



# Отчет по лабораторной работе №15 по курсу Фундаментальная информатика

Студент группы М80-101Б-21 Постнов Александр Вячеславович, № по списку 17

Контакты www, e-mail:

Работа выполнена: «» 2021 г.

Преподаватель: каф. 806 Титов В.К.

Входной контроль знаний с оценкой

Отчет сдан « » 2021 г., итоговая оценка

Подпись преподавателя

1. **Тема:** Динамические массивы, обработка матриц

2. **Цель работы:** составить и отладить программу на СИ на заданную тему.

3. **Задание (вариант: 17):** умножение строки с минимальным элементом матрицы на столбец с максимальным элементом.

4. **Оборудование(лабораторное):**

ЭВМ, процессор, имя узла сети, с ОП, ГБ,

НМД, ГБ, терминал-адрес, принтер,

Другие устройства

*Оборудование ПЭВМ студента, если использовалось:*

Процессор AMD Ryzen 5 4500U, с ОП 8 ГБ

Другие устройства

5. **Программное обеспечение:**

Операционная система семейства, наименование, версия,

интерпретатор команд, версия

Система программирования, версия

Редактор текстов, версия

Утилиты операционной системы

Прикладные системы и программы

Местонахождение и имена файлов программ и данных

*Программное обеспечение ЭВМ студента, если использовалось:*

Операционная система семейства GNU/Linux, наименование Pop!\_OS версия 21.04

интерпретатор команд GNOME Terminal версия 3.38.2

Система программирования

версия Редактор текстов

emacs версия 3.27.20

Утилиты операционной системы cat

Прикладные системы и программы

Местонахождение и имена файлов программ и данных

6. **Идея, метод, алгоритм** решения задачи (в формах: словесной, псевдокода, графической [блок-схема,

диаграмма, рисунок, таблица] или формальные спецификации с пред- и постусловиями)

- 1) Динамический массив будет создан с помощью указателей.
- 2) Матрица будет заполняться случайными элементами с помощью рандома
- 3) Буду использовать 4 переменные:  $minj$  – минимальный элемент строки,  $imin$  – номер строки с этим элементом,  $maxi$  – максимальный элемент столбца,  $jmax$  – номер столбца с этим элементом
- 4) Чтобы заполнить переменную  $imin$ , пройду по матрице по строкам, и буду проверять, текущий элемент, если он меньше  $minj \rightarrow$  обновляю  $minj$  и  $imin$ . Аналогично для  $maxi$  и  $jmax$  (проход по матрице будет по столбцам)
- 5) Так как в матрице может быть несколько минимумов и максимумов, я буду использовать первые строку и столбец с минимумом и максимумом
- 6) Так как умножение строки  $1 \times N$  на столбец  $N \times 1$  это матрица  $1 \times 1$ , поэтому заведу только переменную  $ans$ , в которую я запишу ответ
- 7) В данном случае умножение строки на столбец – это сумма попарных произведений соответствующих элементов.

**7. Сценарий выполнения работы** [план работы, первоначальный текст программы в черновике (можно на отдельном листе) и тесты либо соображения по тестированию].

```
#include <time.h>
#include <stdio.h>
#include <stdlib.h>
```

```
int main(void){
    int **a, n, answer = 0;
    scanf("%d", &n);
    a = new int*[n];
    srand(time(0));
    for (int i = 0; i < n; i++){//создание матрицы
        a[i] = new int[n];
    }
    for (int i = 0; i < n; i++){ //ввод матрицы и вывод матрицы
        for (int j = 0; j < n; j++){
            a[i][j] = rand() % 15;
            printf(" %3d", a[i][j]);
        }
        printf("\n");
    }
    int minj = a[0][0]; //минимальный элемент строки
    int imin = 0; //номер строки
    int maxi = a[0][0]; //максимальный элемент столбца
    int jmax = 0; //номер столбца
    for (int i = 0; i < n; i++){ //поиск номера строки с минимальным элементом(самая первая строка с
        этим элементом)
        for (int j = 0; j < n; j++){
            if (a[i][j] < minj){
                minj = a[i][j];
                imin = i;
            }
        }
    }
}
```

```

    }
    for (int j = 0; j < n; j++){ //поиск номера столбца с максимальным элементом(самый первый столбец
с этим элементом)
        for (int i = 0; i < n; i++){
            if (a[i][j] > maxi){
                jmax = j;
                maxi = a[i][j];
            }
        }
    }
    for (int v = 0; v < n; v++){ //подсчет ответа
        answer += (a[v][jmax] * a[imin][v]);
    }
    printf("%d \n", answer);
    return 0;
}

```

Тесты:

1) Рандом сгенерировал матрицу:

4

```

13  3  0  1
 4  7 12  6
12  7  7  1
 0  8  9  6

```

Ручной просчёт:

imin = 0 (мин элемент - 0)

jmax = 0 (макс элемент - 13)

Ответ:  $13 * 13 + 3 * 4 + 0 * 12 + 1 * 0 = 169 + 12 = 181$

2) Рандом сгенерировал матрицу:

5

```

 7  3  4  7  3
 8 10 11  2  9
 9  1 11  0 11
11 13  2  6 13
 0  4  4  1  9

```

Ручной просчёт:

imin = 2 (мин элемент - 0)

jmax = 1 (макс элемент - 13)

Ответ:  $9 * 3 + 10 * 1 + 1 * 11 + 13 * 0 + 4 * 11 = 27 + 10 + 11 + 44 = 92$

3) Рандом сгенерировал матрицу:

1

7

Ручной просчёт:

imin = 0 (мин элемент 7)

jmax = 0 (макс элемент 7)

Ответ:  $7 * 7 = 49$

Допущен к выполнению работы. Подпись преподавателя \_\_\_\_\_

```
[alex@alex 15c(?)]$ cat head.txt
```

```
[alex@alex 15c(?)]$ cat main.cpp
```

```
#include <stdlib.h>
```

```
int main(void){
    int **a, n, answer = 0;
    scanf("%d", &n);
    a = new int*[n];
    srand(time(0));
    for (int i = 0; i < n; i++){//создание матрицы
        a[i] = new int[n];
    }
    for (int i = 0; i < n; i++){ //ввод матрицы и вывод матрицы
        for (int j = 0; j < n; j++){
            a[i][j] = rand() % 15;
            printf(" %3d", a[i][j]);
        }
        printf("\n");
    }
    int minj = a[0][0]; //минимальный элемент строки
    int imin = 0; //номер строки
    int maxi = a[0][0]; //максимальный элемент столбцы
    int jmax = 0; //номер столбца
    for (int i = 0; i < n; i++){ //поиск номера строки с минимальным элементом(самая первая строка с
    ЭТИМ ЭЛЕМЕНТОМ)
        for (int j = 0; j < n; j++){
            if (a[i][j] < minj){
                minj = a[i][j];
                imin = i;
            }
        }
    }
    for (int j = 0; j < n; j++){ //поиск номера столбца с максимальным элементом(самый первый столбец
    с ЭТИМ ЭЛЕМЕНТОМ)
        for (int i = 0; i < n; i++){
```

```

        if (a[i][j] > maxi){
            jmax = j;
            maxi = a[i][j];
        }
    }
}
for (int v = 0; v < n; v++){ //подсчет ответа
    answer += (a[v][jmax] * a[imin][v]);
}
printf("%d \n", answer);
return 0;
}[alex@alex 15c(?)$ g++ main.cpp
[alex@alex 15c(?)$ ./a.out
4
13 3 0 1
4 7 12 6
12 7 7 1
0 8 9 6
181
[alex@alex 15c(?)$ ./a.out
5
7 3 4 7 3
8 10 11 2 9
9 1 11 0 11
11 13 2 6 13
0 4 4 1 9
92
[alex@alex 15c(?)$ ./a.out
1
7
49

```

**9. Дневник отладки** должен содержать дату и время сеансов отладки и основные события (ошибки в сценарии и программе, нестандартные ситуации) и краткие комментарии к ним. В дневнике отладки приводятся сведения об использовании других ЭВМ, существенном участии преподавателя и других лиц в написании и отладке программы.

№	Лаб. или дом.	Дата	Время	Событие	Действие по исправлению	Примечание
1	дом	27.11	11:00	Программа не скомпилировалась из-за синтаксической ошибки	Добавил ; в конец 15 строки	

**10. Замечания автора :**

## 11. Выводы

\_\_ В ходе лабораторной работы я составил и отладил программу на СИ

---

---

---

---

---

---

---

---

Недочёты при выполнении задания могут быть устранены следующим образом:

---

---

---

---

---

---

---

---

Подпись студента \_Постнов\_\_\_\_\_