Учреждение образования

Белорусский государственный университет информатики и радиоэлектроники

Индивидуальное практическое задание № 2

по дисциплине «Метрология, стандартизация и сертификация  
в информационных технологиях»

Вариант № 18

Выполнил студент: Сотников А.А.

группа 321702

Зачетная книжка № 32170077

Минск 2024

**Цель:** написать текст программы на языке программирования, заданном преподавателем. По тексту программы рассчитать метрики Холстеда и метрики сложности потока данных (спен и метрику Чепина).

**Условие:** Ввести массив *А(10, 10).* Заменить элементы массива, находящиеся на пересечении строк и столбцов с четными номерами, максимальным элементом данного массива. Вывести максимальный элемент, номера строки и столбца, в которых он находится в исходном массиве, исходный и результирующий массивы.

**Исходный код программы:**

#include <iostream>

using namespace std;

int main() {

int n = 10; int A[n][n]; int maxElement = 0;

// Ввод массива

cout << "Введите элементы массива 10x10:" << endl;

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

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

cin >> A[i][j];

}

}

// Поиск максимального элемента

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

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

if (A[i][j] > maxElement) {

maxElement = A[i][j];

}

}

}

// Замена элементов на пересечении чётных строк и столбцов

for (int i = 1; i < n; i += 2) {

for (int j = 1; j < n; j += 2) {

A[i][j] = maxElement;

}

}

// Вывод результатов

cout << "Максимальный элемент: " << maxElement << endl;

cout << "Результирующий массив:" << endl;

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

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

cout << A[i][j] << " ";

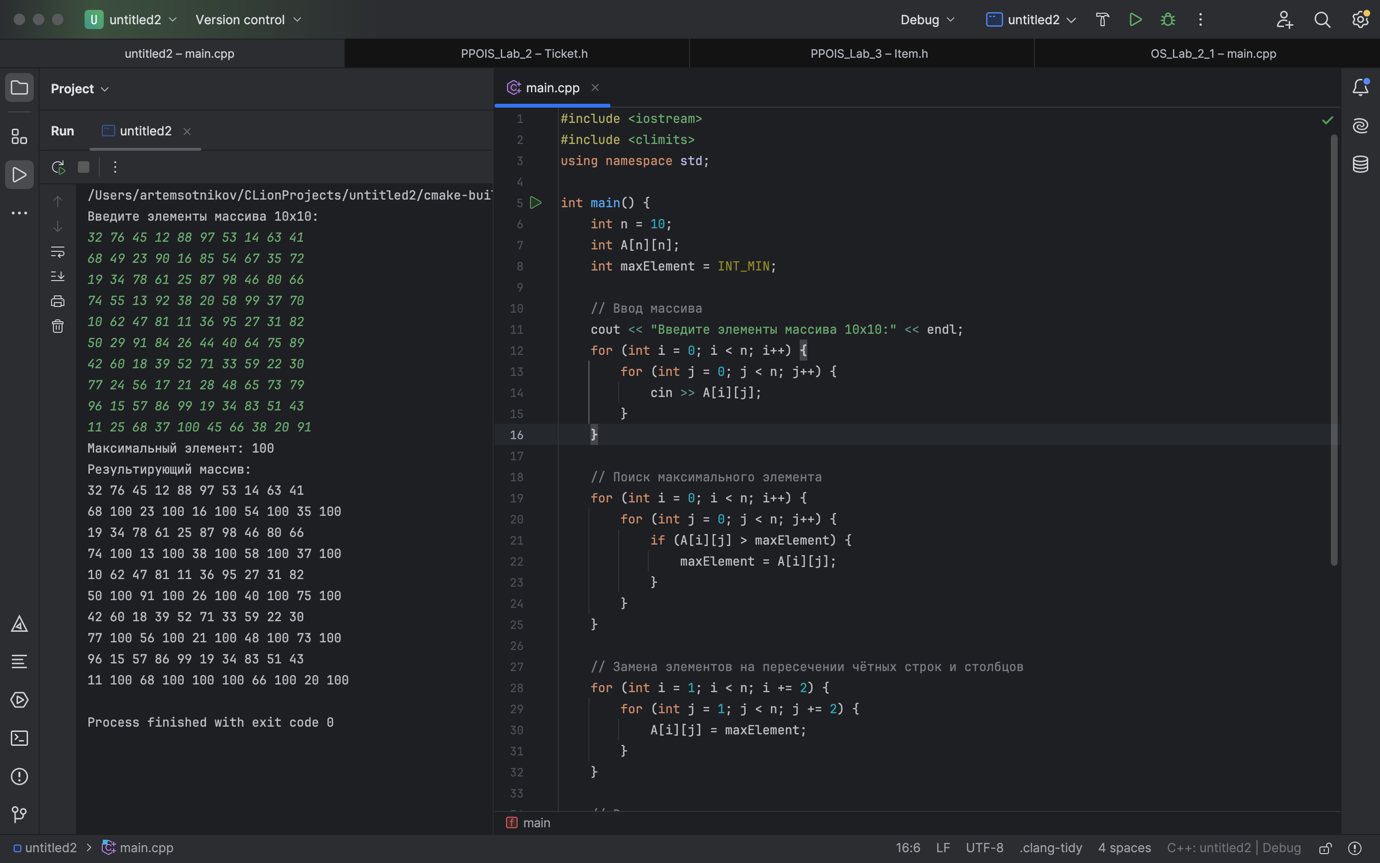
}

cout << endl;

}

return 0;

}



**Описание программы:**

**Назначение программы**

Программа обрабатывает двумерный массив размером 10×10. Она выполняет следующие операции:

1. Ввод массива с клавиатуры.
2. Поиск максимального элемента в массиве.
3. Замена элементов на пересечении **чётных строк и столбцов** (индексация с 0) этим максимальным элементом.
4. Вывод результата на экран.

**Описание переменных**

* int n: Размер массива (10). Используется для удобного изменения размера массива.
* int A[n][n]: Исходный массив размером 10×10.
* int maxElement: Переменная для хранения максимального элемента массива, инициализирована значением 0.

**Этапы работы программы**

1. **Инициализация переменных**  
   Переменная maxElement устанавливается в минимальное возможное значение (0), чтобы гарантировать корректный поиск максимального элемента.
2. **Ввод массива**  
   Пользователь вводит элементы массива построчно. Для ввода используются вложенные циклы:
   * Внешний цикл (for (int i = 0; i < n; i++)) отвечает за строки.
   * Внутренний цикл (for (int j = 0; j < n; j++)) отвечает за столбцы.
3. **Поиск максимального элемента**  
   Во вложенных циклах сравниваются все элементы массива с текущим значением maxElement. Если найден элемент больше текущего значения, то maxElement обновляется.
4. **Замена элементов на пересечении нечётных строк и столбцов**  
   Для всех индексов i и j, где i и j чётны (i=1,3,5,…, j=1,3,5,…), элемент массива A[i][j] заменяется значением maxElement.
5. **Вывод результатов**  
   Программа выводит:
   * Найденный максимальный элемент массива.
   * Модифицированный массив, где элементы на пересечении нечётных строк и столбцов заменены на maxElement.

**Расчёт метрики Холстеда:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| j | Оператор | F1j | i | Операнд | F2i |
| 1 | int | 2 | 1 | n | 6 |
| 2 | = | 4 | 2 | A | 6 |
| 3 | {} | 3 | 3 | maxElement | 4 |
| 4 | () | 7 | 4 | i | 5 |
| 5 | << | 3 | 5 | j | 5 |
| 6 | >> | 2 | 6 | 10 | 2 |
| 7 | for | 3 | 7 | 1 | 2 |
| 8 | if | 1 | 8 | 2 | 2 |
| 9 | > | 1 | 9 | 0 | 2 |
| 10 | += | 1 | 10 |  |  |
| 11 | cout | 2 | 11 |  |  |
| 12 | cin | 1 | 12 |  |  |
| 13 | endl | 3 | 13 |  |  |
| 14 | return | 1 | 14 |  |  |
| 15 | ; | 14 | 15 |  |  |
| η1​=15 |  | N1​=48 | η2​=9 |  | N2​=32 |

Словарь программы: η= **15 + 9 = 24**.

Длина программы: ***N*** **= 48 + 32 = 80.**

Объем программы: ***V*** **= 80 log2 24 ≈ 367**

**Расчёт спена программы:**

|  |  |
| --- | --- |
| Идентификатор | Спен |
| n | 1 |
| A | 1 |
| maxElement | 1 |
| i | 1 |
| j | 1 |
| 10 | 1 |
| 1 | 1 |
| 2 | 1 |
| 0 | 1 |
| Суммарный спен программы | **9** |

**Расчёт полной метрики Чепина:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Переменные | Полная метрика Чепина | | | | Метрика чепина ввода/вывода | | | |
| Группа переменных | ***P*** | ***M*** | ***C*** | ***T*** | ***P*** | ***M*** | ***C*** | ***T*** |
| Переменные относящиеся к группе | n | A, max  Element | i, j | -- | n | A, max  Element | i, j | -- |
| Количество переменных в группе | ***p = 1*** | ***m = 2*** | ***c = 2*** | ***t = 0*** | ***p = 1*** | ***m = 2*** | ***c = 2*** | ***t = 0*** |
| Метрика Чепинна | Q = P + 2⋅M + C + 0.5⋅T  = 1 + 4 + 2 + 0 = 7 | | | | Q = P + 2⋅M + C + 0.5⋅T  = 1 + 4 + 2 + 0 = 7 | | | |

**Вывод:** В ходе работы были рассчитаны метрики Холстеда и метрики сложности потока данных, что позволило оценить сложность программы и её структуры.