**Лабараторная работа № 8**

*«Написание программ с использованием динамических массивов и указателей»*

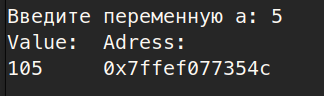
**Вариант 8**

***Цель работы:***научиться использовать операции динамического выделения и освобождения памяти на примере работы с одномерными и двумерными массивами в языке программирования С++.

**Задание 1.** Создается переменная а, вводится ее значение. Через указатель поменять значение переменной (на любое другое). Вывести значение переменной и адрес переменной (все через указатели).

**Решение:**

void task1() {

int a;

std::cout << "Введите переменную а: ";

std::cin >> a;

int\* aPtr = &a;

\*aPtr += 100;

std::cout << "Value:" << '\t' << "Adress:\n";

std::cout << a << '\t' << &a << '\n';

}

**Задание 2.** Ввести с клавиатуры число N. Создать динамический массив длины N. В массив записать одну единицу в случайном месте, остальной массив заполнить нулями. Пользователю нужно угадать, на какой позиции расположена эта единица. Если игрок вводит позицию меньше или больше, чем позиция единицы, программа пишет «Единица левее» или «Единица правее». В конце программы сообщить, с какой попытки пользователь угадал.

**Решение:**

void task2() {

size\_t N, placeOfOne, attempts = 0;

std::cout << "Введите размер массива: ";

std::cin >> N;

int\* arr = new int[N]{0};

placeOfOne = Random::getRand(N);

arr[placeOfOne] = 1;

int input = -1;

std::cout << "Введите возможное место единицы в массиве\n";

do {

std::cin >> input;

if (input > placeOfOne)

std::cout << "Единица левее\n";

else if (input < placeOfOne)

std::cout << "Единица правее\n";

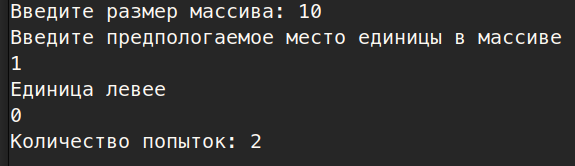
attempts++;

} while (input != placeOfOne);

std::cout << "Количество попыток: " << attempts << '\n';

delete[] arr;

}

**Задание 3.** Дана квадратная матрица *А*порядка *n*. Найдите сумму положительных элементов матрицы, стоящих над побочной диагональю (элементы диагонали не включительно).

**Решение:**

void task3() {

size\_t n, sumOfPositive = 0;

std::cout << "Введите размерность матрицы: ";

std::cin >> n;

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

for (size\_t i = 0; i < n; i++) {

matrix[i] = new int[n]{0};

for (size\_t j = 0; j < n; j++) {

int& elem = matrix[i][j];

elem = Random::getRand(-75, 50);

std::cout << elem << '\t';

}

std::cout << '\n';

}

std::cout << "\nПобочная диагональ\n";

for (size\_t i = 0; i < n; i++) {

for (size\_t j = 0; j < n - i; j++) {

int& elem = matrix[i][j];

std::cout << elem << '\t';

if (elem > 0)

sumOfPositive += elem;

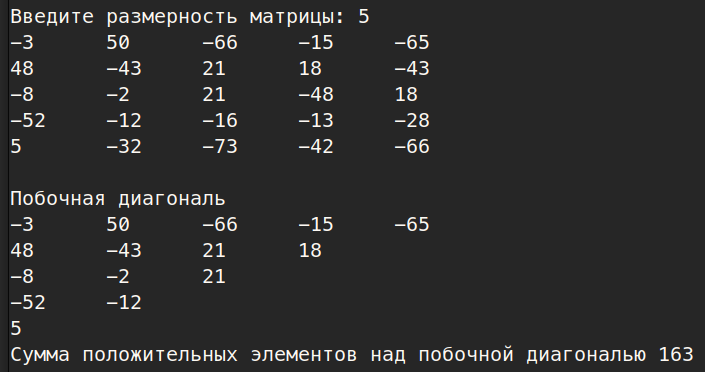
}

std::cout << '\n';

}

std::cout << "Сумма положительных элементов над побочной диагональю " << sumOfPositive << '\n';

for (size\_t i = 0; i < n; i++)

delete[] matrix[i];

delete[] matrix;

}