Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Рязанский государственный радиотехнический университет

имени В.Ф. Уткина»

Кафедра АСУ

Отчёт о практической работе №4

«Указатели в стиле C++. Динамическая память в C. Динамическая память

в C++.»

По дисциплине

«Инструментальные средства информационных систем»

Выполнил ст. гр. 135

Бардин М.С.

Проверил:

доц. Челебаев С.В.

Рязань 2023

**Цель работы:** изучение правил работы с указателями в C++.

**Задача 1**. Напишите программу, которая определяет и инициализирует массив первыми 24 нечетными числами. Выведите числа из массива на консоль по пять в строку. Для вывода используйте нотацию указателей (имя массива в качестве указателя). Подобным образом с помощью указателя выведите элементы массива в обратном порядке.

Таблица 1 – Варианты заданий для задачи № 1

|  |  |  |
| --- | --- | --- |
| № варианта | Количество элементов массива | Количество элементов, выводимых в строку |
| 1 | 24 | 6 |

Решение задачи 1.

int main()

{

const unsigned n{24};

unsigned array[n];

for (unsigned i{}; i < n; i++)

{

array[i] = i \* 2 + 1;

}

const unsigned perline{6};

for (unsigned i{}; i < n; i++)

{

std::cout << \*(array + i) << "\t";

if ((i + 1) % perline == 0)

{

std::cout << std::endl;

}

}

std::cout << "\nReverse order:\n";

for (int i{ n - 1 }; i >= 0; --i)

{

std::cout << \*(array + i) << "\t";

if (i % perline == 0)

{

std::cout << std::endl;

}

}

}

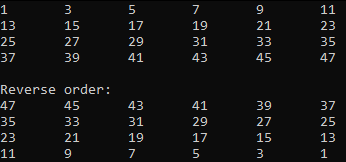


Рисунок 1 – Результат работы программы

**Задача 2.** Напишите программу, которая определяет и инициализирует массив первыми 20 нечетными числами. Выведите числа из массива на консоль по пять в строку. Для вывода определите указатель, который указывает на массив. С помощью инкремента (++) указателя выведите элементы в прямом порядке. А с потом в отдельном цикле с помощью декремента указателя выведите элементы массива в обратном порядке.

Таблица 2 – Варианты заданий для задачи № 2

|  |  |  |
| --- | --- | --- |
| № варианта | Количество элементов массива | Количество элементов, выводимых в строку |
| 1 | 35 | 7 |

Решение задачи 2.

int main()

{

const unsigned n{ 35 };

unsigned array[n];

for (unsigned i{}; i < n; i++)

{

array[i] = i \* 2 + 1;

}

const unsigned perline{ 7 };

unsigned\* ptr{ array };

for (unsigned i{}; i < n; i++)

{

std::cout << \*ptr++ << "\t";

if ((i + 1) % perline == 0)

{

std::cout << std::endl;

}

}

std::cout << "\nReverse order:\n";

for (int i{ n - 1 }; i >= 0; --i)

{

std::cout << \*--ptr << "\t";

if (i % perline == 0)

{

std::cout << std::endl;

}

}

}

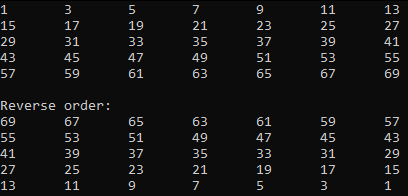


Рисунок 2 – Результат работы программы

**Задача № 3.** Решить с использованием указателей.

|  |  |
| --- | --- |
| № варианта | Задание |
| 1 | Найти сумму отрицательных элементов одномерного массива |

Решение задачи 3.

int main() {

setlocale(LC\_ALL, "Russian");

int n;

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

std::cin >> n;

int\* arr = new int[n];

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

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

std::cin >> arr[i];

}

int sum = 0;

int\* ptr = arr;

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

if (\*ptr < 0) {

sum += \*ptr;

}

ptr++;

}

std::cout << "Сумма отрицательных элементов массива: " << sum << std::endl;

delete[] arr;

return 0;

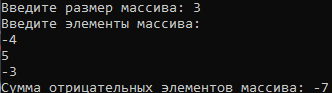
}’

Рисунок 3 – Результат работы программы

**Задача 4**. Напишите программу, которая выделяет память для одного объекта *double*, используя размер типа, и для одного объекта *long*, используя размер указателя. Поместите значения в выделенные участки памяти и выведите эти значения на консоль. После этого освободите выделенную память.

Таблица – Варианты заданий для задачи № 4

|  |  |  |
| --- | --- | --- |
| № варианта | Тип первой переменной | Тип второй переменной |
| 1 | float | unsigned long |

int main() {

setlocale(LC\_ALL, "Russian");

float\* floatPtr = new float;

unsigned long\* ulongPtr = new unsigned long;

\*floatPtr = 3.14f;

\*ulongPtr = 1234567890UL;

std::cout << "Значение float: " << \*floatPtr << std::endl;

std::cout << "Значение unsigned long: " << \*ulongPtr << std::endl;

delete floatPtr;

delete ulongPtr;

return 0;

}



Рисунок 4 – Результат работы программы

**Задача 5.** Напишите программу, которая динамически выделяет память для массива из пяти чисел типа *long*. Используя цикл *for*, присвойте значения элементами массива и выведите их на консоль. После этого освободите память.

Таблица – Варианты заданий для задачи № 5

|  |  |
| --- | --- |
| № варианта | Доп. Задание |
| 1 | Найти сумму элементов массива, делящихся без остатка на 2 |

Решение задачи 5.

int main() {

setlocale(LC\_ALL, "Russian");

const int size = 5;

long\* longArray = new long[size];

std::cout << "Введите 5 чисел типа long:" << std::endl;

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

std::cin >> longArray[i];

}

std::cout << "Значения в массиве:" << std::endl;

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

std::cout << longArray[i] << " ";

}

std::cout << std::endl;

long sum = 0;

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

if (longArray[i] % 2 == 0) {

sum += longArray[i];

}

}

std::cout << "Сумма элементов, делящихся нацело на 2: " << sum << std::endl;

delete[] longArray;

return 0;

}

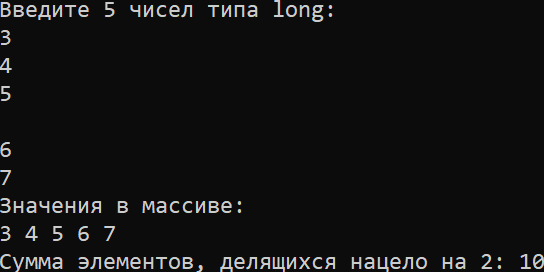


Рисунок 5 – Результат работы программы

**Задача 6.** Напишите программу, которая динамически выделяет память для массива из трех чисел типа *long*. Используя цикл *for*, присвойте значения элементами массива и выведите их на консоль. Затем с помощью функции *realloc*() увеличьте выделенную память до 5 элементов типа *long*. Установите значения для новых двух элементов *long* и выведите их на консоль. После этого освободите память.

Таблица – Варианты заданий для задачи № 6

|  |  |
| --- | --- |
| № варианта | Доп. задание |
| 1 | Добавить в конец массива один элемент, равный сумме элементов массива |

Решение задачи 6.

int main() {

srand(time(NULL));

int size = 5;

int upper = 5, lower = -5;

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

int sum = 0;

printf("Initial array: ");

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

\*(array + i) = (rand() % (upper - lower + 1)) + lower;

printf("%d\t", \*(array + i));

sum += \*(array + i);

}

array = (int\*)realloc(array, (size + 1) \* sizeof(int));

\*(array + size) = sum;

size++;

printf("\nResized array: ");

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

printf("%d\t", \*(array + i));

}

free(array);

return 0;

} 

Рисунок 6 – Результат работы программы

**Задача 7.** Напишите программу, которая динамически выделяет память для массива. При этом длина массива неизвестна и вводится во время выполнения программы пользователем, и также значения всех элементов массива также вводятся пользователем.

Таблица 7 – Варианты заданий для задачи № 7

|  |  |
| --- | --- |
| № варианта | Доп. задание |
| 1 | Отсортировать элементы массива по возрастанию |

Решение задачи 7.

int main()

{

int\* array;

int n;

printf("Размер массива = ");

scanf("%d",&n);

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

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

{

printf("array[%d] = ", i);

scanf("%d",&array[i]);

}

printf("\n");

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

{

printf("%d \t", array[i]);

}

cout << "\n";

sort(array, array + n);

cout<<"\nОтсортированный массив : \n ";

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

{

printf("%d ", array[i]);

}

cout<<endl;

free(array);

return 0;

}

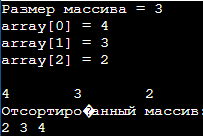


Рисунок 7 – Результат работы программы

**Задача 8**. Объявить указатель на переменную типа *int*. Ввести ее значение с клавиатуры. Объявить второй указатель на переменную того же типа. Передать второму указателю адрес 1-го объекта. Вывести оба значения типа *int*. Удалить адрес 1-го объекта. Присвоить 1-му объекту новое значение.

Таблица 8 – Варианты заданий для задачи № 8

|  |  |
| --- | --- |
| № варианта | Типы переменных |
| 1 | float |

int main() {

setlocale(LC\_ALL, "Russian");

float\* ptr1;

float\* ptr2;

float value1;

std::cout << "Введите значение типа float: ";

std::cin >> value1;

ptr1 = &value1;

ptr2 = ptr1;

std::cout << "Значение 1-го объекта: " << \*ptr1 << std::endl;

std::cout << "Значение 2-го объекта (по адресу 1-го): " << \*ptr2 << std::endl;

ptr1 = nullptr;

\*ptr2 = 42.0;

std::cout << "Новое значение 1-го объекта (через второй указатель): " << \*ptr2 << std::endl;

return 0;

}

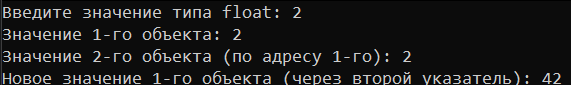


Рисунок 8 – Результат работы программы

**Задача 9**. Напишите программу, в которой пользователь вводит размер массива, и программа динамически выделяет массив такого размера для хранения значений типа *int*. Используя указатель, инициализируйте все элементы массива так, чтобы значение элемента по индексу *i* было равно *i* \* *i* (то есть квадрату числа *i*). Вычислите сумму элементов, используя синтаксис массивов (обращение к элементам по индексу в квадратных скобках), и выведите результат на консоль.

Таблица 9 – Варианты заданий для задачи № 9

|  |  |
| --- | --- |
| № варианта | Задание |
| 1 | Подсчитать количество отрицательных элементов массива |

Решение задачи 9.

int main() {

setlocale(LC\_ALL, "Russian");

unsigned n;

int\* numbers;

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

std::cin >> n;

if (n <= 0) {

std::cout << "Размер массива должен быть положительным числом." << std::endl;

return 1;

}

numbers = new int[n];

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

int countNegatives = 0; // Переменная для подсчета отрицательных элементов

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

std::cin >> numbers[i];

if (numbers[i] < 0) {

countNegatives++;

}

}

std::cout << "Количество отрицательных элементов массива: " << countNegatives << std::endl;

delete[] numbers;

return 0;

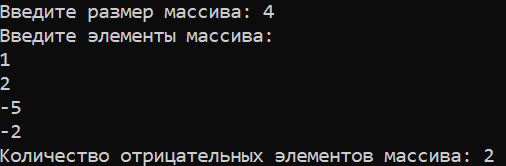
}

Рисунок 9 – Результат работы программы

**Задача 10.** Выделить память под двумерный массив. Ввести значения с клавиатуры. Вывести значения на экран.

Таблица 10 – Варианты заданий для задачи № 10

|  |  |
| --- | --- |
| № варианта | Доп. задание |
| 1 | Найти максимальный диагональный элемент и указать его координаты в главной диагонали матрицы |

Решение задачи 10.

int main() {

setlocale(LC\_ALL, "Russian");

int n, m;

std::cout << "Введите количество строк матрицы: ";

std::cin >> n;

std::cout << "Введите количество столбцов матрицы: ";

std::cin >> m;

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

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

matrix[i] = new int[m];

}

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

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

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

std::cin >> matrix[i][j];

}

}

std::cout << "Матрица:" << std::endl;

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

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

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

}

std::cout << std::endl;

}

int maxDiagonalElement = matrix[0][0];

int rowOfMaxElement = 0;

int colOfMaxElement = 0;

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

if (matrix[i][i] > maxDiagonalElement) {

maxDiagonalElement = matrix[i][i];

rowOfMaxElement = i;

colOfMaxElement = i;

}

}

std::cout << "Максимальный диагональный элемент: " << maxDiagonalElement << std::endl;

std::cout << "Координаты на главной диагонали: [" << rowOfMaxElement + 1 << "][" << colOfMaxElement + 1<< "]" << std::endl;

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

delete[] matrix[i];

}

delete[] matrix;

return 0;

}

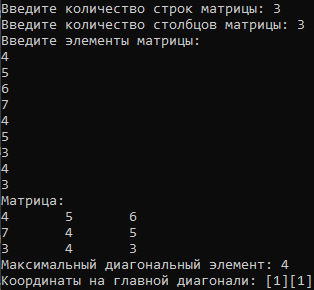


Рисунок 10 – Результат работы программы

**Задача 11.** Осуществить инициализации данными элементов двумерного массива. Вывести значения элементов.

Таблица 11 – Варианты заданий для задачи № 11

|  |  |
| --- | --- |
| № варианта | Доп. задание |
| 1 | Дана целочисленная квадратная матрица *A(N,N)*.Найдите:  а) значение наименьшего из элементов главной диагонали (минимум); б) количество элементов матрицы, меньших чем значение минимума. |

Решение задачи 11.

int main() {

setlocale(LC\_ALL, "Russian");

int N;

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

std::cin >> N;

if (N <= 0) {

std::cout << "Размерность матрицы должна быть положительным числом." << std::endl;

return 1;

}

// Создаем квадратную матрицу размером N x N

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

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

matrix[i] = new int[N];

}

// Ввод матрицы с клавиатуры

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

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

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

std::cin >> matrix[i][j];

}

}

// Начальное значение минимума устанавливаем как максимальное значение int

int minDiagonalElement = std::numeric\_limits<int>::max();

// Подсчет количества элементов меньших чем минимум

int countLessThanMin = 0;

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

// Находим минимум на главной диагонали

if (matrix[i][i] < minDiagonalElement) {

minDiagonalElement = matrix[i][i];

}

// Подсчитываем элементы меньшие чем минимум

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

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

countLessThanMin++;

}

}

}

// Вывод результатов

std::cout << "Минимальный элемент главной диагонали: " << minDiagonalElement << std::endl;

std::cout << "Количество элементов матрицы, меньших чем минимум: " << countLessThanMin << std::endl;

// Освобождение памяти

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

delete[] matrix[i];

}

delete[] matrix;

return 0;

}

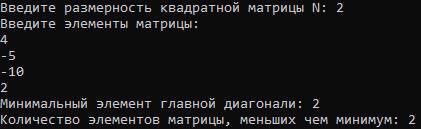


Рисунок 11 – Результат работы программы