Белорусский государственный технологический университет

Факультет информационных технологий

Кафедра программной инженерии

 Лабораторная работа 15

По дисциплине «Основы алгоритмизации и программирования»

На тему «Динамическое выделение памяти»

Выполнил:

Студент 1 курса 6 группы

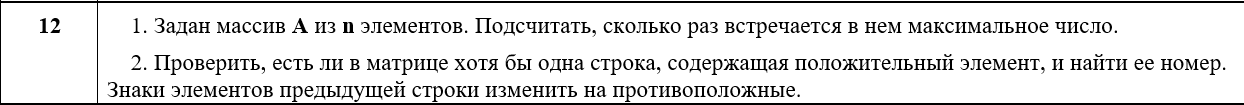
Романов Игорь Вячеславович

Преподаватель: асс. Андронова М.В.

2023, Минск

Задание 4

Вариант 12 (основной)



1.

#include <iostream>

#include <ctime> //для работы с функцией time

using namespace std;

void main() {

setlocale(LC\_CTYPE, "Russian");

srand(time(NULL)); //устанавливает новую точку отсчета для функции rand()

int\* A; //ввод массива

int n = 0, i, imax = -1, counter\_imax = 0; //инициализация переменных

cout << "Введите размер строки: "; cin >> n;

if (!(A = (int\*)malloc(n \* sizeof(int)))) //выделение памяти и проверка, достаточно ли для нее места

{

cout << "Недостаточно памяти";

return;

}

cout << "\nЭлементы массива: " << endl; //заполнение массива случайными числами и его вывод на экран; нахождение максимального элемента массива

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

\*(A + i) = rand() % 21;

cout << \*(A + i) << " ";

if (\*(A + i) > imax) {

imax = \*(A + i);

}

}

for (i = 0; i < n; i++) { //расчет кол-ва повторениий максимального элемента в массиве

if (\*(A + i) == imax) {

counter\_imax++;

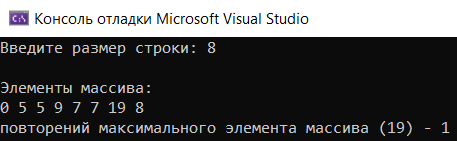
}

}

cout << "\nповторений максимального элемента массива (" << imax << ") - " << counter\_imax; //вывод числа повторениц imax

free(A); //освобожедние памяти

}



2.

#include <iostream>

#include <ctime> //для работы с функцией time

using namespace std;

void main() {

setlocale(LC\_CTYPE, "Russian");

srand(time(NULL)); //устанавливает новую точку отсчета для функции rand()

int i, j, rows = 0, cols = 0, check = 0, index = 0; //инициализация переменных

cout << "Введите размер строк матрицы: "; cin >> rows;

cout << "Введите размер столбцов матрицы: "; cin >> cols;

//выделение памяти для матрицы A

int\*\* A = new int\*[rows]; //выделение памяти для строк

for (i = 0; i < rows; i++) {

A[i] = new int[cols]; //для столбцов

}

for (i = 0; i < rows; i++) { //заполнение матрицы случайными числами и вывод ее элементов

for (j = 0; j < cols; j++) {

A[i][j] = -11 + rand() % 21;

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

}

cout << "\n"; //переход на новую строку

}

for (i = 0; i < rows; i++) { //проверка, есть ли в строке положительные элементы

for (j = 0; j < cols; j++) {

if (A[i][j] > 0) { //если есть, то выйти из подцикла

check++;

index = i;

break;

}

}

if (check) { //если есть полож. элементы, то выйти из цикла

break;

}

}

cout << "Номер строки с положительным элементом: " << index; //вывод номер строки с полож. элементом

if (check) { //замена элементов предыдущей строки на противоположные

for (j = 0; j < cols; j++) {

A[index - 1][j] \*= -1;

}

}

cout << "\nПреобразованная матрица массив:" << endl; //вывод на экран преобразованной матрицы

for (i = 0; i < rows; i++) {

cout << "\n";

for (j = 0; j < cols; j++) {

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

}

}

//освобождение памяти

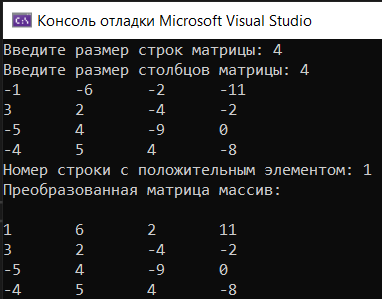
for (i = 0; i < rows; i++) {

delete [] A[i];

}

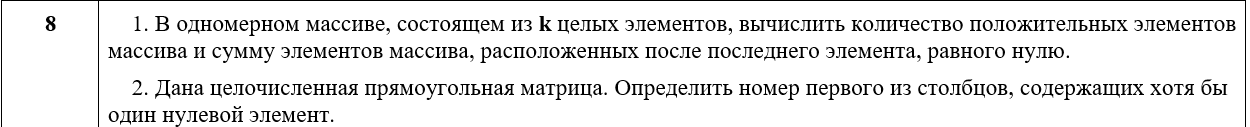
delete[] A;

}



Дополнительные задания:

Вариант 8



1.

#include <iostream>

#include <ctime> //для работы с функцией time

using namespace std;

void main() {

setlocale(LC\_CTYPE, "Russian");

srand(time(NULL)); //устанавливает новую точку отсчета для функции rand()

int\* A; //ввод массива

int k = 0, i, count\_pos = 0, index\_null = -1, sum = 0; //инициализация переменных

cout << "Введите размер строки: "; cin >> k;

if (!(A = (int\*)malloc(k \* sizeof(int)))) //выделение памяти и проверка, достаточно ли для нее места

{

cout << "Недостаточно памяти";

return;

}

cout << "\nЭлементы массива: " << endl;

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

\*(A + i) = -3 + rand() % 11; //заполнение массива случайными числами и его вывод на экран

cout << \*(A + i) << " ";

if (\*(A + i) == 0) { //нахождения номера последнего вхождения 0 в строку

index\_null = i;

}

if (\*(A + i) > 0) { //расчет положительных чисел в строке

count\_pos++;

}

}

for (i = index\_null; i < k; i++) { //расчет суммы элементов после номера последнего вхождения 0 в строку

sum += \*(A + i);

}

//вывод соответствующих сообщений на экран

cout << "\nКоличество положительных чисел в строке: " << count\_pos;

if (index\_null == -1) {

cout << "\nВ строке нет 0";

}

else {

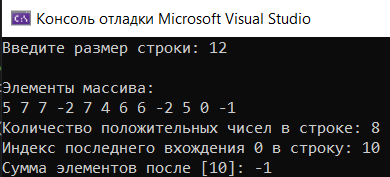
cout << "\nИндекс последнего вхождения 0 в строку: " << index\_null;

cout << "\nСумма элементов после " << index\_null << ": " << sum; //вывод числа повторениц imax

}

free(A); //освобождение памяти

}



2.

#include <iostream>

#include <ctime> //для работы с функцией time

using namespace std;

void main() {

setlocale(LC\_CTYPE, "Russian");

srand(time(NULL)); //устанавливает новую точку отсчета для функции rand()

int i, j, rows = 0, cols = 0, check = 1, index = 0; //инициализация переменных

cout << "Введите размер строк матрицы: "; cin >> rows;

cout << "Введите размер столбцов матрицы: "; cin >> cols;

//выделение памяти для матрицы A

int\*\* A = new int\*[rows]; //выделение памяти для строк

for (i = 0; i < rows; i++) {

A[i] = new int[cols]; //для столбцов

}

for (i = 0; i < rows; i++) { //заполнение матрицы случайными числами и вывод ее элементов

for (j = 0; j < cols; j++) {

A[i][j] = -11 + rand() % 21;

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

}

cout << "\n"; //переход на новую строку

}

for (i = 0; i < rows; i++) { //проверка на нулевой элемент в строке

for (j = 0; j < cols; j++) {

if (A[i][j] == 0 ) { //если есть нулевой элемент, то выйти из подцикла

index = ++j;

check++;

break;

}

}

if (!check) { //если есть нулевой элемент, то выйти из цикла

break;

}

check = 0; //по умолчанию устанавливается, что в строке нет нулевых элементов

}

cout << "Номер первого столбца, содержащий хотя бы один нулевой элемент: " << index; //вывод соответствующего сообщения

//освобождение памяти

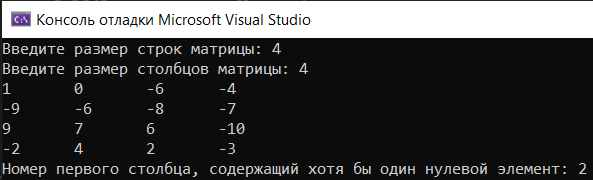
for (i = 0; i < rows; i++) {

delete [] A[i];

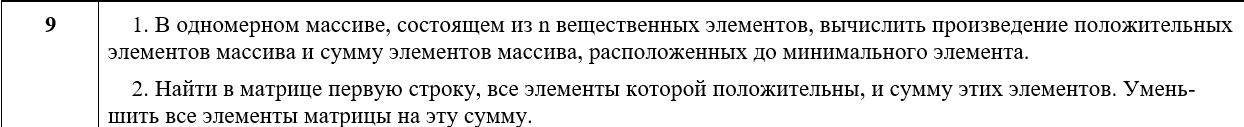
}

delete[] A;

}



Вариант 9



1.

#include <iostream>

#include <ctime> //для работы с функцией time

using namespace std;

void main() {

setlocale(LC\_CTYPE, "Russian");

srand(time(NULL)); //устанавливает новую точку отсчета для функции rand()

int\* A; //ввод массива

int n = 0, i, product = 1, imin = 99999, sum = 0, index = -1, check\_pos = 0; //инициализация переменных

cout << "Введите размер строки: "; cin >> n;

if (!(A = (int\*)malloc(n \* sizeof(int)))) //выделение памяти и проверка, достаточно ли для нее места

{

cout << "Недостаточно памяти";

return;

}

cout << "\nЭлементы массива: " << endl;

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

\*(A + i) = -3 + rand() % 11; //заполнение массива случайными числами и его вывод на экран

cout << \*(A + i) << " ";

if (\*(A + i) < imin) { //нахождения минимального элемента в строке

imin = \*(A + i);

index = i;

}

if (\*(A + i) > 0) { //расчет произведения положительных чисел в строке

check\_pos++;

product \*= \*(A + i);

}

}

for (i = 0; i < index; i++) { //расчет суммы элементов до вхождения минимального элемента строки в строку

sum += \*(A + i);

}

//вывод соответствующих сообщений на экран

if (!check\_pos) {

cout << "В строке нет положительных элементов";

}

else {

cout << "\nПроизведение положительных чисел: " << product;

}

if (index == 0) {

cout << "\nМинимальное число (" << imin << ") находится первым в строке";

}

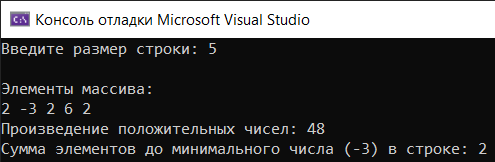
else {

cout << "\nСумма элементов до минимального числа (" << imin << ") в строке: " << sum;

}

free(A); //освобождение памяти

}



2.

#include <iostream>

#include <ctime> //для работы с функцией time

using namespace std;

void main() {

setlocale(LC\_CTYPE, "Russian");

srand(time(NULL)); //устанавливает новую точку отсчета для функции rand()

int i, j, rows = 0, cols = 0, check = 1, sum = 0; //инициализация переменных

cout << "Введите размер строк матрицы: "; cin >> rows;

cout << "Введите размер столбцов матрицы: "; cin >> cols;

//выделение памяти для матрицы A

int\*\* A = new int\*[rows]; //выделение памяти для строк

for (i = 0; i < rows; i++) {

A[i] = new int[cols]; //для столбцов

}

for (i = 0; i < rows; i++) { //заполнение матрицы случайными числами и вывод ее элементов

for (j = 0; j < cols; j++) {

A[i][j] = -5 + rand() % 16;

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

}

cout << "\n"; //переход на новую строку

}

for (i = 0; i < rows; i++) { //проверка на положительные элементы в строке

check = 1; //по умолчанию устанавливается, что в строке нет нулевых элементов

for (j = 0; j < cols; j++) {

if (A[i][j] <= 0 ) { //если есть неположительный элемент,

check = 0;

sum = 0; //обнулить сумму

break; //то выйти из подцикла

}

sum += A[i][j]; //увеличивать сумму

}

if (check) { //если все элементы положительные, то выйти из цикла

break;

}

}

if (check) { //если в матрице есть строки только с положительными элементами

cout << "Сумма положительных элементов строки: " << sum << endl;

for (i = 0; i < rows; i++) {

for (j = 0; j < cols; j++) {

A[i][j] -= sum;

}

}

for (i = 0; i < rows; i++) { //вывод преобразованной матрицы

for (j = 0; j < cols; j++) {

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

}

cout << "\n"; //переход на новую строку

}

}

else { //в противном случае

cout << "В матрице нет строк только с положительными элементами";

}

//освобождение памяти

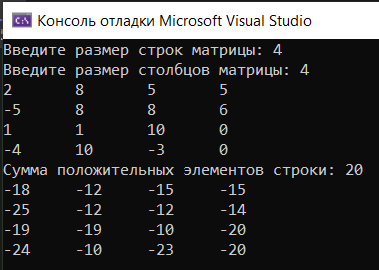
for (i = 0; i < rows; i++) {

delete [] A[i];

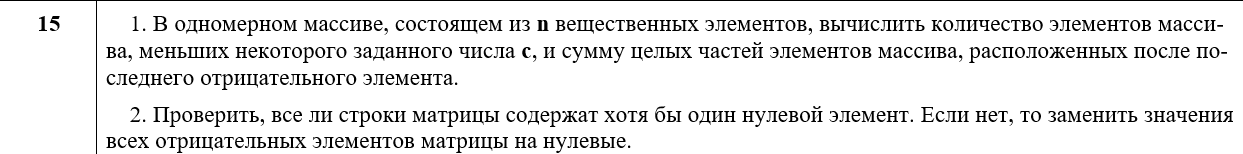
}

delete[] A;

}



Вариант 15



1.

#include <iostream>

#include <ctime> //для работы с функцией time

using namespace std;

void main() {

setlocale(LC\_CTYPE, "Russian");

srand(time(NULL)); //устанавливает новую точку отсчета для функции rand()

int\* A; //ввод массива

int n = 0, i, c, counter\_less = 0, index\_neg = -1, sum = 0; //инициализация переменных

cout << "Введите размер строки: "; cin >> n;

cout << "Введите число c: "; cin >> c;

if (!(A = (int\*)malloc(n \* sizeof(int)))) //выделение памяти и проверка, достаточно ли для нее места

{

cout << "Недостаточно памяти";

return;

}

cout << "\nЭлементы массива: " << endl;

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

\*(A + i) = -3 + rand() % 11; //заполнение массива случайными числами и его вывод на экран

cout << \*(A + i) << " ";

if (\*(A + i) < 0) { //нахождения номера последнего вхождения отрицательного элемента в строку

index\_neg = i;

}

if (\*(A + i) < c) { //расчет чисел меньших, чем введенное c

counter\_less++;

}

}

for (i = index\_neg + 1; i < n; i++) { //расчет суммы элементов после последнего вхождения отрицательного элемента строки в строку

sum += \*(A + i);

}

//вывод соответствующих сообщений на экран

if (index\_neg == -1) {

cout << "\nВ строке нет отрицательных чисел";

}

else if (index\_neg == n - 1) {

cout << "\nПоследний отрицательный элемент строки (" << A[index\_neg] << ") находится в конце строки";

}

else {

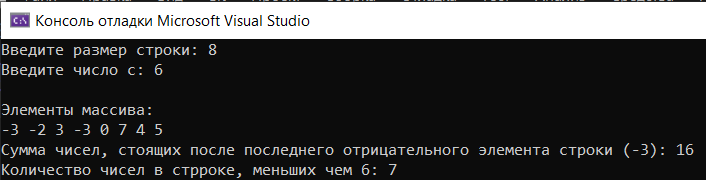
cout << "\nСумма чисел, стоящих после последнего отрицательного элемента строки (" << A[index\_neg] << "): " << sum;

}

cout << "\nКоличество чисел в стрроке, меньших чем " << c << ": " << counter\_less;

free(A); //освобождение памяти

}



2.

#include <iostream>

#include <ctime> //для работы с функцией time

using namespace std;

void main() {

setlocale(LC\_CTYPE, "Russian");

srand(time(NULL)); //устанавливает новую точку отсчета для функции rand()

int i, j, rows = 0, cols = 0, check = 0; //инициализация переменных

cout << "Введите размер строк матрицы: "; cin >> rows;

cout << "Введите размер столбцов матрицы: "; cin >> cols;

//выделение памяти для матрицы A

int\*\* A = new int\*[rows]; //выделение памяти для строк

for (i = 0; i < rows; i++) {

A[i] = new int[cols]; //для столбцов

}

for (i = 0; i < rows; i++) { //заполнение матрицы случайными числами и вывод ее элементов

for (j = 0; j < cols; j++) {

A[i][j] = -5 + rand() %6;

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

}

cout << "\n"; //переход на новую строку

}

cout << endl;

for (i = 0; i < rows; i++) { //проверка, есть ли хотя бы один 0 в строке

check = 0; //по умолчанию устанавливается, что в строке нет нулевых элементов

for (j = 0; j < cols; j++) {

if (A[i][j] == 0 ) { //если есть нулевой элемент, то увеличить check

check++;

}

}

if (!check) { //если в строке нет нулевых элементов, то выйти из цикла

break;

}

}

if (!check) { //если в строке нет нулевых элементов, то обнулить матрицу

for (i = 0; i < rows; i++) { //вывод преобразованной матрицы

for (j = 0; j < cols; j++) {

cout << 0 << "\t";

}

cout << "\n"; //переход на новую строку

}

}

else { //в противном случае вывести сообщение

cout << "В каждой строке есть хотя бы один 0";

}

//освобождение памяти

for (i = 0; i < rows; i++) {

delete [] A[i];

}

delete[] A;

}

