDelete(float**, int, int); //видалення двовимірного масиву

```

```
14 int main()
15 {
16     srand(time(NULL));
17     int an = 1, pn = 1; //кількість всіх цифр і цифр після коми
18     int n = inputSize(); //введення розміру масиву
19
20     //оголошення та ініціалізація двовимірного масиву
21     float **arr = arrayDecl(n, n);
22     arrayRand(arr, n, n, an, pn);
23     cout << "Generated array:" << endl;
24     arrayOutput(arr, n, n, an, pn);
25
26     //оголошення та ініціалізація масиву для діагональних елементів
27     float *d = new float[n];
28     arrayDiag(arr, d, n);
29     cout << "Diagonal elements:" << endl;
30     arrayOutput(d, n, an, pn);
31
32     //заміна стовбців з нульовими значеннями елементами діагоналі
33     arrayChange(arr, d, n);
34     cout << endl << "Final result:" << endl;
35     arrayOutput(arr, n, n, an, pn);
36
37     //видалення динамічних масивів
38     arrayDelete(arr, n, n);
39     delete[]d;
40
41     cout << endl;
42     system("pause");
43 }
```

```

45     //введення розміру масиву
46     int inputSize()
47     {
48         int s;
49         cout << "Enter the size of array: ";
50         cin >> s;
51         return s;
52     }
53
54     //оголошення двовимірного масиву
55     float** arrayDecl(int n, int j)
56     {
57         float **arr = new float* [n];
58         for (int i = 0; i < n; i++)
59             arr[i] = new float[j];
60         return arr;
61     }
62
63     //генерація двовимірного масиву з випадкових значень
64     void arrayRand(float **arr, int n, int j, int an, int pn)
65     {
66         for (int i = 0; i < n; i++)
67         {
68             for (int k = 0; k < j; k++)
69             {
70                 arr[i][k] = rand() % int(2 * pow(10, an) - 1)
71                     - ( pow(10, an) - 1);
72                 arr[i][k] /= pow(10, pn);
73             }
74         }
75     }

```

```

77     //вивід одновимірного масиву
78     void arrayOutput(float* arr, int n, int an, int pn)
79     {
80         int w = an + pn + 3;
81         for (int k = 0; k < n; k++)
82             cout << setbase(pn) << setw(w) << arr[k];
83     }
84
85     //вивід двовимірного масиву
86     void arrayOutput(float **arr, int n, int j, int an, int pn)
87     {
88         for (int k = 0; k < n; k++)
89         {
90             arrayOutput(&arr[k][0], j, an, pn);
91             cout << endl;
92         }
93     }
94
95     //наповнення одновимірного масиву діагональними елементами двовимірного
96     void arrayDiag(float **arr, float *d, int n)
97     {
98         for (int i = 0; i < n; i++)
99             d[i] = arr[i][i];
100    }

```

```

102  /*заміна стовців з нульовими значеннями матриці
103  | елементів одновимірного масиву*/
104  void arrayChange(float **arr, float *d, int n)
105  {
106      bool check;
107      for (int i = 0; i < n; i++)
108      {
109          check = true;
110          for (int k = 0; k < n && check; k++)
111          {
112              if (arr[k][i] == 0)
113              {
114                  for (int l = 0; l < n; l++)
115                      arr[l][i] = d[l];
116                  check = false;
117              }
118          }
119      }
120  }
121
122  //видалення двовимірного масиву
123  void arrayDelete(float** arr, int n, int m)
124  {
125      for (int i = 0; i < m; i++)
126          delete[] arr[i];
127      delete[] arr;
128  }

```

Результат:

```
E:\university\I s...  —  □  ×
Enter the size of array: 5
Generated array:
-1.5 -9.5 -2.5 -4.4 8.2
 1.6  0.2 -2.3 -0.9 -0.6
-8.7  5.5  9.9 -3.7 -9.7
 8.5  1.8 -2.5  4.2 -8.3
 7.7  8.7   8 -0.6 -5.3
Diagonal elements:
-1.5  0.2  9.9  4.2 -5.3
Final result:
-1.5 -9.5 -2.5 -4.4 8.2
 1.6  0.2 -2.3 -0.9 -0.6
-8.7  5.5  9.9 -3.7 -9.7
 8.5  1.8 -2.5  4.2 -8.3
 7.7  8.7   8 -0.6 -5.3
Press any key to continue . . .

E:\university\I semester\Ba...  —  □  ×
Enter the size of array: 8
Generated array:
 0.6  0.7  0.4 -0.9  0.3  0.4 -0.4 -0.9
-0.9 -0.5 -0.2 -0.6  0.8 -0.5 -0.6 -0.6
 0.3 -0.2  0.9 -0.8  0.2  0.8 -0.9 -0.3
 0.4 -0.8  0.2  0.6   0  0.8  0.6  0.6
 0.3 -0.5 -0.3  0.7  0.4  0.3 -0.5 -0.9
-0.5  0.9  0.9 -0.6  0.7  0.3 -0.3 -0.4
-0.3  0.4 -0.2   0 -0.2 -0.5 -0.4 -0.3
 0.9 -0.1   0 -0.7 -0.9  0.6  0.2  0.5
Diagonal elements:
 0.6 -0.5  0.9  0.6  0.4  0.3 -0.4  0.5
Final result:
 0.6  0.7  0.6  0.6  0.6  0.4 -0.4 -0.9
-0.9 -0.5 -0.5 -0.5 -0.5 -0.5 -0.6 -0.6
 0.3 -0.2  0.9  0.9  0.9  0.8 -0.9 -0.3
 0.4 -0.8  0.6  0.6  0.6  0.8  0.6  0.6
 0.3 -0.5  0.4  0.4  0.4  0.3 -0.5 -0.9
-0.5  0.9  0.3  0.3  0.3  0.3 -0.3 -0.4
-0.3  0.4 -0.4 -0.4 -0.4 -0.5 -0.4 -0.3
 0.9 -0.1  0.5  0.5  0.5  0.6  0.2  0.5
Press any key to continue . . .
```

Перевірка правильності виконання коду

1. Перевіримо перший запуск

- a. Головна діагональ визначена правильно
- b. Стовпці з нульовими значеннями відсутні
- c. Матриця залишається без змін

2. Перевіримо другий запуск

- a. Головна діагональ визначена правильно
- b. Стовпці з нульовими значеннями: 3, 4, 5
- c. 3, 4 та 5 стовбці успішно замінені діагональними елементами

Враховуючи те, що результати програми співпали з теоретично визначеними результатами, можна зробити висновок, що програма працює правильно.

Висновок: При виконанні лабораторної роботи було набуто навички використання та обробки динамічних двовимірних масивів. Була побудована

математична модель. Для виконання задачі була написана програма на мові C++. За допомогою перевірки було доведено правильність результатів програми.