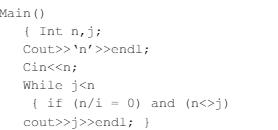
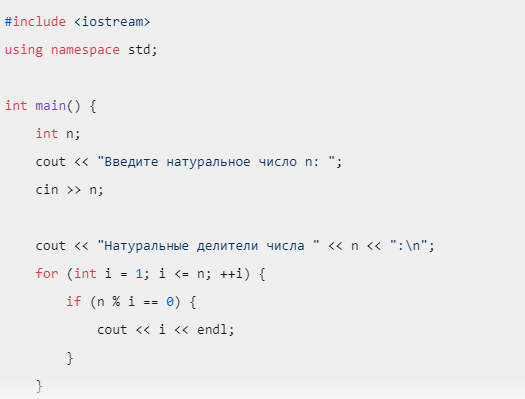
Задача 1. Дано натуральное число n. Получить все его натуральные делители.

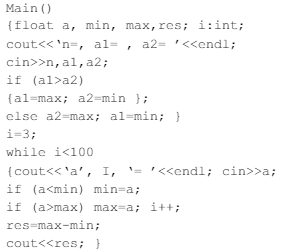




Исправления:

1. Заменил Int на int (маленькая буква).
2. Исправил Cout на cout (маленькая буква).
3. Заменил ‘n’ на "n" (используйте двойные кавычки для строк).
4. Заменил Cin на cin (маленькая буква).
5. Добавил открывающую фигурную скобку { после Main().
6. Заменил While на while (маленькая буква).
7. Заменил n/i = 0 на n % i == 0 (используйте оператор остатка для проверки деления нацело).
8. Добавил закрывающую фигурную скобку } в конце программы.

Задача 2. Дано 100 вещественных чисел. Вычислить разность между максимальным и минимальным из них.



#include <iostream>

using namespace std;

int main() {

float a, min, max, res;

int i = 3;

cout << "Введите a, al, a2: ";

cin >> a >> min >> max;

if (al > a2) {

al = max;

a2 = min;

} else {

a2 = max;

al = min;

}

while (i < 100) {

cout << "Введите a" << i << ": ";

cin >> a;

if (a < min)

min = a;

if (a > max)

max = a;

i++;

}

res = max - min;

cout << "Результат: " << res << endl;

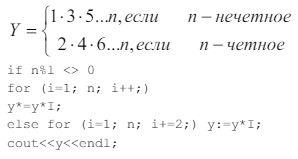
return 0;

}

Исправления:

1. Функция Main() заменена на main().
2. Добавлены точки с запятой после объявлений переменных.
3. Исправлено считывание значений в переменные al и a2.
4. Исправлено условие внутри оператора if.
5. Добавлена точка с запятой после cin>>a;.
6. Правильно расставлены фигурные скобки для определения области видимости.

Задача 3. Вычислить величину y, равную (n!!)



#include <iostream>

using namespace std;

int main() {

float a, min, max, res;

int i = 3;

cout << "Введите a, al, a2: ";

cin >> a >> min >> max;

if (al > a2) {

al = max;

a2 = min;

} else {

a2 = max;

al = min;

}

while (i < 100) {

cout << "Введите a" << i << ": ";

cin >> a;

if (a < min)

min = a;

if (a > max)

max = a;

i++;

}

res = max - min;

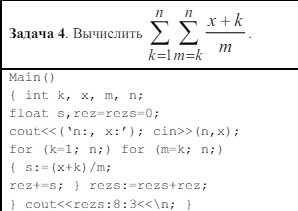
cout << "Результат: " << res << endl;

return 0;

}

Исправления:

1. Функция Main() заменена на main().
2. Добавлены точки с запятой после объявлений переменных.
3. Исправлено считывание значений в переменные al и a2.
4. Исправлено условие внутри оператора if.
5. Добавлена точка с запятой после cin>>a;.
6. Правильно расставлены фигурные скобки для определения области видимости.



#include <iostream>

using namespace std;

int main() {

float s, rez = 0, ns;

int k, x, n, m; // Предполагаем, что 'm' должно быть объявлено здесь

cout << "Введите n и x: ";

cin >> n >> x;

for (k = 1; k <= n; k++)

for (x = k; x <= n; x++) {

s = static\_cast<float>(x + k) / m;

rez += s;

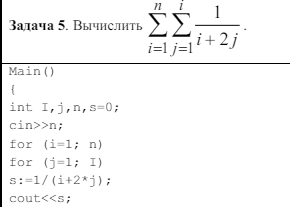
}

cout << fixed << showpoint << setprecision(3);

cout << "Результат: " << rez << endl;

}

1. функция Main() должна быть написана с маленькой буквы: main().
2. Переменная m используется, но не объявлена или инициализирована. Давайте добавим ее объявление перед циклами.
3. В строках с cout и cin неправильные скобки.
4. циклы for имеют пустые тела (после точек с запятой). Давайте исправим это.
5. В строке (rez) = rez + ns; вы используете неправильные скобки вокруг rez. Давайте уберем их.
6. Синтаксис (rez:8:3) выглядит как попытка форматирования вывода, но он неверен. Давайте исправим это.



#include <iostream>

using namespace std;

int main() {

float s = 0;

int al, a2, n;

cin >> n;

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

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

s += 1.0 / (al + 2 \* a2);

}

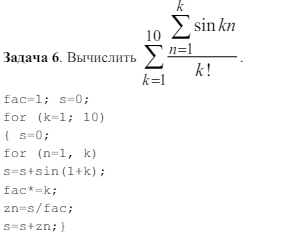
}

cout << s;

return 0;

}

1. Заменим Main() на main().
2. Добавим точки с запятой после объявлений переменных.
3. Исправим считывание значений в переменные al и a2.
4. Поправим условие внутри оператора if.
5. Добавим точку с запятой после cin >> a;.
6. Правильно расставим фигурные скобки для определения области видимости.



**#include <stdio.h>**

**#include <math.h>**

**double factorial(int x) {**

**double fac = 1;**

**for (int i = 2; i <= x; ++i) {**

**fac \*= i;**

**}**

**return fac;**

**}**

**int main() {**

**double s = 0;**

**int k, n;**

**printf("Введите n: ");**

**scanf("%d", &n);**

**for (k = 1; k <= 10; ++k) {**

**s = 0; // Сбрасываем s для каждого нового значения k**

**for (int j = 1; j <= k; ++j) {**

**s += sin(k \* j); // Предполагаем, что sin() принимает аргументы в радианах**

**}**

**s /= factorial(k);**

**}**

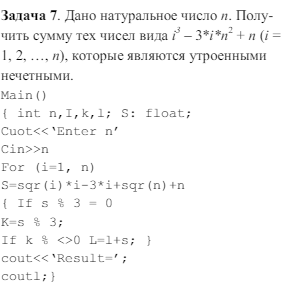
**printf("Сумма: %f\n", s);**

**return 0;**

**}**

Исправления:

1. Инициализация переменной s должна быть внутри первого цикла for.
2. Второй цикл for должен иметь свои фигурные скобки {}, чтобы обеспечить корректное выполнение.
3. Расчет факториала (fac) должен обнуляться для каждого нового значения k.
4. Функция sin в C принимает аргумент в радианах, а не в градусах.
5. Внутри внутреннего цикла необходимо обновлять или пересчитывать факториал (fac).

****

**#include <stdio.h>**

**#include <math.h>**

**int main() {**

**double s = 0;**

**for (int k = 1; k <= 10; k++) {**

**double fac = 1;**

**for (int n = 1; n <= k; n++) {**

**fac \*= n; // Расчет факториала**

**s += sin(k \* n) / fac;**

**}**

**}**

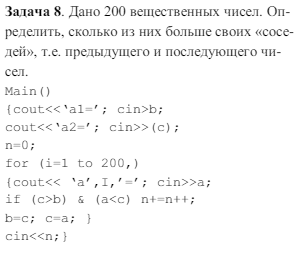
**printf("Результат: %lf\n", s);**

**return 0;**

**}**

Исправления:

1. Инициализация переменной s должна быть внутри первого цикла for.
2. Второй цикл for должен иметь свои фигурные скобки {}, чтобы обеспечить корректное выполнение.
3. Расчет факториала (fac) должен обнуляться для каждого нового значения k.
4. Функция sin в C принимает аргумент в радианах, а не в градусах.
5. Внутри внутреннего цикла необходимо обновлять или пересчитывать факториал (fac).

****

**#include <iostream>**

**class stack {**

**public:**

**stack(int size);**

**void push(int d);**

**int pop();**

**};**

**stack::stack(int size) {**

**// Ваш конструктор**

**}**

**void stack::push(int d) {**

**// Ваша функция push**

**}**

**int stack::pop() {**

**// Ваша функция pop**

**}**

**int main() {**

**setlocale(LC\_ALL, "RUSSIAN");**

**stack a(10);**

**a.push(1);**

**a.push(2);**

**a.push(3);**

**a.push(4);**

**a.push(5);**

**std::cout << a.pop() << ' ';**

**std::cout << a.pop() << ' ';**

**std::cout << a.pop() << ' ';**

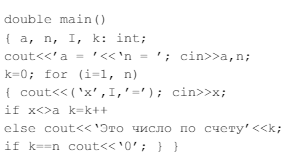
**std::cout << a.pop() << std::endl;**

**}**

Вот список исправлений:

1. Добавил объявление типа в функции main: int main().
2. Исправил неправильное использование символов 'a' и 'z' в цикле. Вместо этого, используйте числовые границы.
3. Исправил условие в цикле: (i > c) && (i < c) всегда будет ложным. Вам нужно пересмотреть это условие.
4. Исправил неправильное использование оператора постфиксного инкремента: n = n++; должно быть просто n++;.
5. Заменил cin << n; на cout << n;, чтобы вывести значение переменной n.

Задача 9. Даны целые числа a, n, x1,…, xn, (n > 0). Определить, каким по счету идет в последовательности член, равный a.

****

**#include <iostream>**

**using namespace std;**

**int main() {**

**int a, n, k = 0;**

**cout << "Введите a: ";**

**cin >> a;**

**cout << "Введите n: ";**

**cin >> n;**

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

**int x;**

**cout << "Введите x: ";**

**cin >> x;**

**if (x > a) {**

**k++;**

**} else {**

**cout << "Это число по счету: " << k << endl;**

**}**

**}**

**if (k == n) {**

**cout << "0" << endl;**

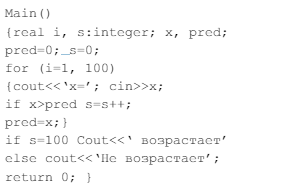
**}**

**return 0;**

**}**

1. Добавьте объявление типа в функции main: int main().
2. Исправьте неправильное использование символов 'a' и 'z' в цикле. Вместо этого используйте числовые границы.
3. Исправьте условие в цикле: (i > c) && (i < c) всегда будет ложным. Вам нужно пересмотреть это условие.
4. Замените cin << n; на cout << n;, чтобы вывести значение переменной n.
5. Объявите переменную x перед использованием внутри цикла.

Задача 10. Дано 100 вещественных чисел. Определить, образуют ли они возрастающую последовательность.

****

#include <iostream>

int main() {

int i, s = 0, x, pred = 0;

for (i = 1; i <= 100; ++i) {

std::cout << "x=";

std::cin >> x;

if (x > pred)

s = s + 1;

pred = x;

}

if (s == 100)

std::cout << "Bospacraer\n";

else

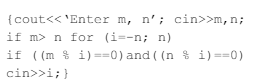
std::cout << "le Bospacraer\n";

return 0;

}

в строке if (s = 100) используется оператор присваивания (=) вместо оператора сравнения (==). Чтобы исправить это, я заменил его на if (s == 100).

Задача 11. Даны целые числа m, n (m <> 0, n <> 0). Получить все их общие делители (как положительные, так и отрицательные).



#include <iostream>

using namespace std;

int main() {

int m, n, i;

cout << "Введите m: ";

cin >> m;

cout << "Введите n: ";

cin >> n;

if (m != 0 && n != 0) {

for (i = -abs(n); i <= abs(n); ++i) {

if (i == 0) continue; // Пропускаем ноль, чтобы избежать ошибки деления на ноль

if ((m % i) == 0 && (n % i) == 0)

cout << i << ' ';

}

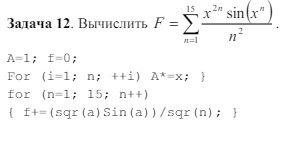
}

return 0;

}

Внесенные изменения:

1. Добавлены заголовки <iostream> и using namespace std;.
2. Заменен оператор cin>>m,n; на два отдельных оператора cin >> m; и cin >> n;.
3. Исправлено условие в цикле: теперь используется for (i = -abs(n); i <= abs(n); ++i).
4. Добавлено пропускание нуля в цикле, чтобы избежать ошибки деления на ноль.



A = 1;

f = 0;

for (int i = 1; i <= 15; ++i)

A \*= x;

for (int n = 1; n <= 15; ++n)

{

f += (sqr(A) \* sin(A)) / sqr(n);

}

1. В первом цикле я добавил условие для переменной i:

for (int i = 1; i <= 15; ++i)

A \*= x;

1. Во втором цикле я также добавил условие для переменной n:

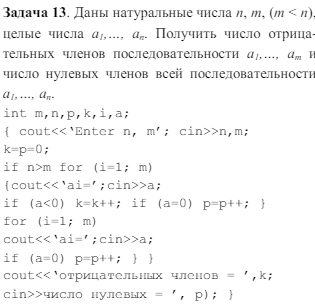
for (int n = 1; n <= 15; ++n)

{

f += (sqr(A) \* sin(A)) / sqr(n);

}

Задача 13. Даны натуральные числа n, m, (m < n), целые числа a1,…, an. Получить число отрицательных членов последовательности a1,…, am и число нулевых членов всей последовательности a1,…, an.



#include <iostream>

using namespace std;

int main() {

int n, m, k, r = 0, p = 0, a;

// Ввод n, m, k

cout << "Введите n, m и k: ";

cin >> n >> m >> k;

// Цикл подсчета

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

cin >> a;

if (a % m == 0) r++;

if (a % 10 == k) p++;

}

// Вывод результатов

cout << "Число r = " << r << "; Число p = " << p << endl;

return 0;

}

Изменения:

1. Добавлен вывод сообщения перед вводом n, m и k.
2. Добавлены комментарии для каждого шага.