Белорусский государственный технологический университет

Факультет информационных технологий

Кафедра программной инженерии

 Лабораторная работа 15

По дисциплине «Основы алгоритмизации и программирования»

На тему «Динамическое выделение памяти»

 Выполнила:

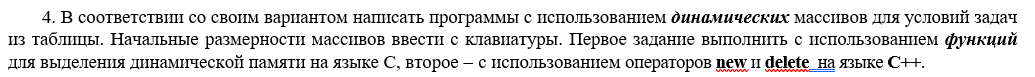
Студентка1 курса 6 группы

Альшевская Алина Михайловна

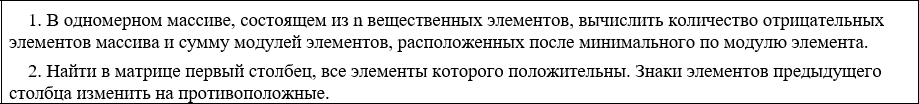
Преподаватель: асс. Андронова М.В.

2023, Минск

Задание 4



Вариант 1



#include <iostream>

#include <Windows.h>

#include <ctime>

using namespace std;

int main()

{

SetConsoleOutputCP(1251);

int n, minIndex = 0, counter = 0;

double min, sum = 0.0;

double rmn = -20, rmx = 20;

cout << "Введите размер массива: ";

cin >> n;

double\* array = (double\*)malloc(n \* sizeof(double));

srand((unsigned)time(NULL));

// Создаем массив случайных чисел и сразу выводим его

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

\*(array + i) = ((double)rand() / (double)RAND\_MAX) \* (rmx - rmn) + rmn;

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

}

cout << "\n";

min = abs(\*(array)); // За минимальное число берем первое число массива

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

if (abs(\*(array + i)) < min) { // Если число меньше текущего минимального, сохраняем его как минимальное и его индекс

min = \*(array + i);

minIndex = i;

}

// Знаходим количество отрицательных чисел

if (\*(array + i) < 0) {

counter++;

}

}

// От индекса минимального значения до конца

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

// Находим сумму модулей

sum += abs(\*(array + i));

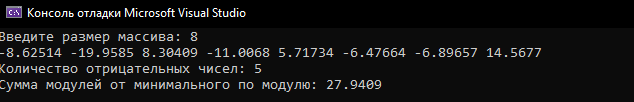
}

cout << "Количество отрицательных чисел: " << counter<<endl;

cout << "Сумма модулей от минимального по модулю: " << sum<<endl;

free(array);

}



#include <iostream>

#include <Windows.h>

using namespace std;

int main()

{

SetConsoleOutputCP(1251);

int rows, cols, rmn = -3, rmx = 3, columnIndex = -1;

bool flag = false;

// Создаем массив rows на cols с помощью new

cout << "Введите через пробел количество строк и столбцов: ";

cin >> rows >> cols;

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

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

matrix[i] = new int[cols];

srand((unsigned)time(NULL));

// Создаем массив из случайных чисел и сразу выводим его

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

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

matrix[i][j] = (int)(((double)rand() / (double)RAND\_MAX) \* (rmx - rmn) + rmn);

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

}

cout << '\n';

}

cout << '\n';

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

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

if (matrix[j][i] <= 0) { // Если это отрицательное число, меняем флаг и выходим из цикла

flag = true;

break;

}

}

if (!flag) {

columnIndex = (i != 0) ? i : cols; // Сохраняем индекс предыдущего столбца. Если это был самый первый столбец, берем последний

for (int j = 0; j < rows; j++) { // Выводим столбец, для которого все числа неотрицательные

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

matrix[j][columnIndex] = -matrix[j][columnIndex]; // Меняем предыдущий столбец

}

cout << "\n";

break; //только первый столбец, для которого выполняется условие

}

flag = false;

}

// Если индекс равен -1, то не было неотрицательного столбца и значит, мы не меняли значения для предыдущего. Иначе выводим предыдущий столбец

if (columnIndex == -1) {

cout << "Нет строк, где все числа неотрицательны";

}

else {

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

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

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

}

cout << "\n";

}

}

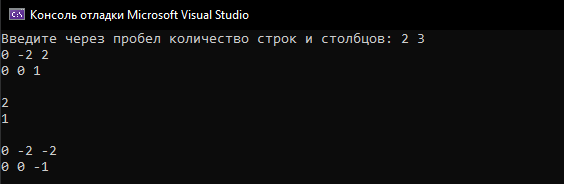
// Удаляем массив из памяти

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

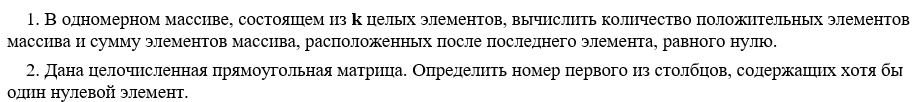
delete[] matrix[i];

delete[] matrix;

}



Вариант 8



#include <iostream>

#include <Windows.h>

#include <ctime>

using namespace std;

int main()

{

SetConsoleOutputCP(1251);

int size, lastZeroIndex, counter = 0;

int rmn = -5, rmx = 5, sum = 0;

bool foundZero = false;

cout << "Введите размер массива: ";

cin >> size;

lastZeroIndex = size; // Сразу устанавливаем индекс последнего нуля на size, так как если нет нуля, то сумма не будет находиться и будет равна 0

double\* array = (double\*)malloc(size \* sizeof(double));

srand((unsigned)time(NULL));

// Создаем массив из случайных чисел и сразу выводим его

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

\*(array + i) = (int)(((double)rand() / (double)RAND\_MAX) \* (rmx - rmn) + rmn);

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

}

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

if (\*(array + i) > 0) { // Если число больше 0, увеличиваем счетчик

counter++;

}

if (\*(array + i) == 0) { // Если 0, сохраняем его индекс

lastZeroIndex = i;

foundZero = true;

}

}

for (int i = lastZeroIndex + 1; i < size; i++) { // Находим сумму чисел после последнего 0

sum += \*(array + i);

}

cout << "\nКоличество положительных чисел: " << counter;

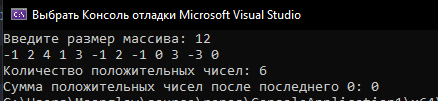
cout << "\nСумма положительных чисел после последнего 0: " << sum;

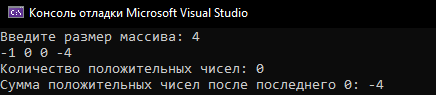
// Освобождаем выделенную память

free(array);

return 0;

}





#include <iostream>

#include <Windows.h>

using namespace std;

int main()

{

SetConsoleOutputCP(1251);

int n, m, rmn = -10, rmx = 10, columnIndex = -1;

bool foundZero = false;

// Создаем массив n на m с помощью new

cout << "Введите через пробел количество строк и количество столбцов: ";

cin >> n >> m;

int\*\* array = new int\* [n];

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

array[i] = new int[m];

srand((unsigned)time(NULL));

// Создаем массив из случайных чисел и сразу выводим его

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

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

array[i][j] = (int)(((double)rand() / (double)RAND\_MAX) \* (rmx - rmn) + rmn);

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

}

cout << '\n';

}

// Ищем столбец с нулевым элементом

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

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

if (array[j][i] == 0) { // Если это 0, сохраняем индекс этого столбца и выходим из цикла

columnIndex = i;

foundZero = true;

break;

}

}

if (foundZero) { // Если индекс найден, выводим его и выходим из цикла

cout << ++columnIndex; // ++ потому что индексы начинаются с 0, а в математике с 1

break;

}

}

// Удаляем массив из памяти

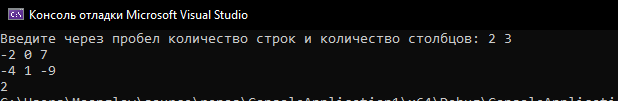
for (int i = 0; i < n; ++i)

delete[] array[i];

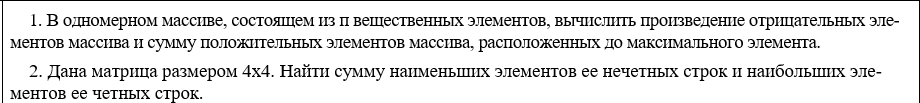
delete[] array;

return 0;

}



Вариант 3



#include <iostream>

#include <Windows.h>

#include <ctime>

using namespace std;

int main()

{

SetConsoleOutputCP(1251);

int n, indexOfMaxValue = 0;

double rmn = -20, rmx = 20, product = 1.0, maxValue, sum = 0.0;

cout << "Введите размер массива: ";

cin >> n;

double\* ptr = (double\*)malloc(n \* sizeof(double));

srand((unsigned)time(NULL));

// Создаем массив из случайных чисел и сразу выводим его

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

\*(ptr + i) = ((double)rand() / (double)RAND\_MAX) \* (rmx - rmn) + rmn;

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

}

maxValue = \*(ptr + indexOfMaxValue); // За максимальное число берем первое число массива

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

if (\*(ptr + i) > maxValue) { // Если число больше максимального, сохраненного ранее, сохраняем его как максимальное и его индекс

maxValue = \*(ptr + i);

indexOfMaxValue = i;

}

// Находим произведение отрицательных чисел

if (\*(ptr + i) < 0) {

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

}

}

// От начала до индекса максимального значения

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

if (\*(ptr + i) > 0) { // Находим сумму положительных чисел

sum += \*(ptr + i);

}

}

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

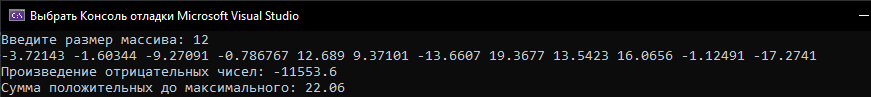
cout << "\nСумма положительных до максимального: " << sum;

// Освобождаем массив из памяти

free(ptr);

return 0;

}



#include <iostream>

#include <Windows.h>

using namespace std;

int main()

{

SetConsoleOutputCP(1251);

int n = 4, rmn = 0, rmx = 20, minValue, maxValue, sum = 0;

// Создаем массив 4 на 4 с помощью new

int\*\* ptr = new int\* [4];

for (int i = 0; i < 4; ++i)

ptr[i] = new int[4];

srand((unsigned)time(NULL));

// Создаем массив из случайных чисел и сразу выводим его

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

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

ptr[i][j] = (int)(((double)rand() / (double)RAND\_MAX) \* (rmx - rmn) + rmn);

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

}

cout << '\n';

}

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

minValue = ptr[i][0], maxValue = ptr[i][0]; // Присваиваем начальные значения для каждой строки как первое

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

if ((i + 1) % 2 != 0) { // Если это нечетная строка, то меняем значение максимального значения для этой строки на 0 и находим минимальное

maxValue = 0;

minValue = (ptr[i][j] < minValue) ? ptr[i][j] : minValue;

}

else { // Если строка четная, то меняем минимальное значение на 0 и находим максимальное

minValue = 0;

maxValue = (ptr[i][j] > maxValue) ? ptr[i][j] : maxValue;

}

}

// Добавляем полученные числа к сумме. Для этого устанавливались значения на 0, чтобы они не влияли на сумму и можно было объединить нахождение для четных и нечетных строк в одном цикле

sum += minValue;

sum += maxValue;

}

// Освобождаем массив из памяти

for (int i = 0; i < 4; ++i)

delete[] ptr[i];

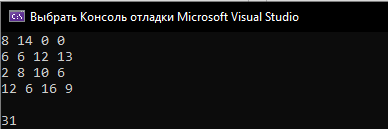
delete[] ptr;

// Выводим результат

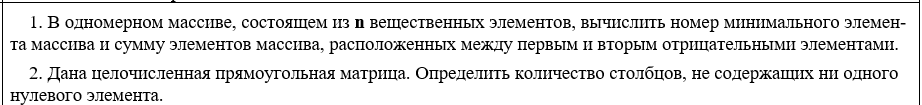
cout << '\n' << sum;

return 0;

}



Вариант 4



#include <iostream>

#include <Windows.h>

#include <ctime>

using namespace std;

int main()

{

SetConsoleOutputCP(1251);

int n, minIndex = 0;

double rmn = -20, rmx = 20, minValue, sum = 0.0;

bool flag = false;

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

double\* ptr = (double\*)malloc(n \* sizeof(double));

srand((unsigned)time(NULL));

// Создаем массив из случайных чисел и сразу выводим его

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

\*(ptr + i) = ((double)rand() / (double)RAND\_MAX) \* (rmx - rmn) + rmn;

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

}

minValue = \*(ptr + minIndex); // За минимальное число берем первое число массива

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

if (\*(ptr + i) < minValue) { // Если число меньше минимального, сохраняем его как минимальное и его индекс

minValue = \*(ptr + i);

minIndex = i;

}

if (!flag && \*(ptr + i) < 0) { // Если число отрицательное и wasFound установлен в false

flag = true; // Меняем на true, чтобы следующие отрицательные не учитывались

for (int k = i + 1; k < n; k++) { // Перебираем все числа от позиции этого отрицательного до конца массива

if (\*(ptr + k) < 0) { // Если встретилось отрицательное, выходим из цикла

break;

}

else {

sum += \*(ptr + k); // Иначе увеличиваем сумму

}

}

}

}

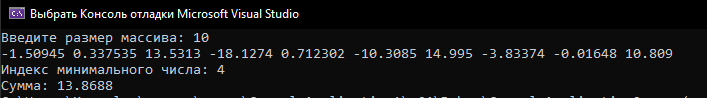
cout << "\nИндекс минимального числа: " << ++minIndex; // Поскольку индексы начинаются с 0, увеличиваем на 1

cout << "\nСумма: " << sum;

free(ptr); // Освобождаем выделенную память

return 0;

}



#include <iostream>

#include <Windows.h>

using namespace std;

int main()

{

SetConsoleOutputCP(1251);

int n, m, rmn = -3, rmx = 3, index = 0;

bool flag = false;

// Создаем массив n на m с помощью new

cout << "Введите через пробел количество строк и количество столбцов: "; cin >> n >> m;

int\*\* ptr = new int\* [n];

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

ptr[i] = new int[m];

srand((unsigned)time(NULL));

// Создаем массив из случайных чисел и сразу выводим его

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

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

ptr[i][j] = (int)(((double)rand() / (double)RAND\_MAX) \* (rmx - rmn) + rmn);

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

}

cout << '\n';

}

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

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

if (ptr[j][i] < 0) { // Если это отрицательное, меняем флаг и выходим из цикла

flag = true;

break;

}

}

if (!flag) {

index++; // Увеличиваем счетчик, если не было встречено отрицательных

}

flag = false;

}

cout << "Неотрицательных столбцов: " << index;

// Освобождаем массив из памяти

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

delete[] ptr[i];

delete[] ptr;

return 0;

}

