**Белорусский государственный технологический университет**

**Факультет информационных технологий**

**Кафедра ПИ**

Лабораторная работа № 12

По дисциплине «Основы алгоритмизации и программирования»

На тему «Использование указателей»

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

2024, Минск  
  
  
  
  
  
**Задание 1**  
Изучить работу с *указателями*, выполнив программы, записанные в правой части.  
Использовать средства отладки, проанализировать память компьютера.  
Записать условия задач.  
  
**Код(слева)**

#include <iostream>  
using namespace std;

void main()

{

int a = 10, \* pa, b = 20, \* pb;

pa = &a;

cout << &a << " " << a << endl;

cout << pa << " " << \*pa << endl;

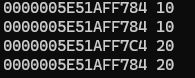
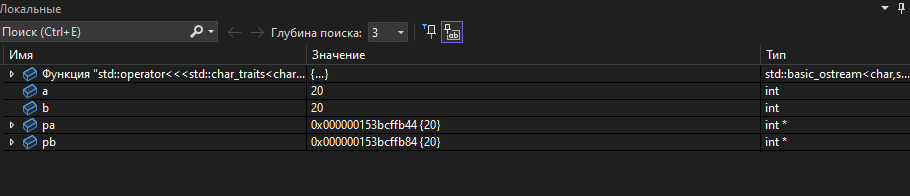
pb = &b;

cout << pb << " " << \*pb << endl;

\*pa = \*pb;

cout << &a << " " << a << endl;

}

**Условие**  
Написать алгоритм, который показывает как работают указатели, присваивает им адреса переменных, получает значения по адресам(или разыменование) и изменяет их через указатели  
  
**Результат**  
  
  
  
**Отладка**  
  
  
  
**Код(справа)**

#include <iostream>  
void main()

{

using namespace std;

float a, \* pa, s = 0; int i;

pa = &a;

for (i = 1; i <= 4; i++)

{

cout << "a=";

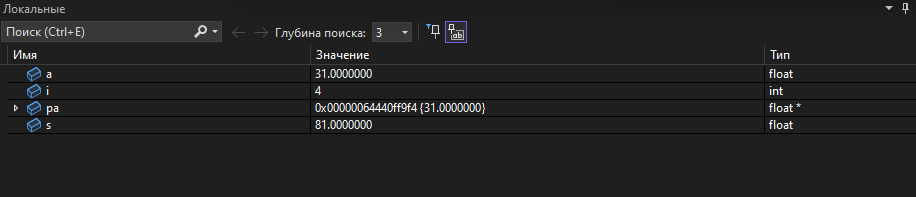
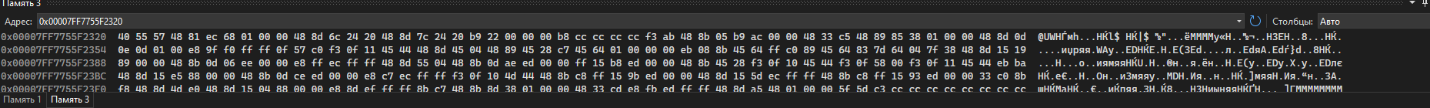
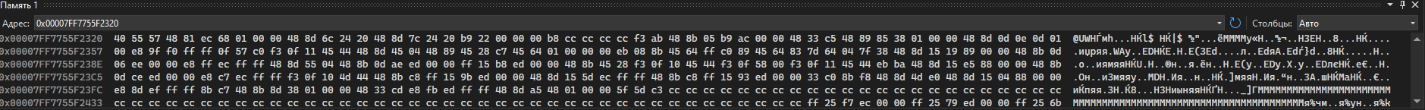
cin >> a;

s += \*pa;

}

cout << "s=" << s << endl;

}

**Условие**Написать алгоритм, результат которого будет вывод суммы четырёх чисел с использованием указателя.  
**Результат**  
  
  
**Отладка**  
  
  
  
  
**Пример содержимого памяти компьютера**  
  
  
  
  
  
  
  
  
  
  
 **Задание 2**В правой части записаны фрагменты программ с использованием *указателя на константу, константного указателя, константного указателя на константу*.  
Убрать ошибочные операторы, дописать операторы вывода и выполнить программы на компьютере.  
Написать программу генерации элементов *массива* **А** из случайных чисел, их вывода и определения максимального элемента массива двумя способами (с указателями и без них).  
  
  
**Код(Указатель на константу)**

#include <iostream>  
int main()

{

setlocale(LC\_CTYPE, "Russian");

using namespace std;

int a = 9;

const int\* pa = &a; // \*pa = 12; нельзя изменять значение по указателю на константу // необходимо было перенаправить на другую переменную по типу &a

cout << "Значение a: " << a << endl;

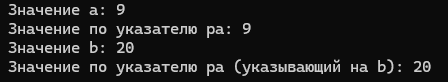
cout << "Значение по указателю pa: " << \*pa << endl;

int b = 20;

pa = &b; // pa - указатель на константу, а не константный указатель. В свою очередь он может указывать на разные переменные.

cout << "Значение b: " << b << endl;

cout << "Значение по указателю pa (указывающий на b): " << \*pa << endl;   
}

**Результат  
  
**  
  
**Код(Константный указатель)**

#include <iostream>  
int main()

{

setlocale(LC\_CTYPE, "Russian");

using namespace std;

int a = 99;

int b = 44;

int\* const pa = &a; // pa = &b; // нельзя изменять константный указатель

\*pa = 12; // изменять значение по указателю можно

cout << "Значение a после изменения: " << a << endl;

cout << "Значение по указателю pa: " << \*pa << endl;   
}

**Результат**  
  
  
  
**Код(Константный указатель на константу)**

#include <iostream>  
int main()

{

setlocale(LC\_CTYPE, "Russian");

using namespace std;

int a = 99;

const int\* const pa = &a;

// \*pa = 33; // нельзя изменять значение по указателю на константу

// pa = &b; // нельзя изменять константный указатель

int b = 44;

cout << "Значение a: " << a << endl;

cout << "Значение по указателю pa: " << \*pa << endl;   
}

**Результат  
  
**

**Код**

#include <iostream>  
#include <ctime> // для srand(time(0))

int main() {

using namespace std;

setlocale(LC\_ALL, "Russian");

int n;

cout << "Введите размер массива: ";

cin >> n;

int\* A = new int[n]; // динамическое выделение памяти для массива

srand(time(0)); // инициализация генератора случайных чисел

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

\*(A + i) = rand() % 99; // генерация случайных чисел от 0 до 98

cout << "Случайный массив: "; // выводим массив

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

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

cout << endl;

// находим максимальный элемент без указателей

int max1 = A[0];

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

if (A[i] > max1) {

max1 = A[i];

}

}

cout << "Максимальный элемент (без указателей): " << max1 << endl;

// находим максимальный элемент с использованием указателей

int max2 = \*A;

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

if (\*(A + i) > max2) {

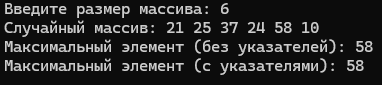
max2 = \*(A + i);

}

}

cout << "Максимальный элемент (с указателями): " << max2 << endl;

delete[] A; // освобождаем память  
}

**Результат  
  
  
  
Задание 3**   
Выполнить программу, приведенную в правой части, которая разработана с использованием указателей.  
Внести изменения с тем, чтобы программа стала содержать ошибки. Исследовать их с помощью отладки.

**Код(с ошибкой)**

#include <iostream>  
void main()

{

setlocale(LC\_CTYPE, "Russian");

using namespace std;

int i, k, sz = 5;

float A[] = { 5, -4, 17.1, 9, 1 };

cout << "Введите номер элемента (от 0 до 4) " << endl;

cin >> k;

cout << endl;

for (i = k; i <= sz; i++)

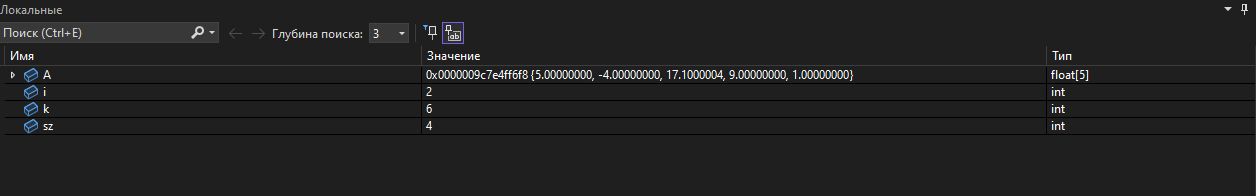
\*(A + i) = \*(A + i + 1);

sz--;

for (i = 0; i < sz; i++)

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

}

**Комментарий**  
При вводе номера элемента >= 4, программа пытается получить доступ к элементу A[sz], который находится за пределами массива A. Это выход за границы массива и может произойти неопределённое поведение программы.  
  
**Отладка**  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
 **Задание 4**Изучить отличия ссылок от указателей. Выполнить программы, записанные в правой части.  
  
**Использование ссылки:**  
**Код**

#include <iostream>  
void main()

{

int val = 1;

std::cout << "value = " << val;

int& rv = val;

rv = 5;

std::cout << " value = " << val;

}

**Результат  
  
**  
**Переход от ссылки к указателю:**  
**Код**

#include <iostream>  
void main()

{

int val = 1;

std::cout << "value = " << val;

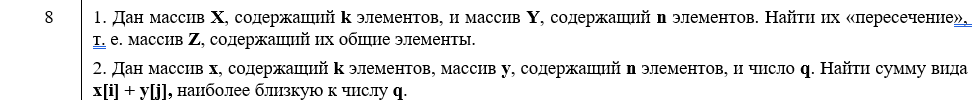
int& rv = val;

rv = 5;

int\* pval = &rv;

std::cout << " value = " << \*pval;

}

**Результат**  
  
  
  
**Комментарий**  
В первом случае код напрямую выводит значение переменной val после того, как оно было изменено через ссылку rv.  
Во втором случае код тоже изменяет значение val через ссылку rv, но затем выводит это значение непосредственно через указатель pval(он хранит адрес переменной, которая является ссылкой на val).  
Разница в способе получения значений, а не в результате.  
  
  
  
  
**Задание 5**  
В соответствии со своим вариантом написать программы для условий задач из таблицы, представленной ниже, с использованием ***указателей***для доступа к элементам массива. Проанализировать содержимое памяти компьютера.  
  
**Вариант 8**  
 **Код(1)**

#include <iostream>  
int main() {

using namespace std;

setlocale(LC\_ALL, "Russian");

int k, n;

cout << "Введите размер массива X: ";

cin >> k;

cout << "Введите размер массива Y: ";

cin >> n;

int\* X = new int[k];

int\* Y = new int[n];

cout << "Введите элементы массива X: ";

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

cin >> X[i];

}

cout << "Введите элементы массива Y: ";

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

cin >> Y[i];

}

int\* Z = new int[min(k, n)]; // создаём массива Z для пересечения

int newZ = 0;

// находим пересечения массивов

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

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

if (X[i] == Y[j]) { // если элемент X[i] есть в Y

int elementexists = 0; // проверка, нет ли уже этого элемента в Z; используем 0 = ложь

for (int z = 0; z < newZ; z++) {

if (Z[z] == X[i])

{

elementexists = 1; // 1 = истина

break;

}

}

if (elementexists == 0) { // если элемент отсутствует или еще не добавлен

Z[newZ] = X[i];

newZ++;

}

}

}

}

cout << "Пересечение массивов (массив Z): ";

if (newZ == 0) {

cout << "нет общих элементов";

}

else {

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

cout << Z[i] << " ";

}

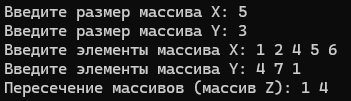
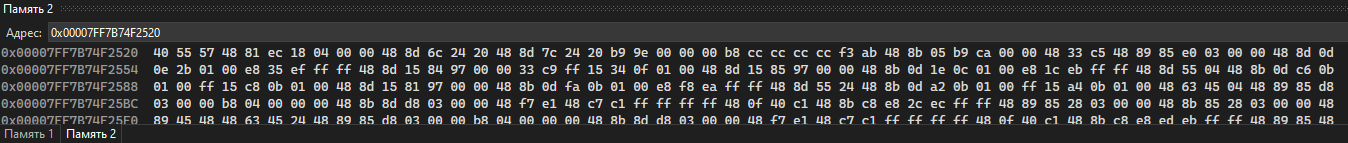
}

cout << endl;

delete[] X;

delete[] Y;

delete[] Z;  
}

**Результат(1)  
  
  
  
Анализ содержимого памяти компьютера(1)**  
  
  
  
 **Код(2)**

#include <iostream>  
#include <cmath> // для использования модуля(abs)

int main() {

setlocale(LC\_ALL, "Russian");

using namespace std;

int k, n, q;

cout << "Введите размер массива x: ";

cin >> k;

cout << "Введите размер массива y: ";

cin >> n;

cout << "Введите число q: ";

cin >> q;

int\* x = new int[k];

int\* y = new int[n];

cout << "Введите элементы массива x: ";

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

cin >> x[i];

}

cout << "Введите элементы массива y: ";

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

cin >> y[i];

}

int Sum = x[0] + y[0]; // хранение результата

int minDiff = abs(Sum - q);

for (int i = 0; i < k; i++) { // Поиск наиболее близкой суммы

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

int newSum = x[i] + y[j];

int newDiff = abs(newSum - q);

if (newDiff < minDiff) { // если найденная разница меньше минимальной

Sum = newSum;

minDiff = newDiff;

}

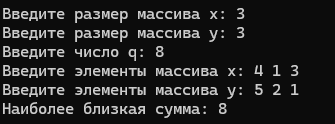
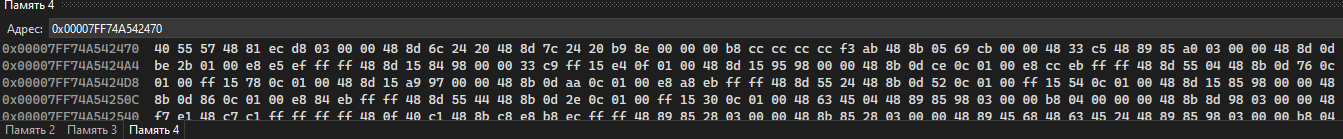
}

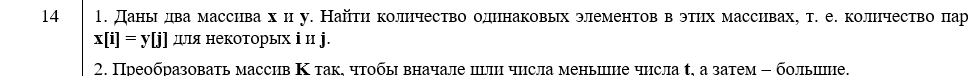
}

cout << "Наиболее близкая сумма: " << Sum << endl;

delete[] x; // освобождаем память

delete[] y; // освобождаем память  
}

**Результат(2)**  
  
  
  
**Анализ содержимого памяти компьютера(2)**  


**Дополнительные задания:  
  
Вариант 14  
  
  
  
Код(1)**

#include <iostream>  
using namespace std;

int main()

{

setlocale(LC\_ALL, "rus");

int const max = 50;

int n, i = 0, j = 0, val = 0;

cout << "Введите размерность массивов (не более 50): "; cin >> n;

int x[max], y[max];

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

{

cout << "Введите " << i + 1 << "-й элемент массива X: "; cin >> (\*(x + i));

}

cout << endl;

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

{

cout << "Введите " << j + 1 << "-й элемент массива Y: "; cin >> (\*(y + j));

}

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

{

if (\*(x + i) == (\*(y + j)))

{

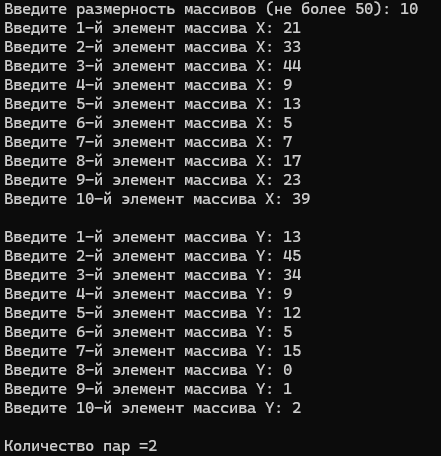
val++;

}

j++;

}

cout << "\nКоличество пар =" << val;  
}

**Результат(1)  
  
  
  
  
Код(2)**

#include <iostream>  
using namespace std;

int main()

{

setlocale(LC\_ALL, "rus");

int const max = 50;

int n, i = 0, j = 0, val = 0, t;

cout << "Введите размерность массива (не более 50): "; cin >> n;

int K[max], y[max];

cout << "Введите значение t: ";

cin >> t;

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

{

cout << "Введите " << i + 1 << "-й элемент массива: "; cin >> (\*(K + i));

}

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

cout << K[i] << " ";

cout << endl;

for (int i = 0; i < n - 1; i++)

for (int j = 0; j < n - i - 1; j++)

if (K[j] > t && K[j + 1] < t)

{

K[j] = K[j] + K[j + 1];

K[j + 1] = K[j] - K[j + 1];

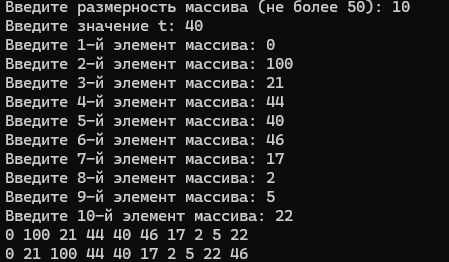
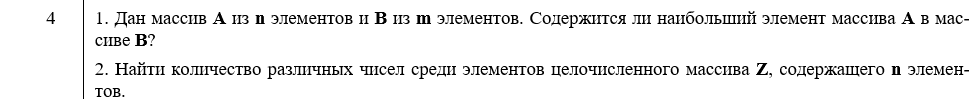
K[j] = K[j] - K[j + 1];

}

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

cout << K[i] << " ";

cout << endl;  
}

**Результат(2)  
  
  
  
  
Вариант 4  
  
  
  
Код(1)**

#include <iostream>  
int main() {

using namespace std;

setlocale(LC\_ALL, "Russian");

int n, m;

cout << "Введите размер массива A: ";

cin >> n;

cout << "Введите размер массива B: ";

cin >> m;

int\* A = new int[n];

int\* B = new int[m];

cout << "Введите элементы массива A: ";

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

cin >> A[i];

}

cout << "Введите элементы массива B: ";

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

cin >> B[i];

}

int maxA = A[0]; // находим наибольший элемент в массиве A

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

if (A[i] > maxA) {

maxA = A[i];

}

}

int found = 0; // проверяем, содержится ли maxA в массиве B. Используем 0, если не найден

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

if (B[i] == maxA) {

found = 1; // используем 1, если найден

}

}

if (found == 1) {

cout << "Наибольший элемент массива A содержится в массиве B. ";

}

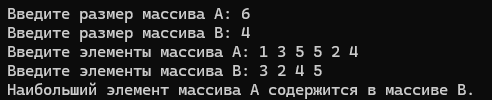
else {

cout << "Наибольший элемент массива A не содержится в массиве B.";

}

delete[] A; // освобождаем память

delete[] B; // освобождаем память  
}

**Результат(1)  
  
  
  
  
Код(2)**

#include <iostream>  
using namespace std;

int main() {

using namespace std;

setlocale(LC\_ALL, "Russian");

int n;

cout << "Введите размер массива Z: ";

cin >> n;

int\* Z = new int[n];

cout << "Введите элементы массива Z: ";

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

cin >> Z[i];

}

int Count = 0; // считаем количество элементов

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

int Duplicate = 0; // 0 — новый, 1 — дубликат

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

if (Z[i] == Z[j]) {

Duplicate = 1; // найден дубликат

break;

}

}

if (Duplicate == 0) {

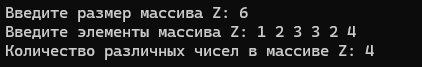
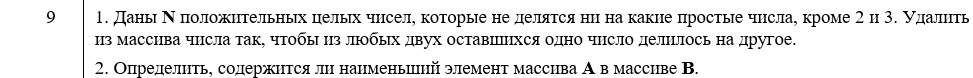
Count++;

}

}

cout << "Количество различных чисел в массиве Z: " << Count << endl;

delete[] Z;  
}

**Результат(2)  
  
  
  
  
Вариант 9  
  
  
Код(1)**

#include <iostream>  
int main() {

using namespace std;

setlocale(LC\_ALL, "Russian");

int N;

cout << "Введите количество чисел N: ";

cin >> N;

int\* A = new int[N];

cout << "Введите элементы массива A (числа, делящиеся на 2 и 3): ";

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

cin >> A[i];

}

for (int i = 0; i < N; i++) { // удаляем лишние элементы

for (int j = i + 1; j < N; j++) {

if (A[i] != 0 && A[j] != 0) {

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

{

A[j] = 0; // удаляем A[j], если оно делится на A[i]

}

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

{

A[i] = 0; // удаляем A[i], если оно делится на A[j]

}

}

}

}

cout << "Оставшиеся элементы массива: "; // выводим оставшихся элементов

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

if (A[i] != 0) {

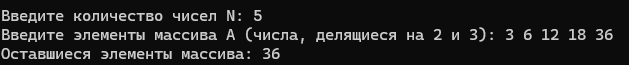
cout << A[i] << " ";

}

}

cout << endl;

delete[] A;  
}

**Результат(1)  
  
  
  
Код(2)**

#include <iostream>  
int main() {

setlocale(LC\_ALL, "Russian");

using namespace std;

int n, m;

cout << "Введите размер массива A: ";

cin >> n;

cout << "Введите размер массива B: ";

cin >> m;

int\* A = new int[n];

int\* B = new int[m];

cout << "Введите элементы массива A: ";

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

cin >> A[i];

}

cout << "Введите элементы массива B: ";

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

cin >> B[i];

}

int minA = A[0]; // находим наименьший элемент в массиве A

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

if (A[i] < minA) {

minA = A[i];

}

}

int found = 0; // проверяем, содержится ли minA в массиве B. Используем 0, если не найден

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

if (B[i] == minA) {

found = 1; // используем 1, если найден

}

}

if (found == 1) {

cout << "Наименьший элемент массива A содержится в массиве B.";

}

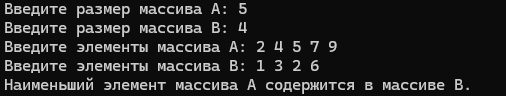
else {

cout << "Наименьший элемент массива A не содержится в массиве B.";

}

delete[] A; // освобождаем память

delete[] B; // освобождаем память  
}

**Результат(2)  
  
**