Отчёт по лабораторной работе №6

**Обработка двумерных статических массивов**

**15 вариант**

1.Матрица B[N1,N2], содержащая целые числа, вводится с клавиатуры. Найти максимальный и минимальный элементы матрицы и поменять их местами.

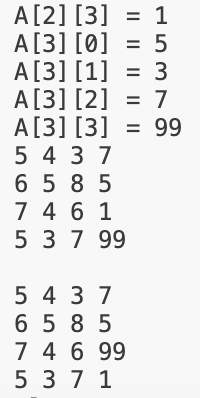
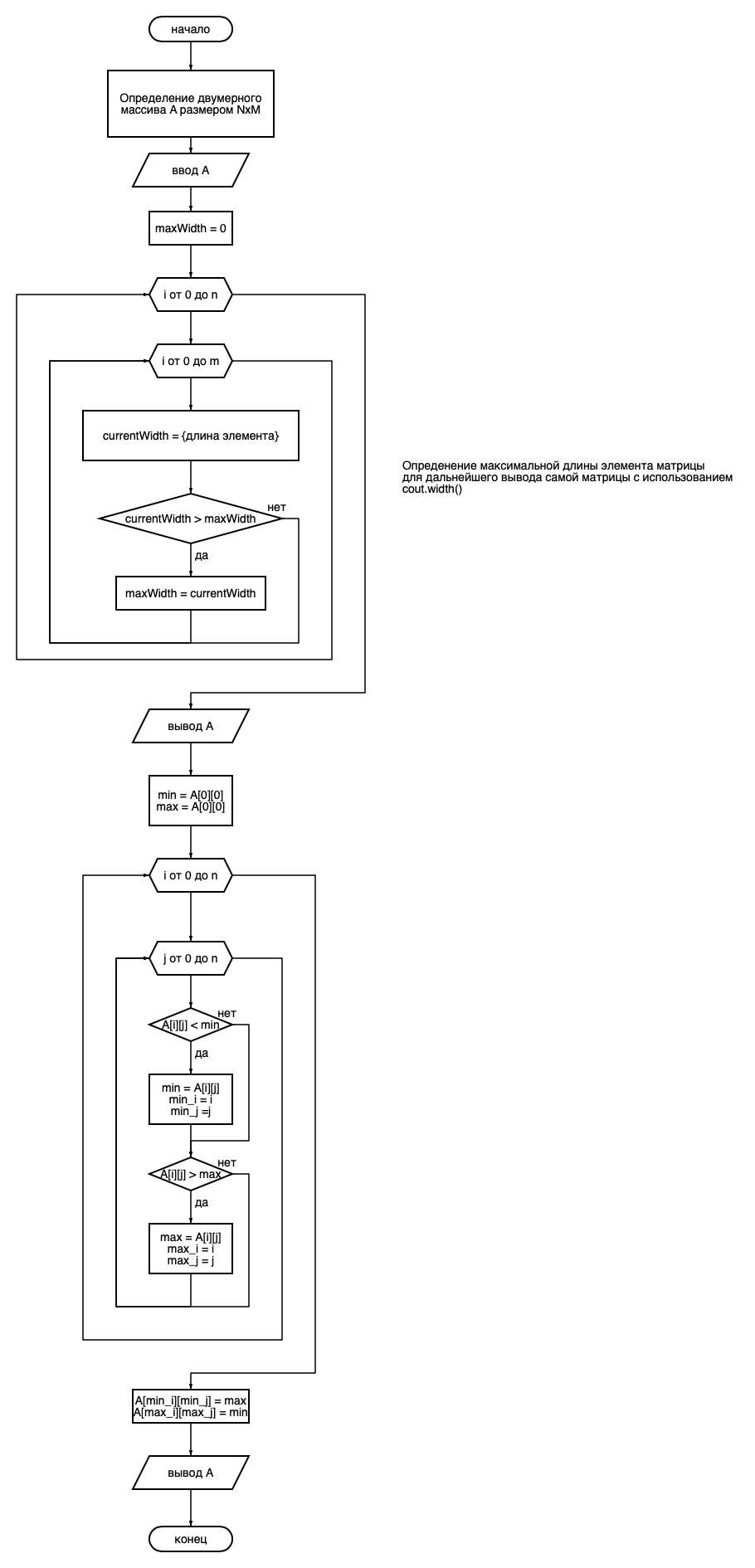
2. Дан двумерный статический массив действительных чисел размером N1xN2.

Поменять местами столбцы с номерами 0 и 1, 2 и 3, 4 и 5 и т.д.

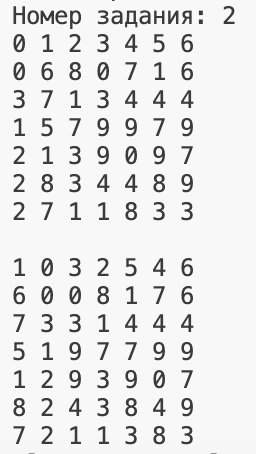
Код:

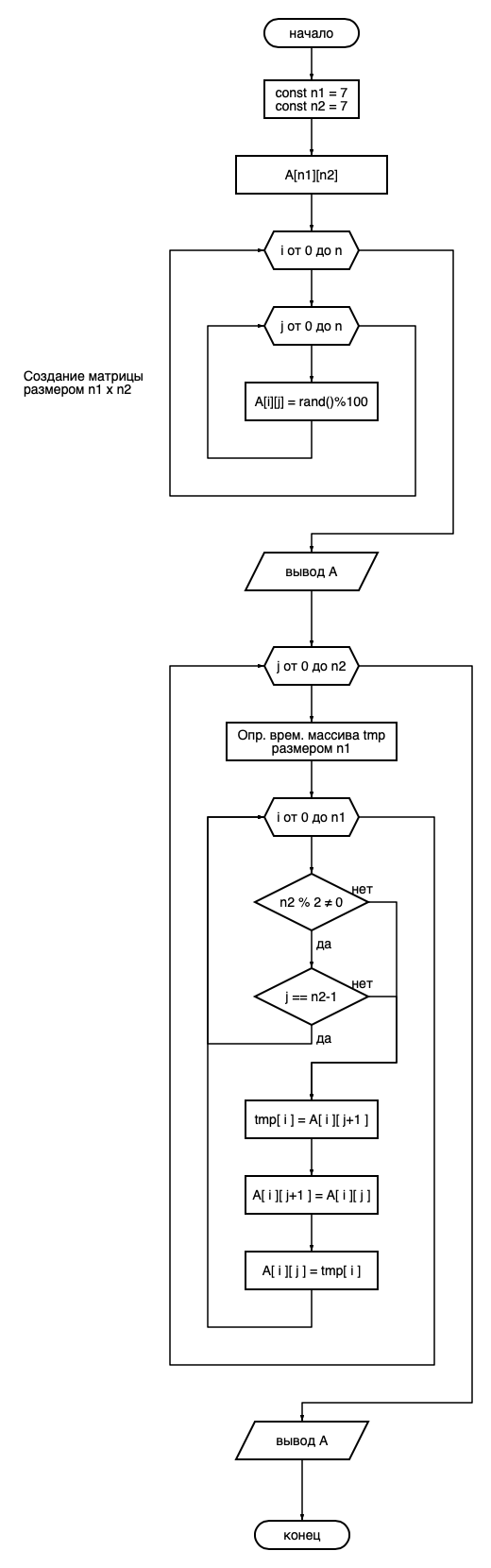
|  |
| --- |
| // Variant 15  #include <iostream>  int main() {  int num;  std::cout << "Номер задания: "; std::cin >> num;  switch (num) {  case 1: {  const int n = 4, m = 4;  // Создание двумерного массива  int a[n][m];  // Ввод матрицы  for (int i=0;i<n;i++) {  for (int j=0;j<m;j++) {  std::cout << "A[" << i << "][" << j << "] = "; std::cin >> a[i][j];  }  }  //Находим самый длинный элемент  int maxWidth = 0, currentWidth = 0;  for (int i=0;i<n;i++) {  for (int j=0;j<m;j++) {  currentWidth = std::to\_string(a[i][j]).length();  if (currentWidth > maxWidth) {  maxWidth = currentWidth;  }  }  }  // Вывод матрицы  for (int i=0;i<n;i++) {  for (int j=0;j<m;j++) {  std::cout.width(maxWidth + 1);  std::cout << a[i][j] << " ";  }  std::cout << std::endl;  }  // Поиск минимума и максимума  int min = a[0][0], max = a[0][0], min\_i, min\_j, max\_i, max\_j;  for (int i=0;i<n;i++) {  for (int j=0;j<m;j++) {  if (a[i][j] < min) {min = a[i][j]; min\_i = i; min\_j = j;};  if (a[i][j] > max) {max = a[i][j]; max\_i = i; max\_j = j;};  }  }  a[min\_i][min\_j] = max; a[max\_i][max\_j] = min;  // Вывод матрицы  std::cout << std::endl;  for (int i=0;i<n;i++) {  for (int j=0;j<m;j++) {  std::cout.width(maxWidth + 1);  std::cout << a[i][j];  }  std::cout << std::endl;  }  break;  }  case 2: {  const int n1 = 7, n2 = 7;  // Создание заданной матрицы  int a[n1][n2];  srand(time(0));  for (int i=0;i<n1;i++) {  for (int j=0;j<n2;j++)  a[i][j] = rand()%100;  }  // Вывод матрицы  for (int i=0;i<n1;i++) {  for (int j=0;j<n2;j++) {  std::cout << a[i][j] << " ";  }  std::cout.width(3);  std::cout << std::endl;  }    // Смена столбцов  for (int j=0;j<n2;j+=2) {  int tmp;  for (int i=0;i<n1;i++) {  if (n2 % 2 != 0) {  if (j==n2-1) break;  }  tmp = a[i][j+1];  a[i][j+1] = a[i][j];  a[i][j] = tmp;  }  }  // Вывод матрицы  std::cout << std::endl;  for (int i=0;i<n1;i++) {  for (int j=0;j<n2;j++) {  std::cout.width(3);  std::cout << a[i][j] << " ";  }  std::cout << std::endl;  }  break;  }  }  return 0;  } |

Задание 1:



Задание 2:





Вывод: В итоге данной экспериментальной процедуры, направленной на исследование функциональности и потенциала статических массивов в сфере программной инженерии на базе языков программирования С и С++, выявлены заметные корреляции между размером, структурой данных и эффективностью алгоритмов, реализованных с использованием указанных структур. Эта работа провела анализ, демонстрирующий эффекты различных размеров массивов на скорость выполнения операций, выявляя взаимосвязь между сложностью алгоритмов и пространственной сложностью данных. При исследовании предельных значений статических массивов обнаружены применимые границы, уточняющие ограничения операционной среды и общую производительность программных структур, отмечая, что увеличение размерности массивов приводит к асимптотическому росту времени выполнения с учетом ресурсов доступного процессорного времени и объема памяти. Более того, анализ результатов утверждает, что эффективность доступа к элементам статических массивов связана с использованием кэш-памяти, что подчеркивает важность понимания аппаратной архитектуры для оптимизации производительности при работе с данными определенного типа. Таким образом, данный эксперимент позволил получить углубленное представление о влиянии размеров и структуры статических массивов на работу алгоритмов, что открывает перспективы для более эффективного использования данных структур в контексте программирования на языках С и С++.