

Отчет по лабораторной 15

Задания

1	<p>1. Изучить формирование динамического массива с использованием стандартных функций, выполнив программу на языке C, записанную справа.</p> <p>Написать пояснения к программе.</p> <p>Добавить операторы вычисления максимального по модулю элемента массива.</p>
2	<div><div data-bbox="284 1151 647 1993"><p>2. Изучить способы выделения динамической памяти для одномерного массива, выполнив программу на языке C++, записанную в правой части.</p><p>Опробовать работу программы с разными значениями вводимых символов.</p></div><div data-bbox="647 1151 1437 1993"><p>Операция <code>new</code> позволяет выделить и сделать доступным блок памяти, размер которого соответствует типу данных. Оператор возвращает указатель на блок памяти. Операция <code>delete</code> освобождает участок памяти, выделенный оператором <code>new</code>. Аргументом оператора <code>delete</code> выступает адрес первой ячейки массива.</p><pre>#include <iostream> using namespace std; void main() { setlocale(LC_CTYPE, "Russian"); char *pStr; // указатель для адреса массива char c, *pc; int j, length, count = 0; cout << "Введите размер строки "; cin >> length; pStr = new char[length + 1]; //выделение динам.памяти cout << "Введите строку "; cin >> pStr; cout << "Введите символ "; cin >> c; pc = &c; for (j = 0; j < length; j++) if (pStr[j] == *pc) count++; cout << count << " повторений символа " << c << endl; delete[] pStr; // освобождение динам. памяти }</pre></div></div>

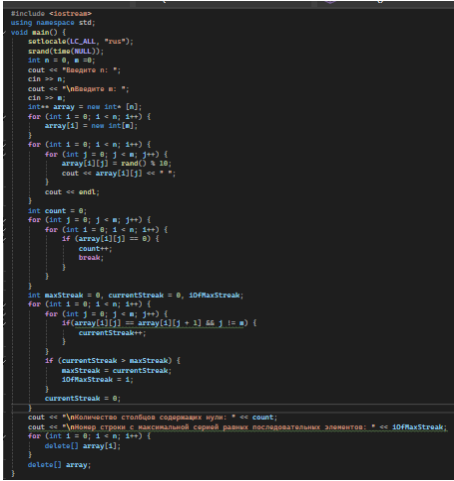
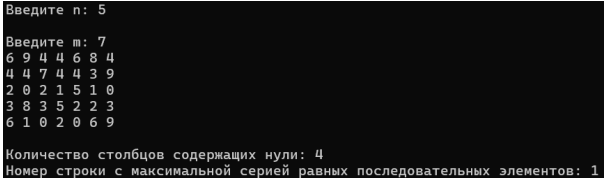
3	<div data-bbox="288 91 635 286" data-label="Text"> <p>3. В программе, записанной справа, демонстрируется использование динамической памяти при работе с двумерным массивом.</p> <p>Выполнить программу с различными размерами исходного массива.</p> </div> <div data-bbox="667 91 1422 887" data-label="Text"> <pre>#include <ctime> #include <iostream> using namespace std; void main() { int size, mult = 1, sum = 0, **A; cout << "Enter size "; cin >> size; srand(time(0)); A = new int*[size]; //выделение динам.памяти for (int i = 0; i < size; i++) A[i] = new int[size]; for (int j = 0; j < size; j++) A[i][j] = 1 + rand() % 10; for (int **ptr = A; ptr != A + size; ++ptr) { for (int* it = *ptr; it != *ptr + size; ++it) { cout << "\t" << *it; if (*it % 2 == 0) { mult *= *it; sum += *it; } } cout << endl; } cout << "sum: " << sum << endl; cout << "mult: " << mult << endl; for(int k=0; k < size; k++) //освобождение памяти delete[] A[k]; delete[] A; }</pre> </div>		
4 7	<div data-bbox="288 913 1422 1037" data-label="List-Group"> <ol style="list-style-type: none"> 1. Задан массив A из n элементов. Найти количество элементов этого массива, больших среднего арифметического всех его элементов. 2. Дана целочисленная прямоугольная матрица. Определить количество строк, не содержащих ни одного нулевого элемента и максимальное из чисел, встречающихся в заданной матрице более одного раза. </div>		
Доп 1	<div data-bbox="288 1059 1422 1182" data-label="List-Group"> <ol style="list-style-type: none"> 1. В одномерном массиве, состоящем из n вещественных элементов, вычислить минимальный элемент массива и сумму элементов, расположенных между первым и последним положительными элементами. 2. Проверить, есть ли в матрице хотя бы одна строка, содержащая отрицательный элемент, и найти ее номер. Все элементы столбца с таким же номером уменьшить вдвое. </div>		
Доп 2 14	<div data-bbox="288 1205 1422 1630" data-label="List-Group"> <ol style="list-style-type: none"> 1. В массиве, состоящем из вещественных элементов, вычислить количество элементов массива, больших некоторого заданного числа C, и произведение элементов массива, расположенных после максимального по модулю элемента. <div data-bbox="288 1294 1422 1507" data-label="Text"> <p style="text-align: center;">104</p> </div> <div data-bbox="288 1507 1422 1574" data-label="Section-Header"> <p style="text-align: center;">Условия задач</p> </div> <ol style="list-style-type: none"> 2. Проверить, есть ли в матрице хотя бы одна строка, содержащая элемент, равный нулю, и найти ее номер. Уменьшить все элементы матрицы на значение первого элемента найденной строки </div>		
Доп 3	<div data-bbox="288 1653 1422 1776" data-label="List-Group"> <table border="1"> <tr> <td data-bbox="288 1653 368 1776">5</td> <td data-bbox="368 1653 1422 1776"> <ol style="list-style-type: none"> 1. В одномерном массиве, состоящем из n вещественных элементов, вычислить количество элементов массива, равных 0, и сумму элементов массива, расположенных после минимального элемента. 2. Дана целочисленная прямоугольная матрица. Определить количество столбцов, содержащих хотя бы один нулевой элемент и номер строки, в которой находится самая длинная серия одинаковых элементов. </td> </tr> </table> </div>	5	<ol style="list-style-type: none"> 1. В одномерном массиве, состоящем из n вещественных элементов, вычислить количество элементов массива, равных 0, и сумму элементов массива, расположенных после минимального элемента. 2. Дана целочисленная прямоугольная матрица. Определить количество столбцов, содержащих хотя бы один нулевой элемент и номер строки, в которой находится самая длинная серия одинаковых элементов.
5	<ol style="list-style-type: none"> 1. В одномерном массиве, состоящем из n вещественных элементов, вычислить количество элементов массива, равных 0, и сумму элементов массива, расположенных после минимального элемента. 2. Дана целочисленная прямоугольная матрица. Определить количество столбцов, содержащих хотя бы один нулевой элемент и номер строки, в которой находится самая длинная серия одинаковых элементов. 		

Номер задания	Скриншот кода	Скриншот результата
1	<pre> #include <stdio.h> #include <stdlib.h> void main() { int* ptr, i, n; printf("Input size of massiv, n<30 \n"); scanf_s("%d", &n); if (!(ptr = (int*)malloc(n * sizeof(int)))) { puts("Not enough memory"); return; } for (i = 0; i < n; i++) { printf("Input element [%d]\n", i + 1); scanf_s("%d", ptr + i); printf_s("\nMassiv: \n", i + 1); for (i = 0; i < n; i++) printf_s("%d ", *(ptr + i)); int max = *ptr; for (int i = 0; i < n; i++) { if (*(ptr + i) > max) { max = *(ptr + i); } } printf_s("\nMax: %d", max); free(ptr); //освобождение динамической памяти } } </pre>	<pre> 7 Input element [5] 4 Input element [6] 3 Input element [7] 53 Input element [8] 35 Input element [9] 64 Input element [10] 24 Input element [11] 75 Input element [12] 35 Input element [13] 4 Input element [14] 3 Input element [15] 87 Massiv: 3 4 6 7 4 3 53 35 64 24 75 35 4 3 87 Max: 87 </pre>
2	<pre> #include <iostream> using namespace std; void main() { setlocale(LC_ALL, "rus"); char* pStr; char c, * pc; int j, length, count = 0; cout << "Введите размер строки: "; cin >> length; pStr = new char[length + 1]; cout << "Введите строку: "; cin >> pStr; cout << "Введите символ: "; cin >> c; pc = &c; for (int j = 0; j < length; j++) { if (pStr[j] == *pc) { count++; } } cout << count << " повторений символа " << c << endl; delete[] pStr; } </pre>	<pre> Введите размер строки: 4 Введите строку: mark Введите символ: k 1 повторений символа k </pre>
		<pre> Введите размер строки: 10 Введите строку: 2332323 Введите символ: 3 4 повторений символа 3 </pre>

Номер задания	Скриншот кода	Скриншот результата
3	<pre> #include <ctime> #include <iostream> using namespace std; void main() { int size, sum = 0, **A; long long mult = 1; cout << "Enter size "; cin >> size; srand(time(0)); A = new int*[size]; //выделение динам. памяти for (int i = 0; i < size; i++) { A[i] = new int[size]; for (int j = 0; j < size; j++) A[i][j] = 1 + rand() % 10; } for (int** ptr = A; ptr != A + size; ++ptr) { for (int* it = *ptr; it != *ptr + size; ++it) { cout << "\t" << *it; if (*it % 2 == 0 && *it != 0) { mult *= *it; sum += *it; } } cout << endl; } cout << "sum: " << sum << endl; cout << "mult: " << mult << endl; for (int k = 0; k < size; k++) //освобождение памяти delete[] A[k]; delete[] A; } </pre>	<pre> Enter size 5 5 5 5 2 5 9 5 5 1 4 3 5 8 10 8 6 7 8 5 8 9 9 6 3 6 sum: 66 mult: 70778880 </pre>
		<pre> Enter size 2 8 4 7 1 sum: 12 mult: 32 </pre>
4 1	<pre> #include <stdio.h> #include <stdlib.h> void main() { int* ptr, i, n, sum = 0, count = 0; printf_s("Input size of massiv, n<30 \n"); scanf_s("%d", &n); if (!(ptr = (int*)malloc(n * sizeof(int)))) //выделение памяти и проверка, { puts("Not enough memory"); //достаточно ли для нее места return; } for (i = 0; i < n; i++) { printf_s("Input element [%d]\n", i + 1); scanf_s("%d", ptr + i); sum += *(ptr + i); } sum /= n; for (int i = 0; i < n; i++) { if (*(ptr + i) > sum) count++; } printf_s("\nMassiv: \n", i); for (i = 0; i < n; i++) printf_s("%d ", *(ptr + i)); printf_s("Amount of elements higher then average sum: %d", count); free(ptr); //освобождение динамической памяти } </pre>	<pre> Input size of massiv, n<30 10 Input element [1] 23 Input element [2] 42 Input element [3] 52 Input element [4] 48 Input element [5] 92 Input element [6] 42 Input element [7] 52 Input element [8] 30 Input element [9] 67 Input element [10] 23 Massiv: 23 42 52 48 92 42 52 30 67 23 Amount of elements higher then average sum: 5 </pre>
4 2	<pre> #include <iostream> using namespace std; void main() { setlocale(LC_ALL, "rus"); srand(time(NULL)); int n, m, count = 0; cout << "Введите n: "; cin >> n; cout << "Введите m: "; cin >> m; int** array = new int*[n]; for (int i = 0; i < n; i++) { array[i] = new int[m]; } bool isZero = false; int max = array[0][0]; cout << "Полученный массив:\n "; for (int i = 0; i < n; i++) { isZero = false; for (int j = 0; j < m; j++) { array[i][j] = (rand() % 10); if (array[i][j] == 0) { isZero = true; } bool isRepeat = false; for (int k = 0; k < m; k++) { for (int o = 0; o < m; o++) { if (array[i][j] == array[k][o] && i != k && j != o) { isRepeat = true; } } if (isRepeat && array[i][j] > max) { max = array[i][j]; } } cout << array[i][j] << " "; } if (!isZero) count++; cout << endl; } cout << "\nКоличество строк без нулей: " << count; cout << "\nМаксимальное число среди повторяющихся: " << max; for (int i = 0; i < n; i++) { delete[] array[i]; } delete[] array; } </pre>	<pre> Введите n: 3 Введите m: 4 Полученный массив: 5 3 4 7 8 6 3 7 0 8 9 1 Количество строк без нулей: 2 Максимальное число среди повторяющихся: 8 </pre>

Номер задания	Скриншот кода	Скриншот результата
Доп 1 1	<pre> #include <stdio.h> #include <stdlib.h> void main() { int* ptr, i, n; printf_s("Input size of massiv, n<30 \n"); scanf_s("%d", &n); if ((ptr = (int*)malloc(n * sizeof(int))) //выделение памяти и проверка, { puts("Not enough memory"); //достаточно ли для нее места return; } int iFirst, iLast; bool isFirst = true; for (i = 0; i < n; i++) { printf_s("Input element [%d]\n", i + 1); scanf_s("%d", ptr + i); } int min = *ptr; for (int i = 0; i < n; i++) { if (*(ptr + i) < min) { min = *(ptr + i); } if (*(ptr + i) > 0 && isFirst == true) { isFirst = false; iFirst = i; } if (*(ptr + i) > 0) { iLast = i; } } int sum = 0; for (int i = iFirst; i <= iLast; i++) { sum += *(ptr + i); } printf_s("Massiv: \n", i + 1); for (i = 0; i < n; i++) printf_s("%d ", *(ptr + i)); printf_s("\nMin element: %d", min); printf_s("\nSum between: %d", sum); free(ptr); //освобождение динамической памяти } </pre>	<pre> Input size of massiv, n<30 10 Input element [1] -2 Input element [2] 3 Input element [3] 5 Input element [4] 2 Input element [5] -4 Input element [6] 3 Input element [7] 3 Input element [8] 3 Input element [9] 8 Input element [10] 3 Massiv: -2 3 5 2 -4 3 3 3 8 3 Min element: -4 Sum between: 23 </pre>
Доп 1 2	<pre> #include <iostream> using namespace std; void main() { setlocale(LC_ALL, "rus"); srand(time(NULL)); int iOfStr = -1; int n = 0; cout << "Введите размер матрицы nхn: "; cin >> n; int** array = new int* [n]; for (int i = 0; i < n; i++) { array[i] = new int[n]; } cout << "Исходная матрица:\n"; for (int i = 0; i < n; i++) { for (int j = 0; j < n; j++) { array[i][j] = (rand() % 30) - 1; if (array[i][j] < 0) { iOfStr = i; } cout << array[i][j] << " "; } cout << endl; } for (int i = 0; i < n; i++) { array[i][iOfStr] /= 2; } cout << "Полученная матрица: \n"; for (int i = 0; i < n; i++) { for (int j = 0; j < n; j++) { cout << array[i][j] << " "; } cout << endl; } for (int i = 0; i < n; i++) { delete array[i]; } delete[] array; } </pre>	<pre> Введите размер матрицы nхn: 4 Исходная матрица: 11 5 11 27 23 17 8 21 3 6 17 16 8 1 1 -1 Полученная матрица: 11 5 11 13 23 17 8 10 3 6 17 8 8 1 1 0 </pre>

Номер задания	Скриншот кода	Скриншот результата
Доп 2 1	<pre> #include <stdio.h> #include <stdlib.h> #include <math.h> int main() { int i, n; double* ptr; printf_s("Input size of array, n < 30: "); scanf_s("%d", &n); ptr = (double*)malloc(n * sizeof(double)); if (!ptr) { puts("Not enough memory"); return 1; } for (i = 0; i < n; i++) { printf_s("Input element [%d]: ", i + 1); scanf_s("%lf", ptr + i); } int c; printf_s("\nInput c: "); scanf_s("%d", &c); int count = 0, imax = 0; double max = fabs(ptr[0]); for (i = 0; i < n; i++) { if (fabs(ptr[i]) > max) { max = fabs(ptr[i]); imax = i; } if (ptr[i] > c) { count++; } } double sum = 0; for (i = imax; i < n; i++) { sum += ptr[i]; } printf_s("Massiv: \n"); for (i = 0; i < n; i++) { printf_s("%.2f ", ptr[i]); } printf_s("\nCount of elements greater than c: %d", count); printf_s("\nSum of elements after abs(max): %.2f", sum); free(ptr); return 0; } </pre>	<pre> Input size of array, n < 30: 10 Input element [1]: 23.32 Input element [2]: 65.34 Input element [3]: -13.2 Input element [4]: -49.12 Input element [5]: 20.12 Input element [6]: 56.3 Input element [7]: -96.85 Input element [8]: 15 Input element [9]: 90.2 Input element [10]: 3 Input c: 20 Massiv: 23.32 65.34 -13.20 -49.12 20.12 56.30 -96.85 15.00 90.20 3.00 Count of elements greater than c: 5 Sum of elements after abs(max): 11.35 </pre>
Доп 2 2	<pre> #include <iostream> using namespace std; void main() { setlocale(LC_ALL, "rus"); srand(time(NULL)); int n = 0; cout << "Введите размерность матрицы nxn: "; cin >> n; int** array = new int* [n]; for (int i = 0; i < n; i++) { array[i] = new int[n]; } int iZero = -1; cout << "\nИсходная матрица:\n "; for (int i = 0; i < n; i++) { for (int j = 0; j < n; j++) { array[i][j] = rand() % 10; if (array[i][j] == 0 && iZero == -1) { iZero = i; } cout << array[i][j] << " "; } cout << endl; } int number = array[iZero][0]; cout << "\nПолученная матрица:\n"; for (int i = 0; i < n; i++) { for (int j = 0; j < n; j++) { array[i][j] -= number; cout << array[i][j] << " "; } cout << endl; } for (int i = 0; i < n; i++) { delete[] array[i]; } delete[] array; } </pre>	<pre> Введите размерность матрицы nxn: 5 Исходная матрица: 6 9 9 5 3 2 4 6 8 1 5 6 7 9 6 3 4 6 4 7 9 7 0 3 2 Полученная матрица: -3 0 0 -4 -6 -7 -5 -3 -1 -8 -4 -3 -2 0 -3 -6 -5 -3 -5 -2 0 -2 -9 -6 -7 </pre>
Доп 3 1	<pre> 4 int main() { 5 int i, n; 6 double* ptr; 7 printf_s("Input size of array, n < 30: "); 8 scanf_s("%d", &n); 9 ptr = (double*)malloc(n * sizeof(double)); 10 if (!ptr) { 11 puts("Not enough memory"); 12 return 1; 13 } 14 15 for (i = 0; i < n; i++) { 16 printf_s("Input element [%d]: ", i + 1); 17 scanf_s("%lf", ptr + i); 18 } 19 20 int count = 0, imin = 0; 21 double min = *ptr, sum = 0; 22 for (i = 0; i < n; i++) { 23 if (*ptr + i == 0) { 24 count++; 25 } 26 if (*ptr + i < min) { 27 min = *ptr + i; 28 imin = i; 29 } 30 } 31 for (i = imin; i < n; i++) { 32 sum += ptr[i]; 33 } 34 35 printf_s("Massiv: \n"); 36 for (i = 0; i < n; i++) { 37 printf_s("%.2f ", ptr[i]); 38 } 39 printf_s("\nCount of elements equals zero: %d", count); 40 printf_s("\nSum of elements after min: %.2f", sum); 41 free(ptr); 42 return 0; 43 } </pre>	<pre> Input size of array, n < 30: 7 Input element [1]: 43.2 Input element [2]: 0 Input element [3]: 24.1 Input element [4]: 0 Input element [5]: 65.34 Input element [6]: 12 Input element [7]: -1.3 Massiv: 43.20 0.00 24.10 0.00 65.34 12.00 -1.30 Count of elements equals zero: 2 Sum of elements after min: -1.30 </pre>

Номер задания	Скриншот кода	Скриншот результата
Доп 3 2		

Номер задания	Описание шагов	Блок-схема