**Белорусский государственный технологический университет**

**Факультет информационных технологий**

**Кафедра ПИ**

Лабораторная работа № 15

По дисциплине «Основы алгоритмизации и программирования»

На тему «Динамическое выделение памяти»

Выполнил:  
Студент 1 курса, 10 группа  
Короткевич Артём Сергеевич  
Преподаватель: асс. Харланович А.В

2024, Минск

**Задание 2**Изучить способы выделения динамической памяти для *одномерного* *массива*, выполнив программу на *языке* ***С++***, записанную в правой части.  
Опробовать работу программы с разными значениями вводимых символов. **Код**

using namespace std;

void main() {

setlocale(LC\_CTYPE, "Russian");

char\* pStr; // указатель для динамического массива

char c, \* pC;

int j, length, count = 0;

cout << "Введите размер строки" << endl;

cin >> length;

pStr = new char[length + 1]; // выделение динамической памяти

cout << "Введите строку" << endl;

cin >> pStr;

cout << "Введите символ " << endl;

cin >> c;

pC = &c; // указатель на символ

for (j = 0; j < length; j++) { // подсчет повторений символа

if (pStr[j] == \*pC) {

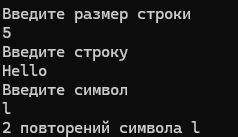
count++;

}

}

cout << count << " повторений символа " << c << endl;

delete[] pStr; // освобождение памяти  
}

**Результат**  


**Задание 3**  
В программе, записанной справа, демонстрируется использование динамической памяти при работе с *двумерным массивом*.  
Выполнить программу с различными размерами исходного массива.

**Код**

#include <ctime>  
#include <iostream>

using namespace std;

int main() {

int size, mult = 1, sum = 0, \*\* A;

cout << "Enter size: ";

cin >> size;

srand(time(0));

A = new int\* [size];

for (int i = 0; i < size; i++)

A[i] = new int[size];

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

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;

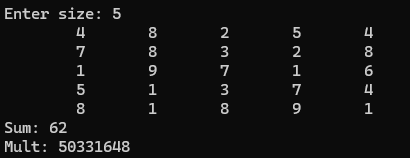
for (int i = 0; i < size; i++)

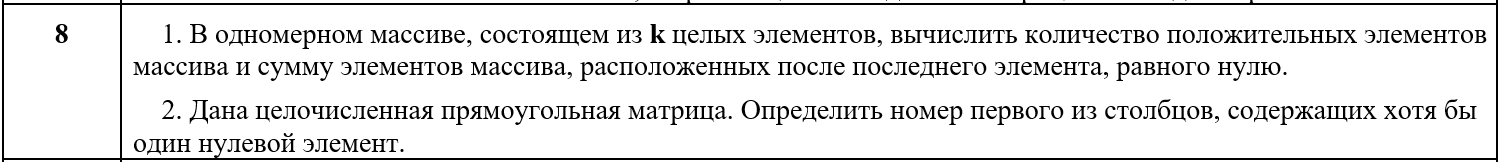
delete[] A[i];

delete[] A;

return 0;

}

**Результат**  
  
  
  
**Задание 4**В соответствии со своим вариантом написать программы с использованием ***динамических*** массивов для условий задач из таблицы. Начальные размерности массивов ввести с клавиатуры. Первое задание выполнить с использованием ***функций*** для выделения динамической памяти на языке С, второе − с использованием операторов **new** и **delete**  на языке **С++**.   
  
**Вариант 8**

****

**Задание 1**

**Код**

#include <iostream>

using namespace std;

int main()

{

setlocale(LC\_ALL, "Russian");

int k;

cout << "Введите длину массива: ";

cin >> k;

int\* A = new int[k]; //создаем динамический массив

cout << "Введите элементы массива: ";

for (int i = 0; i < k; i++)

{

cin >> A[i];

}

// инициализируем переменные для счетчика положительных элементов, поиска суммы элементов после 0, и переменную для поиска 0

int count = 0;

int lastzero = 0;

bool zero = false;

// с помощью цикла осуществляем поиск, сумму и подсчет

for (int i = 0; i < k; i++)

{

if (A[i] > 0) {

count++;

}

if (A[i] == 0) {

zero = true;

lastzero = 0;

}

if (zero) {

lastzero += A[i];

}

}

cout << "Количество положительных элементов: " << count << endl;

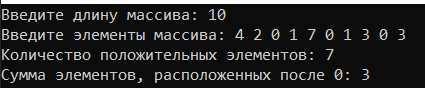
cout << "Сумма элементов, расположенных после 0: " << lastzero << endl;

delete[] A; // очищаем массив из памяти

return 0;

}

**Результат**

****

**Задание 2**

**Код**

#include <iostream>

using namespace std;

int main()

{

setlocale(LC\_ALL, "Russian");

int n, m;

cout << "Введите количество строк матрицы: ";

cin >> n;

cout << "Введите количество столбцов матрицы: ";

cin >> m;

int\*\* A = new int\* [n]; // выделение динамической памяти

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

{

A[i] = new int[n];

}

cout << "Введите матрицу: " << endl;

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

{

for (int j = 0; j < m; j++)

{

cout << "A(" << i + 1 << ", " << j + 1 << "): ";

cin >> A[i][j];

}

}

cout << "Исходная матрица: " << endl;

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

{

cout << "[";

for (int j = 0; j < m; j++)

{

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

}

cout << "]" << endl;

}

// инициализируем переменную, которая будет искать комер столбика, содержащего хотя бы 1 нулевой элемент

int stolb = 0;

for (int i = 0; i < m; i++)

{

int a = 0;

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

{

if (A[j][i] == 0)

{

a++;

break;

}

}

if (a > 0)

{

stolb = i + 1;

break;

}

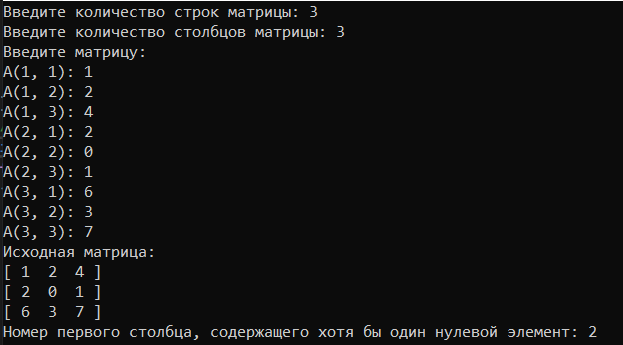
}

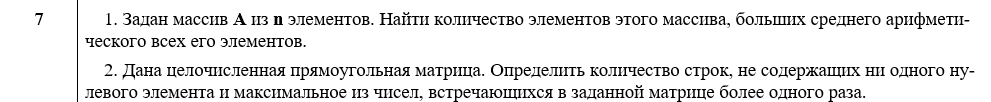
cout << "Номер первого столбца, содержащего хотя бы один нулевой элемент: " << stolb << endl;

delete[] A; // освобождение динамической памяти

}

**Результат**

****

**Дополнительные задания  
  
Вариант 7**  
 **Задание 1**  
**Код**

#include <iostream>

using namespace std;

int main()

{

setlocale(LC\_ALL, "Russian");

int\* ptr, n;

cout << "Введите размер массива: ";

cin >> n;

if (!(ptr = (int\*)malloc(n \* sizeof(int)))) // выделение памяти и проверка

{ // достаточно ли для нее места

cout << "Недостаточно памяти";

return 0;

}

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

{

cout << "Введите элементы массива " << i + 1 << " : ";

cin >> \*(ptr + i);

}

int sum = 0;

cout << "Введенный массив: ";

for (int i = 0; i < n; i++) // находим сумму элементов массива

{

cout << \*(ptr + i) << ' ';

sum += \*(ptr + i);

}

double number = sum / n; //инициализируем переменную среднего арифметического и счетчик

int count = 0;

for (int i = 0; i < n; i++) //с помощью цикла определяем больше или меньше элемент среднего арифметического

{

if (\*(ptr + i) > number) {

count++;

}

else

continue;

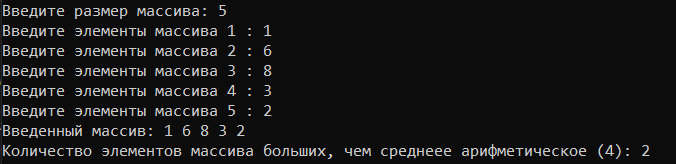
}

cout << endl;

cout << "Количество элементов массива больших, чем среднеее арифметическое (" << number << "): " << count;

free(ptr); // освобождение динамической памяти

}

**Результат**  
  
  
**Задание 2  
  
Код**

#include <iostream>  
using namespace std;

int main()

{

setlocale(LC\_ALL, "Russian");

int n, m;

cout << "Введите количество строк матрицы: ";

cin >> n;

cout << "Введите количество столбцов матрицы: ";

cin >> m;

int\*\* A = new int\* [n]; // выделение динамической памяти

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

{

A[i] = new int[n];

}

cout << "Введите матрицу: " << endl;

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

{

for (int j = 0; j < m; j++)

{

cout << "A(" << i + 1 << ", " << j + 1 << "): ";

cin >> A[i][j];

}

}

cout << "Исходная матрица: " << endl;

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

{

cout << "[";

for (int j = 0; j < m; j++)

{

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

}

cout << "]" << endl;

}

int str = 0; // инициализируем переменную, которая будет искать строку, не содержащую 0

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

{

int a = 0;

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

{

if (A[i][j] == 0)

{

a++;

}

}

if (a == 0)

str++;

}

cout << "Kоличество строк, не содержащих ни одного нулевого элемента: " << str << endl;

int maxCount = 0, maxItem = 0; // инициализируем переменные, которые будут находить и считать наиболее часто встречающийся элемент матрицы

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

{

for (int j = 0; j < m; j++)

{

int count = 0;

for (int p = 0; p < n; p++)

{

for (int q = 0; q < m; q++)

{

if (A[i][j] == A[p][q])

{

count++;

}

}

}

if (count > maxCount)

{

maxCount = count;

maxItem = A[i][j];

}

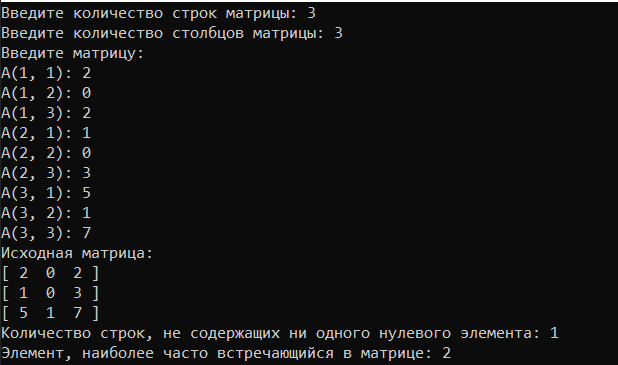
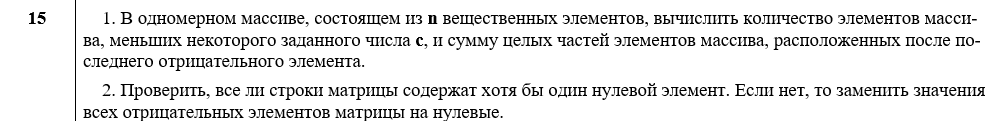
}

}

cout << "Элемент, наиболее часто встречающийся в матрице: " << maxItem << endl;

delete[] A; // освобождение динамической памяти

}

**Результат** **Вариант 15  
  
  
  
  
  
Задание 1**  
**Код**

#include <iostream>  
#include <cmath> // для использования функции floor()

using namespace std;

int main() {

setlocale(LC\_ALL, "Russian");

int n;

double c;

cout << "Введите количество элементов массива (n): ";

cin >> n;

double\* array = new double[n]; // создаём динамический массив

cout << "Введите элементы массива: ";

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

cin >> array[i];

}

cout << "Введите число c: ";

cin >> c;

int countless = 0; // счет количества элементов, меньших c

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

if (array[i] < c) {

++countless;

}

}

int NegativeIndex = -1; // нахождение индекса последнего отрицательного элемента

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

if (array[i] < 0) {

NegativeIndex = i;

}

}

int sum= 0; // сумма целых частей элементов после последнего отрицательного элемента

if (NegativeIndex != -1) { // проверка, был ли отрицательный элемент

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

sum += static\_cast<int>(floor(array[i]));// отбрасываем дробную часть

}

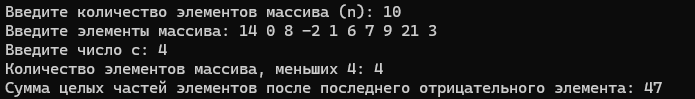
}

cout << "Количество элементов массива, меньших " << c << ": " << countless << endl;

cout << "Сумма целых частей элементов после последнего отрицательного элемента: " << sum << endl;

delete[] array; // освобождение динамеческой памяти

return 0;  
}

**Результат  
  
**  
**Задание 2**  
  
**Код**

#include <iostream>  
#include <iostream>  
using namespace std;

int main() {

setlocale(LC\_ALL, "Russian");

int row, column;

cout << "Введите количество строк (row): ";

cin >> row;

cout << "Введите количество столбцов (column): ";

cin >> column;

int\*\* matrix = new int\* [row]; // создание динамической матрицы

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

matrix[i] = new int[column];

}

cout << "Введите элементы матрицы:" << endl;

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

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

cin >> matrix[i][j];

}

}

// проверка, содержат ли все строки хотя бы один нулевой элемент

bool allRowsHaveZero = true; // флаг, что все строки содержат хотя бы один ноль

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

bool rowHasZero = false;

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

if (matrix[i][j] == 0) {

rowHasZero = true;

break;

}

}

if (!rowHasZero) {

allRowsHaveZero = false;

break;

}

}

if (!allRowsHaveZero) { // если не все строки содержат хотя бы один нулевой элемент, заменяем отрицательные элементы на нули

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

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

if (matrix[i][j] < 0) {

matrix[i][j] = 0;

}

}

}

}

cout << "Изменённая матрица:" << endl;

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

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

cout << matrix[i][j] << "\t";

}

cout << endl;

}

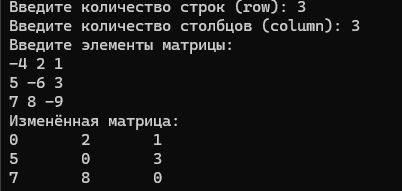
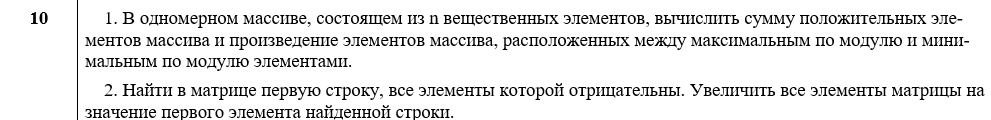
for (int i = 0; i < row; ++i) {

delete[] matrix[i];

}

delete[] matrix; // освобождение динамической памяти

return 0;  
}

**Результат  
  
  
  
  
  
  
Вариант 10  
  
  
  
Задание 1**  
**Код**

#include <iostream>  
using namespace std;

int main() {

setlocale(LC\_ALL, "Russian");

int n;

cout << "Введите количество элементов массива: ";

cin >> n;

double\* arr = new double[n]; // создаем динамический массив

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

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

cin >> arr[i];

}

double sumPositive = 0; // сумма положительных элементов

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

if (arr[i] > 0) {

sumPositive += arr[i];

}

}

int maxIndex = 0, minIndex = 0; // поиск максимального и минимального по модулю элементов

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

if (abs(arr[i]) > abs(arr[maxIndex])) {

maxIndex = i;

}

if (abs(arr[i]) < abs(arr[minIndex])) {

minIndex = i;

}

}

double product = 1; // вычисление произведения между maxIndex и minIndex

if (maxIndex > minIndex) {

swap(maxIndex, minIndex);

}

for (int i = maxIndex + 1; i < minIndex; ++i) {

product \*= arr[i];

}

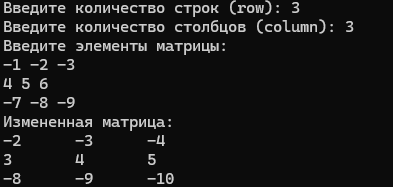
cout << "Сумма положительных элементов: " << sumPositive << endl;

cout << "Произведение элементов между максимальным и минимальным по модулю: " << product << endl;

delete[] arr; // освобождаем динамическую память

return 0;

}

**Результат**  
  
  
**Задание 2**  
  
**Код**

#include <iostream>  
using namespace std;

int main() {

setlocale(LC\_ALL, "Russian");

int row, column;

cout << "Введите количество строк (row): ";

cin >> row;

cout << "Введите количество столбцов (column): ";

cin >> column;

// Создание динамической матрицы

int\*\* matrix = new int\* [row];

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

matrix[i] = new int[column];

}

// Ввод элементов матрицы

cout << "Введите элементы матрицы:" << endl;

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

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

cin >> matrix[i][j];

}

}

// Поиск первой строки, где все элементы отрицательны

int firstNegRow = -1;

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

bool allNegative = true;

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

if (matrix[i][j] >= 0) {

allNegative = false;

break;

}

}

if (allNegative) {

firstNegRow = i;

break;

}

}

// Если такая строка найдена, увеличиваем элементы матрицы на значение первого элемента строки

if (firstNegRow != -1) {

int increment = matrix[firstNegRow][0];

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

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

matrix[i][j] += increment;

}

}

}

// Вывод измененной матрицы

cout << "Измененная матрица:" << endl;

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

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

cout << matrix[i][j] << "\t";

}

cout << endl;

}

// Очистка памяти

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

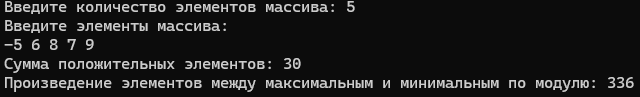
delete[] matrix[i];

}

delete[] matrix;

return 0;

}

**Результат  
  
**