**2)**

1. Напишите программу, которая объявляет и инициализирует переменную типа int с именем number и присваивает ей значение 42. Затем выведите на экран значение этой переменной.

#include <iostream>

using namespace std;

int main() {

int number = 42; // Объявление переменной number типа int и присваивание ей значения 42

cout << "Значение number: " << number << endl; // Вывод на экран значения number

return 0;

}

2. Напишите программу, которая объявляет и инициализирует две переменные типа double с именами a и b и присваивает им значения 3.14 и 2.71 соответственно. Затем выведите на экран результат сложения, вычитания, умножения и деления этих переменных.

#include <iostream>

using namespace std;

int main() {

double a = 3.14; // Объявление переменной a типа double и присваивание ей значения 3.14

double b = 2.71; // Объявление переменной b типа double и присваивание ей значения 2.71

cout << "сложение: " << a + b << endl; // Вывод результата сложения

cout << "вычитание: " << a - b << endl; // Вывод результата вычитания

cout << "умножение: " << a \* b << endl; // Вывод результата умножения

cout << "деление: " << a / b << endl; // Вывод результата деления

return 0;

}

3. Напишите программу, которая объявляет и инициализирует переменную типа char с именем letter и присваивает ей значение ‘A’. Затем выведите на экран код этого символа в ASCII-таблице.

#include <iostream>

using namespace std;

int main() {

char letter = 'A'; // Объявление переменной letter типа char и присваивание ей значения 'A'

cout << "код символа 'A': " << (int)letter << endl; // Вывод ASCII-кода символа 'A'

return 0;

}

4. Напишите программу, которая объявляет и инициализирует переменную типа bool с именем flag и присваивает ей значение true. Затем выведите на экран значение этой переменной в виде слов true или false.

#include <iostream>

using namespace std;

int main() {

bool flag = true; // Объявление переменной flag типа bool и присваивание ей значения true

cout << (flag ? "true" : "false") << endl; // Вывод значения flag в виде текста

return 0;

}

5.Написать программу, которая проверяет, является ли введенное пользователем число четным и положительным. И выводит 1, если это так, иначе 0.

```c++

#include <iostream>

using namespace std;

int main() {

int number;

cout << "введите число: ";

cin >> number; // Ввод числа пользователем

int result = (number % 2 == 0 && number > 0); // Проверка на четность и положительность

cout << result << endl; // Вывод результата (1 - true, 0 - false)

return 0;

}

6. Напишите программу для перевода температуры из шкалы Цельсия в шкалу Фаренгейта. программа должна просить ввести пользователя температуру по Цельсию и выводить температуру по Фаренгейту. Формула перевода из Цельсия в Фаренгейт выглядит следующим образом: Ф = (Ц \* 9) / 5 + 32.

#include <iostream>

using namespace std;

int main() {

double celsii, farenheit; // Объявление переменных типа double для хранения температуры

cout << "введите температуру по цельсию: ";

cin >> celsii; // Ввод температуры по Цельсию

farenheit = (celsii \* 9) / 5 + 32; // Вычисление температуры по Фаренгейту

cout << "температура по фаренгейту: " << farenheit << endl; // Вывод результата

return 0;

}

7. Напишите программу, в которой реализован следующий алгоритм:  
1. Объявите переменную a типа int, равную 10.  
2. Используя инкремент, увеличьте значение переменной на 1.  
3. Используя операцию присваивания-вычитания, уменьшите значение на 3.  
4. Используя присваивание-остатка от деления, поделите по модулю переменную на 2.

Выведите полученный результат.

#include <iostream>

using namespace std;

int main() {

int a = 10; // Объявление переменной a типа int и присваивание ей значения 10

a++; // Инкремент (увеличение на 1)

a -= 3; // Присваивание-вычитание (уменьшение на 3)

a %= 2; // Присваивание-остатка от деления (остаток от деления на 2)

cout << "результат: " << a << endl; // Вывод полученного результата

return 0;

}

8.

Из курса математики мы знаем (надеюсь), что длина окружности равна:



где C — длина окружности, Pi — число π (математическая константа, которая выражает отношение длины окружности к её диаметру, равная примерно 3,14159), r — радиус окружности.

Поскольку число π — математическая константа, мы не хотим, чтобы кто-то в программе мог изменять её значение. Напишите программу, вычисляющую длину окружности с использование константы π.

#include <iostream>

using namespace std;

#define PI 3.14159 // Определение константы PI с помощью макроса

int main() {

double radius, dlina; // Объявление переменных для радиуса и длины

cout << "введите радиус окружности: ";

cin >> radius; // Ввод радиуса окружности

dlina = 2 \* PI \* radius; // Вычисление длины окружности

cout << "длина окружности: " << dlina << endl; // Вывод результата

return 0;

}

9. Напишите программу, которая вычисляет с помощью макроса куб числа 10 — cubeof(X)

#include <iostream>

using namespace std;

#define cubeof(X) ((X)\*(X)\*(X)) // Определение макроса cubeof для вычисления куба

int main() {

int result = cubeof(10); // Вызов макроса для вычисления куба 10

cout << "куб числа 10: " << result << endl; // Вывод результата

return 0;

}

10. Напишите программу (простейший консольный калькулятор), которая принимает от пользователя два целых числа и поочередно выводит результаты простейших математических (сложение, вычитание, умножение, деление) для этих чисел.

#include <iostream>

using namespace std;

int main() {

int a, b;

cout << "введите первое число: ";

cin >> a;

cout << "введите второе число: ";

cin >> b;

cout << "сложение: " << a + b << endl;

cout << "вычитание: " << a - b << endl;

cout << "умножение: " << a \* b << endl;

cout << "деление: " << a / b << endl;

return 0;

}

**3333333333333333333333333333)**

1.Используя оператор switch напишите программу, которая принимает от пользователя день недели и выводит этот день на консоль, и не забудьте учесть случай, если число не от 1 до 7.

#include <iostream> // Подключение библиотеки ввода/вывода

using namespace std; // Использование пространства имен std

int main() {

int day; // Объявление переменной для хранения дня недели

cout << "введите день недели (1-7): "; // Вывод запроса на ввод дня

cin >> day; // Ввод дня пользователем

switch (day) { // Оператор switch для проверки дня

case 1: // Если день равен 1

cout << "понедельник" << endl; // Вывод "понедельник"

break; // Прерывание выполнения оператора switch

case 2: // Если день равен 2

cout << "вторник" << endl;

break;

case 3: // Если день равен 3

cout << "среда" << endl;

break;

case 4: // Если день равен 4

cout << "четверг" << endl;

break;

case 5: // Если день равен 5

cout << "пятница" << endl;

break;

case 6: // Если день равен 6

cout << "суббота" << endl;

break;

case 7: // Если день равен 7

cout << "воскресенье" << endl;

break;

default: // Если день не от 1 до 7

cout << "некорректный номер дня" << endl; // Вывод сообщения об ошибке

}

return 0; // Возвращает 0, сигнализируя об успешном завершении программы

}

2.Напишите программу, которая возводит число 2 в степень 10 с помощью цикла while.

#include <iostream>

using namespace std;

int main() {

int pow = 10; // Объявление степени (10)

int result = 1; // Инициализация результата (1)

int i = 1; // Инициализация счетчика (1)

while (i <= pow) { // Цикл while, пока счетчик меньше или равен степени

result \*= 2; // Умножение результата на 2

i++; // Увеличение счетчика на 1

}

cout << "2 в степени " << pow << " равно: " << result << endl; // Вывод результата

return 0; // Возвращает 0, сигнализируя об успешном завершении программы

}

3.Напишите программу, которая запрашивает у пользователя целое число n и выводит на экран факториал этого числа. Факториал числа n — это произведение всех натуральных чисел от 1 до n. Используй цикл while и переменную для хранения результата.

#include <iostream>

using namespace std;

int main() {

int n; // Объявление переменной для хранения числа

int factorial = 1; // Инициализация факториала (1)

int i = 1; // Инициализация счетчика (1)

cout << "введите число: "; // Вывод запроса на ввод числа

cin >> n; // Ввод числа пользователем

while (i <= n) { // Цикл while, пока счетчик меньше или равен введенному числу

factorial \*= i; // Умножение факториала на текущее значение счетчика

i++; // Увеличение счетчика на 1

}

cout << "факториал числа " << n << " равен: " << factorial << endl; // Вывод факториала

return 0; // Возвращает 0, сигнализируя об успешном завершении программы

}

4.Напишите программу, которая запрашивает у пользователя целое число n и выводит на экран все четные числа от 0 до n включительно.

#include <iostream>

using namespace std;

int main() {

int n; // Объявление переменной для хранения верхнего предела

cout << "введите число: "; // Вывод запроса на ввод числа

cin >> n; // Ввод числа пользователем

cout << "четные числа от 0 до " << n << ": "; // Вывод заголовка

int i = 0; // Инициализация счетчика (0)

while (i <= n) { // Цикл while, пока счетчик меньше или равен введенному числу

if (i % 2 == 0) { // Проверка на четность

cout << i << " "; // Вывод четного числа

}

i++; // Увеличение счетчика на 1

}

cout << endl; // Переход на новую строку

return 0; // Возвращает 0, сигнализируя об успешном завершении программы

}

5.Перепишите данный код, с использованием цикла for: 

#include <iostream>

using namespace std;

int main() {

int x = 10; // Инициализация переменной x (10)

for (int i = x; i > 0; i -= 2) { // Цикл for с начальным значением x, условием i > 0 и шагом -2

cout << i << endl; // Вывод значения i

}

return 0; // Возвращает 0, сигнализируя об успешном завершении программы

}

6.Напишите программу, которая считает сумму чисел от 1 до 100, но останавливается, когда сумма станет больше 1000 и выводит полученную сумму.

#include <iostream>

using namespace std;

int main() {

int sum = 0; // Инициализация суммы (0)

int i = 1; // Инициализация счетчика (1)

while (sum <= 1000) { // Цикл while, пока сумма не превысит 1000

sum += i; // Добавление текущего значения счетчика к сумме

i++; // Увеличение счетчика на 1

}

cout << "сумма чисел от 1 до " << i - 1 << ": " << sum << endl; // Вывод суммы и последнего числа

return 0; // Возвращает 0, сигнализируя об успешном завершении программы

}

7.Напишите программу, которая запрашивает у пользователя целое число n и проверяет, является ли оно простым. Простое число — это число, которое делится только на себя и на единицу. Используйте цикл for и оператор break для прерывания цикла, если найден делитель числа.

#include <iostream>

using namespace std;

int main() {

int n; // Объявление переменной для хранения числа

cout << "введите целое число: "; // Вывод запроса на ввод числа

cin >> n; // Ввод числа пользователем

if (n <= 1) { // Проверка на 1 или меньше

cout << n << " - не простое число" << endl; // Вывод сообщения, если число не простое

return 0; // Завершение программы

}

for (int i = 2; i \* i <= n; i++) { // Цикл for, проверяет делимость до корня квадратного из n

if (n % i == 0) { // Проверка на делимость без остатка

cout << n << " - не простое число" << endl; // Вывод сообщения, если число не простое

return 0; // Завершение программы

}

}

cout << n << " - простое число" << endl; // Вывод сообщения, если число простое

return 0; // Завершение программы

}

8.Напишите программу (интерактивный консольный калькулятор), которая запрашивает у пользователя на выбор математическую операцию (операторы: + - \* / ! ^, где "^" - возведение в степень, а "!" - факториал) и целые числа, с которыми нужно выполнить эту операцию. Программа должна выводить результат выполнения выбранной математической операции.

#include <iostream>

using namespace std;

int factorial(int n) {

if (n == 0) {

return 1; // Базовый случай: факториал 0 равен 1

} else {

return n \* factorial(n - 1); // Рекурсивное вычисление факториала

}

}

int pow(int base, int exp) {

int result = 1; // Инициализация результата

for (int i = 0; i < exp; i++) { // Цикл для вычисления степени

result \*= base; // Умножение результата на base

}

return result; // Возвращает результат

}

int main() {

char operation; // Переменная для хранения операции

int num1, num2, result; // Переменные для чисел и результата

cout << "выберите операцию (+, -, \*, /, !, ^): "; // Вывод запроса на операцию

cin >> operation; // Ввод операции пользователем

cout << "введите первое число: "; // Запрос первого числа

cin >> num1; // Ввод первого числа

if (operation != '!') { // Если операция не факториал, запрашиваем второе число

cout << "введите второе число: ";

cin >> num2;

}

switch (operation) {

case '+':

result = num1 + num2;

break;

case '-':

result = num1 - num2;

break;

case '\*':

result = num1 \* num2;

break;

case '/':

if (num2 == 0) {

cout << "нельзя делить на ноль" << endl;

return 1;

}

result = num1 / num2;

break;

case '^':

result = pow(num1, num2);

break;

default:

cout << "неверная операция!" << endl;

return 1;

}

cout << "результат: " << result << endl;

}

return 0;

}

**4444444444444444444444444444)**

1.Дан массив int array[7] = {2, 3, 4, 5, 2, 8, 10};. Выведите на консоль все его элементы.

#include <iostream>

using namespace std;

int main() {

int array[7] = {2, 3, 4, 5, 2, 8, 10}; // Объявление массива целых чисел

for (int i = 0; i < 7; i++) { // Цикл for для перебора элементов массива

cout << array[i] << " "; // Вывод элемента массива с пробелом

}

cout << endl; // Переход на новую строку

return 0; // Возвращение значения 0, сигнализируя об успешном завершении программы

}

2. Напишите код, который найдёт сумму всех элементов массива int arr[ ] = {500, 4, 24, 15, 100, 105} с помощью цикла *for* и с помощью цикла *while*.

Цикл for

#include <iostream>

using namespace std;

int main() {

int arr[ ] = {500, 4, 24, 15, 100, 105}; // Объявление массива

int sum = 0; // Инициализация суммы (0)

for (int i = 0; i < 6; i++) { // Цикл for для перебора элементов

sum += arr[i]; // Добавление значения элемента к сумме

}

cout << "сумма элементов массива: " << sum << endl; // Вывод суммы

return 0; // Возвращение значения 0, сигнализируя об успешном завершении программы

}

Цикл while

#include <iostream>

using namespace std;

int main() {

int arr[ ] = {500, 4, 24, 15, 100, 105}; // Объявление массива

int sum = 0; // Инициализация суммы (0)

int i = 0; // Инициализация счетчика (0)

while (i < 6) { // Цикл while, пока счетчик меньше 6

sum += arr[i]; // Добавление значения элемента к сумме

i++; // Увеличение счетчика на 1

}

cout << "сумма элементов массива: " << sum << endl; // Вывод суммы

return 0; // Возвращение значения 0, сигнализируя об успешном завершении программы

}

3.Напишите программу, которая считает количество элементов в массиве, кратных числу, которое вводится с консоли.

#include <iostream>

using namespace std;

int main() {

int arr[] = {2, 3, 4, 5, 2, 8, 10}; // Объявление массива

int n = sizeof(arr) / sizeof(arr[0]); // Вычисление размера массива

int number; // Переменная для хранения введенного числа

cout << "введите число: "; // Вывод запроса на ввод числа

cin >> number; // Ввод числа пользователем

int count = 0; // Инициализация счетчика (0)

for (int i = 0; i < n; i++) { // Цикл for для перебора элементов массива

if (arr[i] % number == 0) { // Проверка на кратность

count++; // Увеличение счетчика, если элемент кратен введенному числу

}

}

cout << "количество элементов, кратных " << number << ": " << count << endl; // Вывод результата

return 0; // Возвращение значения 0, сигнализируя об успешном завершении программы

}

cout << "количество элементов, кратных " << number << ": " << count << endl;

return 0;

}

4.Дан массив int arr[] = {5, 4, 3, 4, 9};  
Найдите разницу между самым большим и самым маленьким элементом массива.

#include <iostream>

using namespace std;

int main() {

int arr[] = {5, 4, 3, 4, 9}; // Объявление массива

int n = sizeof(arr) / sizeof(arr[0]); // Вычисление размера массива

int min = arr[0]; // Инициализация минимального значения (первый элемент массива)

int max = arr[0]; // Инициализация максимального значения (первый элемент массива)

for (int i = 1; i < n; i++) { // Цикл for для перебора элементов массива

if (arr[i] < min) { // Проверка на минимальное значение

min = arr[i]; // Обновление минимального значения

}

if (arr[i] > max) { // Проверка на максимальное значение

max = arr[i]; // Обновление максимального значения

}

}

int diff = max - min; // Вычисление разницы между максимальным и минимальным значениями

cout << "разница между максимальным и минимальным элементом: " << diff << endl; // Вывод результата

return 0; // Возвращение значения 0, сигнализируя об успешном завершении программы

}

5.Дан массив int arr[] = {5, 4, 3, 4, 9};  
Найдите позицию максимального элемента массива.

Например, 3 — минимальный элемент, его позиция 2. Эта задача немного сложнее, чем просто нахождение максимума, тут нам понадобится еще одна переменная, которая будет хранить позицию текущего максимума.

#include <iostream>

using namespace std;

int main() {

int arr[] = {5, 4, 3, 4, 9}; // Объявление массива

int n = sizeof(arr) / sizeof(arr[0]); // Вычисление размера массива

int max = arr[0]; // Инициализация максимального значения (первый элемент массива)

int pos = 0; // Инициализация позиции максимального значения (0)

for (int i = 1; i < n; i++) { // Цикл for для перебора элементов массива

if (arr[i] > max) { // Проверка на максимальное значение

max = arr[i]; // Обновление максимального значения

pos = i; // Обновление позиции максимального значения

}

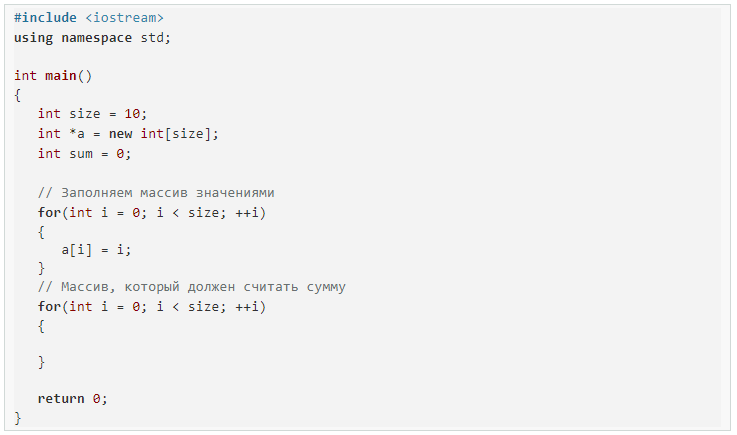
}

cout << "позиция максимального элемента: " << pos << endl; // Вывод результата

return 0; // Возвращение значения 0, сигнализируя об успешном завершении программы

}

6.Допишите программу, которая будет считать сумму элементов динамического массива.



#include <iostream>

using namespace std;

int main()

{

int size = 10; // Размер динамического массива

int \*a = new int[size]; // Выделение памяти для массива

int sum = 0; // Инициализация суммы

for(int i = 0; i < size; ++i) { // Заполнение массива значениями от 0 до 9

a[i] = i;

}

for(int i = 0; i < size; ++i) { // Цикл для суммирования элементов

sum += a[i]; // Добавление значения элемента к сумме

}

cout << "сумма элементов массива: " << sum << endl; // Вывод суммы

delete[] a; // Освобождение выделенной памяти

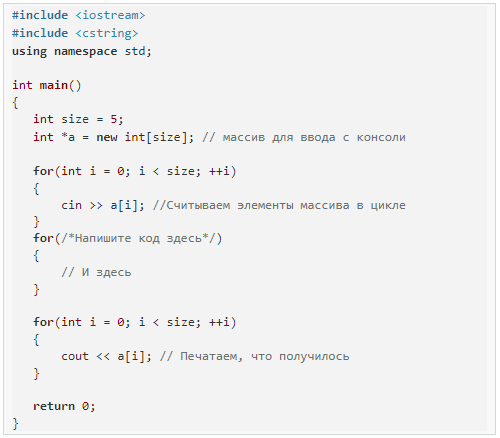
return 0; // Возвращение значения 0, сигнализируя об успешном завершении программы

}

7.

1. Допишите программу по развороту динамического массива.

Пример: если массив был [1, 2, 3, 4, 5], то должен стать [5, 4, 3, 2, 1]. Не забывайте про освобождение памяти!



#include <iostream>

#include <cstring>

using namespace std;

int main()

{

int size = 5; // Размер динамического массива

int \*a = new int[size]; // Выделение памяти для массива

for(int i = 0; i < size; ++i) { // Ввод элементов массива пользователем

cin >> a[i];

}

for(int i = 0; i < size / 2; ++i) { // Цикл для разворота массива

int te = a[i]; // Сохранение значения текущего элемента

a[i] = a[size - i - 1]; // Замена текущего элемента последним

a[size - i - 1] = te; // Замена последнего элемента текущим

}

for(int i = 0; i < size; ++i) { // Вывод развернутого массива

cout << a[i] << " ";

}

delete[] a; // Освобождение выделенной памяти

return 0; // Возвращение значения 0, сигнализируя об успешном завершении программы

}

8.Подсчитайте количество прописных (заглавных) букв в строке.

#include <iostream>

using namespace std;

int main() {

char stroka[] = {'A','a','B','b','C','c','c','D','d','E','e','F','f','G','g','g','g','g','g','g'};

int count = 0;

for (int i = 0; stroka[i] != '\0'; i++) { Цикл for для перебора символов строки

if (isupper(stroka[i])) { Проверка на прописную букву

count++;

}

}

cout << "кол-во прописных букв: " << count << endl;

return 0;

}

9.Подсчитайте количество вхождений буквы "A" в этой строке. Не забывайте, что буква "A" должна быть английской, а не русской.

#include <iostream>

using namespace std;

int main() {

char stroka[] = "Aaa1323#aa";

int count = 0;

for (int i = 0; stroka[i] != '\0'; i++) {

if (stroka[i] == 'A') {

count++;

}

}

cout << "кол-во вхождений буквы 'A': " << count << endl;

return 0;

}

10.Дана строка, содержащая только строчные буквы. Замените все буквы в строке на заглавные, выведите результат.

#include <iostream>

#include <cstring>

using namespace std;

int main() {

char stroka[] = "aaaabbbbccccddd";

int n = strlen(stroka);

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

stroka[i] = toupper(stroka[i]);

}

cout << stroka << endl;

return 0;

}

**5555555555555555555555)**

1.

#include <iostream>

using namespace std;

int main() {

int number;

cout << "введите число: ";

cin >> number;

if (number == 10) {

cout << "вы ввели 10";

}

return 0;

}

2.

#include <iostream>

using namespace std;

int main() {

int arr[5];

for (int i = 0; i < 5; i++) {

arr[i] = i \* 2;

cout << arr[i] << " ";

}

return 0;

}

3.

#include <iostream>

using namespace std;

int main() {

int count = 0;

while (count < 5) {

cout << count << endl;

count++;

}

return 0;

}

4.

#include <iostream>

using namespace std;

int main() {

float num;

cin >> num;

int numInt = (int)num;

cout << "Целое число: " << numInt << endl;

if (numInt == 5) {

cout << "numInt равно 5";

}

return 0;

}

5.

#include <iostream>

using namespace std;

int main() {

int numArray[5];

for (int i = 0; i < 5; i++) {

numArray[i] = i \* 2;

}

cout << "Array values: " << endl;

for (int i = 0; i < 5; i++) {

cout << "Element " << i << ": " << numArray[i] << endl;

}

int sum = 0;

for (int i = 0; i < 5; i++) {

sum += numArray[i];

}

cout << "Sum of array elements: " << sum << endl;

return 0;

}

6.

#include <iostream>

using namespace std;

int main() {

int x = 10;

float y = 3.14;

char myChar = 'A';

bool isTrue = true;

if (x > 5) {

cout << "x is greater than 5" << endl;

} else {

cout << "x is less or equal to 5" << endl;

}

cout << "Value of y: " << y << endl;

cout << "Character: " << myChar << endl;

cout << "Boolean: " << isTrue << endl;

return 0;

}

7.

#include <iostream>

using namespace std;

int main() {

unsigned int uVal = 10;

signed int sVal = -20;

cout << "Unsigned Value: " << uVal << endl;

cout << "Signed Value: " << sVal << endl;

int user\_input;

cout << "Enter a number: ";

cin >> user\_input;

if (user\_input < 0) {

cout << "Negative number!" << endl;

} else {

cout << "Positive number!" << endl;

}

return 0;

}

8.

#include <iostream>

using namespace std;

int main() {

int a = 10;

int b = 20;

int \*ptrA;

int \*ptrB;

ptrA = &a;

ptrB = &b;

cout << "Value of a: " << \*ptrA << endl;

cout << "Value of b: " << \*ptrB << endl;

\*ptrA = \*ptrB;

cout << "After assigning b to a: " << \*ptrA << endl;

int c = 30;

int \*ptrC;

ptrC = &c;

cout << "Value of c: " << \*ptrC << endl;

cout << "End of program." << endl;

return 0;

}

9.

#include <iostream>

using namespace std;

int main() {

int num = 5;

int \*pNum;

pNum = &num;

cout << "Number before: " << \*pNum << endl;

\*pNum = 10;

cout << "Number after: " << \*pNum << endl;

float \*pFloat;

pFloat = new float; // Выделяем память для float

\*pFloat = 5.5;

cout << "Float value: " << \*pFloat << endl;

int arr[5] = {1, 2, 3, 4, 5};

for (int i = 0; i < 5; i++) {

cout << arr[i] << " ";

}

cout << endl;

delete pFloat;

return 0;

}

10.

#include <iostream>

using namespace std;

int main() {

float fValue = 3.14;

float \*pFValue;

pFValue = &fValue;

cout << "Float value: " << \*pFValue << endl;

int arr[5] = {1, 2, 3, 4, 5};

int \*pArr;

pArr = arr;

for (int i = 0; i < 5; i++) {

cout << pArr[i] << " ";

}

cout << endl;

for (int i = 0; i < 5; i++) {

cout << arr[i] << " ";

}

cout << endl;

pFValue = nullptr;

delete pFValue;

return 0;

}

**7777777777777777777777777777)**

1.Напишите функцию с именем square, которая принимает один параметр типа int и возвращает его квадрат. В main запросите данные у пользователя и воспользуйтесь функцией для вывода результата на экран.

#include <iostream>

using namespace std;

int square(int i){ // Функция square, принимает целое число i

int sq = i \* i; // Вычисляет квадрат числа i

return sq; // Возвращает квадрат числа

}

int main(){

int i; // Объявление переменной i для ввода числа

cin >> i; // Ввод числа пользователем

int res = square(i); // Вызов функции square с аргументом i, результат сохраняется в res

cout << res; // Вывод результата на экран

}

2.Напишите функцию sqrti, которая принимает вещественное число x и возвращает квадратный корень из x. Если x отрицательное, то функция должна сгенерировать исключение типа с сообщением “Negative argument”.

Примечание: sqrt() подключаем через <cmath>

Примечание: для выхода из функции можно использовать exit(0) из библиотеки stdlib.h

Продемонстрируйте работу функци

#include <iostream>

#include <cmath> // Подключение библиотеки для математических функций

float sqrti(float x) { // Функция sqrti, принимает вещественное число x

if(x < 0) { // Проверка на отрицательное число

std::cout <<"Negative argument"; // Вывод сообщения об ошибке

return 0; // Возвращает 0, если аргумент отрицательный

}

else {

return sqrt(x); // Возвращает квадратный корень из x с помощью функции sqrt()

}

}

int main() {

float x; // Объявление переменной x для ввода числа

std::cout <<"input int: "; // Вывод запроса на ввод числа

std::cin >> x; // Ввод числа пользователем

std::cout <<"sqrt x :" << sqrti(x); // Вызов функции sqrti и вывод результата

return 0; // Возвращает 0, сигнализируя об успешном завершении программы

}

3.Напишите определение функции multiplyOrAdd, которая принимает на вход три аргумента: булевский и два вещественных и возвращает вещественный тип. Внутри тела функции должно быть произведено умножение или сложение переданных вещественных чисел в зависимости от значения булевской переменной. Выведите значение в консоль. Перепишете функцию multiplyOrAdd так, чтобы она принимала аргумент по умолчанию add = true. Вызовите новую функцию в main.

#include <iostream>

using namespace std;

double multiplyOrAdd(bool add, double a, double b) { // Функция multiplyOrAdd, принимает булеву переменную add, а также два вещественных числа a и b

if (add) { // Если add равно true

return a + b; // Возвращает сумму a и b

} else { // Если add равно false

return a \* b; // Возвращает произведение a и b

}

}

double multiplyOrAdd(double a, double b, bool add = true) { // Новая версия функции multiplyOrAdd с аргументом по умолчанию

if (add) {

return a + b;

} else {

return a \* b;

}

}

int main() {

double a = 5.0, b = 3.0; // Объявление переменных a и b

cout << "сумма: " << multiplyOrAdd(true, a, b) << endl; // Вызов функции с явно заданным add (true)

cout << "произведение: " << multiplyOrAdd(false, a, b) << endl; // Вызов функции с явно заданным add (false)

cout << "сумма по умолчанию: " << multiplyOrAdd(a, b) << endl; // Вызов функции с использованием аргумента по умолчанию (add = true)

return 0;

}

4.Напишите функцию countOdd, которая принимает на вход массив и возвращает количество четных элементов массива. В main запросите данные у пользователя и воспользуйтесь функцией для вывода результата на экран. Работайте с динамическим массивом.

#include <iostream>

using namespace std;

int countOdd(int arr[], int size) { // Функция countOdd, принимает массив arr и его размер size

int count = 0; // Инициализация счетчика (0)

for (int i = 0; i < size; i++) { // Цикл for для перебора элементов массива

if (arr[i] % 2 == 0) { // Проверка, является ли элемент четным

count++; // Увеличение счетчика, если элемент четный

}

}

return count; // Возвращает количество четных элементов

}

int main() {

int n; // Объявление переменной для хранения размера массива

cout << "введите количество элементов массива: "; // Вывод запроса на ввод размера

cin >> n; // Ввод размера массива

int\* arr = new int[n]; // Выделение памяти для динамического массива

cout << "Введите элементы массива: "; // Вывод запроса на ввод элементов

for (int i = 0; i < n; i++) { // Цикл for для ввода элементов

cin >> arr[i];

}

int oddCount = countOdd(arr, n); // Вызов функции countOdd и сохранение результата

cout << "кол-во четных элементов в массиве: " << oddCount << endl; // Вывод результата

delete[] arr; // Освобождение выделенной памяти

return 0; // Завершение программы

}

5.Напишите функцию, которая принимает два параметра по значению: целое число n и массив целых чисел arr. Функция должна возвращать сумму n-го элемента массива и n. Например, если n = 2 и arr = {1, 2, 3, 4}, то функция должна вернуть 5 (2 + 3). Продемонстрируйте работу функции.

#include <iostream>

#include <stdlib.h> // Подключение библиотеки stdlib

using namespace std;

int sum(int n, int arr[], int size) { // Функция sum, принимает индекс n, массив arr и его размер size

if (n < 0 || n >= size) { // Проверка на допустимость индекса

cerr << "ошибка: индекс n вне допустимого диапазона." << endl; // Вывод сообщения об ошибке

exit(0); // Завершение программы с кодом 0

}

return arr[n] + n; // Возвращает сумму n-го элемента массива и n

}

int main() {

int arr[] = {1, 2, 3, 4}; // Объявление массива

int n; // Объявление переменной для ввода индекса

int size = sizeof(arr) / sizeof(arr[0]); // Вычисление размера массива

cout << "ведите номер элемента в массиве n: "; // Вывод запроса на ввод индекса

cin >> n; // Ввод индекса пользователем

cout << "cумма n-го элемента массива и n: " << sum(n, arr, size) << endl; // Вывод результата

return 0; // Завершение программы

}

6.Напишите функцию, которая принимает два параметра по ссылке: целое число n и массив целых чисел arr. Функция должна увеличить n на 1 и заменить n-й элемент массива на 0. Функция не должна ничего возвращать. Например, если n = 2 и arr = {1, 2, 3, 4}, то после вызова функции n должно стать 3, а arr должен стать {1, 2, 0, 4}.

#include <iostream>

using namespace std;

void modify(int &n, int arr[], int size) { // Функция modify, принимает индекс n по ссылке, массив arr и его размер size

if (n < 0 || n >= size) { // Проверка на допустимость индекса

cerr << "ошибка: индекс n вне допустимого диапазона." << endl; // Вывод сообщения об ошибке

return; // Завершение функции

}

n++; // Увеличение индекса n на 1

arr[n - 1] = 0; // Замена n-го элемента массива на 0

}

int main() {

int arr[] = {1, 2, 3, 4}; // Объявление массива

int n = 2; // Инициализация индекса

int size = sizeof(arr) / sizeof(arr[0]); // Вычисление размера массива

modify(n, arr, size); // Вызов функции modify

cout << "n: " << n << endl; // Вывод нового значения n

cout << "массив: ";

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

cout << arr[i] << " ";

}

cout << endl;

return 0; // Завершение программы

}

7.Напишите функцию `compareStrings`, которая сравнивает две C-style строки и возвращает результат сравнения (0, если строки равны, -1, если первая строка меньше, и 1, если больше). Используйте указатели для работы со строками.

#include <iostream>

using namespace std;

int compareStrings(const char\* str1, const char\* str2) { // Функция для сравнения строк

const char\* ptr1 = str1; // Указатель на начало первой строки

while (\*ptr1 != '\0') { // Цикл по первой строке, пока не встретим символ конца строки

++ptr1; // Переход к следующему символу

}

int length1 = ptr1 - str1; // Вычисление длины первой строки

const char\* ptr2 = str2; // Указатель на начало второй строки

while (\*ptr2 != '\0') { // Цикл по второй строке

++ptr2; // Переход к следующему символу

}

int length2 = ptr2 - str2; // Вычисление длины второй строки

cout << length1 << endl; // Вывод длины первой строки

cout << length2 << endl; // Вывод длины второй строки

if (length1 == length2) { // Если длины строк равны

return 0; // Возврат 0, сигнализируя о равенстве строк

} else if (length1 < length2) { // Если длина первой строки меньше

return -1; // Возврат -1, сигнализируя о том, что первая строка меньше

} else { // Если длина первой строки больше

return 1; // Возврат 1, сигнализируя о том, что первая строка больше

}

}

int main() {

const char\* str1 = "Broken"; // Объявление первой строки

const char\* str2 = "World!"; // Объявление второй строки

int result = compareStrings(str1, str2); // Вызов функции для сравнения

cout << result << endl; // Вывод результата сравнения

return 0; // Завершение программы

}

8.Напишите функцию `findCommonElements`, которая принимает два статических массива и находит общие элементы. Используйте указатели для прохода по массивам. Верните массив общих элементов и его размер.

#include <iostream>

using namespace std;

int\* findCommonElements(int\* arr1, int size1, int\* arr2, int size2, int& commonSize) {

static int commonElements[99]; // Статический массив для хранения общих элементов

commonSize = 0; // Инициализация счетчика общих элементов (0)

for (int\* ptr1 = arr1; ptr1 < arr1 + size1; ++ptr1) { // Цикл по первому массиву

for (int\* ptr2 = arr2; ptr2 < arr2 + size2; ++ptr2) { // Цикл по второму массиву

if (\*ptr1 == \*ptr2) { // Проверка на равенство элементов

commonElements[commonSize++] = \*ptr1; // Добавление общего элемента в массив

break; // Прерывание внутреннего цикла, если найден общий элемент

}

}

}

return commonElements; // Возвращает указатель на массив общих элементов

}

int main() {

int arr1[] = {1, 2, 3, 4, 5}; // Объявление первого массива

int arr2[] = {3, 4, 5, 6, 7}; // Объявление второго массива

int commonSize = 0; // Инициализация счетчика общих элементов

int\* commonElements = findCommonElements(arr1, 5, arr2, 5, commonSize); // Вызов функции

cout << "Общие элементы: ";

for (int i = 0; i < commonSize; ++i) { // Вывод общих элементов

cout << commonElements[i] << " ";

}

cout << "\nКоличество: " << commonSize << endl; // Вывод количества общих элементов

return 0; // Завершение программы

}

9.Напишите функцию `rotateArray`, которая поворачивает массив на заданное количество позиций влево. Функция должна приниматься статическим массивом и целым числом (количество позиций). Используйте указатели и ссылку для изменения массива.

#include <iostream>

using namespace std;

void rotateArray(int\* arr, int size, int positions) { // Функция для поворота массива

positions = positions % size; // Учет цикличности поворота (поворот на 7 позиций влево равен повороту на 0)

int\* temp = new int[positions]; // Выделение памяти для временного массива

for (int i = 0; i < positions; ++i) { // Копирование первых `positions` элементов во временный массив

temp[i] = arr[i];

}

for (int i = 0; i < size - positions; ++i) { // Сдвиг оставшихся элементов влево

arr[i] = arr[i + positions];

}

for (int i = 0; i < positions; ++i) { // Копирование элементов из временного массива в конец основного массива

arr[size - positions + i] = temp[i];

}

delete[] temp; // Освобождение памяти для временного массива

}

int main() {

int arr[] = {1, 2, 3, 4, 5, 6, 7}; // Объявление массива

int size = sizeof(arr) / sizeof(arr[0]); // Вычисление размера массива

int positions = 3; // Количество позиций сдвига влево

rotateArray(arr, size, positions); // Вызов функции для поворота массива

cout << "Сдвинутый массив: ";

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

cout << arr[i] << " ";

}

cout << endl;

return 0; // Завершение программы

}

10.Напишите функцию `mergeAndCompare`, которая принимает два динамических массива и сравнивает их. Если размеры массивов разные, возвращает -1. Если размеры одинаковые, объединяет их в новый массив и возвращает этот массив и его размер через параметры.

#include <iostream>

using namespace std;

int mergeAndCompare(int\* arr1, int size1, int\* arr2, int size2, int\*& mergedArray, int& mergedSize) {

if (size1 != size2) { // Проверка на равенство размеров массивов

return -1; // Возврат -1, если размеры разные

}

mergedSize = size1 + size2; // Вычисление размера объединенного массива

mergedArray = new int[mergedSize]; // Выделение памяти для объединенного массива

for (int i = 0; i < size1; ++i) { // Копирование элементов первого массива

mergedArray[i] = arr1[i];

}

for (int i = 0; i < size2; ++i) { // Копирование элементов второго массива

mergedArray[size1 + i] = arr2[i];

}

return 0; // Возврат 0, если размеры массивов одинаковые

}

**9999999999999999999999999999999999999999999999999999999**

1. Напишите перегруженные функции `copyArray`, которые могут копировать массивы целых чисел и массивы чисел с плавающей точкой. Используйте их в main. Разделите программу на заголовочный и исполняемый файлы.

ChatGPT4 | Midjourney, [30.09.2024 16:47]

```c++

#ifndef HEADER\_H

#define HEADER\_H

#include <iostream>

namespace std {

// Перегруженные функции copyArray

template <typename T>

void copyArray(const T\* sourceArray, int sourceSize, T\* destinationArray, int destinationSize) {

if (destinationSize < sourceSize) {

cerr << "Ошибка: размер целевого массива меньше, чем размер исходного массива." << endl;

return;

}

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

destinationArray[i] = sourceArray[i];

}

}

// Перегруженные функции contains

template <typename T>

bool contains(const T\* array, int size, T value) {

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

if (array[i] == value) {

return true;

}

}

return false;

}

} // namespace std

#endif

```

ChatGPT4 | Midjourney, [30.09.2024 16:47]

\*\*main.cpp:\*\*

ChatGPT4 | Midjourney, [30.09.2024 16:47]

```c++

#include "header.h"

int main() {

// copyArray

int intArray1[] = {1, 2, 3, 4, 5};

int intArray2[5];

std::copyArray(intArray1, 5, intArray2, 5);

cout << "Копия массива целых чисел: ";

for (int i = 0; i < 5; ++i) {

cout << intArray2[i] << " ";

}

cout << endl;

float floatArray1[] = {1.1, 2.2, 3.3, 4.4, 5.5};

float floatArray2[5];

std::copyArray(floatArray1, 5, floatArray2, 5);

cout << "Копия массива вещественных чисел: ";

for (int i = 0; i < 5; ++i) {

cout << floatArray2[i] << " ";

}

cout << endl;

// contains

cout << "Содержится ли 3 в массиве intArray1: " << std::contains(intArray1, 5, 3) << endl;

cout << "Содержится ли 6.6 в массиве floatArray1: " << std::contains(floatArray1, 5, 6.6) << endl;

return 0;

}

```

1. Напишите перегруженные функции `contains`, которые проверяют, содержится ли элемент в массиве целых чисел и массиве с плавающей точкой. Используйте их в main. Разделите программу на заголовочный и исполняемый файлы.

ChatGPT4 | Midjourney, [30.09.2024 16:47]

## 2. Перегрузка функции resizeArray

header.h:

ChatGPT4 | Midjourney, [30.09.2024 16:47]

```c++

#ifndef HEADER\_H

#define HEADER\_H

#include <iostream>

namespace std {

// Перегруженные функции copyArray

template <typename T>

void copyArray(const T\* sourceArray, int sourceSize, T\* destinationArray, int destinationSize) {

if (destinationSize < sourceSize) {

cerr << "Ошибка: размер целевого массива меньше, чем размер исходного массива." << endl;

return;

}

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

destinationArray[i] = sourceArray[i];

}

}

// Перегруженные функции contains

template <typename T>

bool contains(const T\* array, int size, T value) {

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

if (array[i] == value) {

return true;

}

}

return false;

}

// Перегруженные функции resizeArray

template <typename T>

void resizeArray(T\*& array, int& size, int newSize) {

if (newSize <= 0) {

cerr << "Ошибка: новый размер массива должен быть положительным числом." << endl;

return;

}

T\* newArray = new T[newSize];

int minSize = min(size, newSize);

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

newArray[i] = array[i];

}

delete[] array;

array = newArray;

size = newSize;

}

} // namespace std

#endif

```

ChatGPT4 | Midjourney, [30.09.2024 16:47]

\*\*main.cpp:\*\*

ChatGPT4 | Midjourney, [30.09.2024 16:47]

```c++

#include "header.h"

int main() {

// resizeArray

int intArray[] = {1, 2, 3, 4, 5};

int intSize = 5;

std::resizeArray(intArray, intSize, 3);

cout << "Измененный размер массива intArray: " << intSize << endl;

cout << "Новый массив intArray: ";

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

cout << intArray[i] << " ";

}

cout << endl;

float floatArray[] = {1.1, 2.2, 3.3, 4.4, 5.5};

int floatSize = 5;

std::resizeArray(floatArray, floatSize, 7);

cout << "Измененный размер массива floatArray: " << floatSize << endl;

cout << "Новый массив floatArray: ";

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

cout << floatArray[i] << " ";

}

cout << endl;

return 0;

}

```