|  |  |  |
| --- | --- | --- |
| Министерство науки и высшего образования РФ ФГБОУ ВПО “Ульяновский государственный университет” | Форма |  |
| Ф-Отчет к лабораторной работе |  |  |

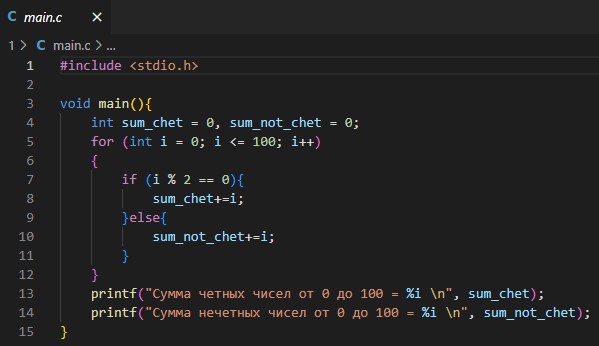
**ОТЧЕТ К ЛАБОРАТОРНОЙ РАБОТЕ**

Вариант 2. Составьте программы на языке С, С++

|  |  |
| --- | --- |
| **Факультет** математики, информационных и авиационных технологий | **Работа** №2 |
| **Группа** ИС-З-21 | **Дата выполнения** 03.02.2023 |
| **ФИО преподавателя**  Андреев Александр Сергеевич | **ФИО студента**  Гусев Владислав Евгеньевич |

**Задание №1. Найти сумму всех нечетных (s1) и сумму всех четных (s2) целых чисел в диапазоне от 1 до 100.**

Листинг программы выглядит следующим образом:



* Сперва мы объявляем 2 переменные типа int:

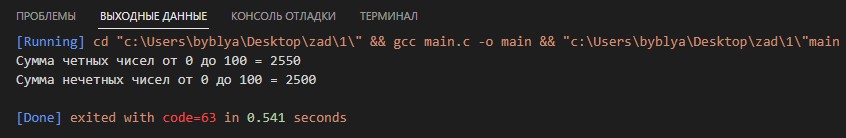
1. «Sum\_chet» для суммы четных чисел;
2. «Sum\_not\_chet» для суммы нечетных чисел.

* Далее формируется цикл от 1 до 100.
  + В цикле пишем условие «Если число i делится на 2 без остатка», то это число **четное**, и мы суммируем и присваиваем его к числу **Sum\_chet**. Иначе (если будет остаток), то число **нечетное**, и мы суммируем, и присваиваем его к числу **Sum\_not\_chet.**

Далее, результат работы программы выводим в консоль:

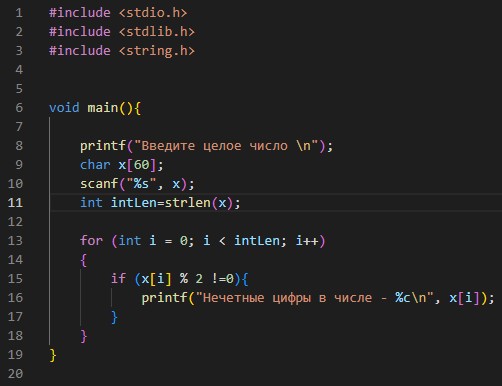
Результат суммы четных чисел от 1 до 100 = 2550

Результат суммы нечетных чисел от 1 до 100 = 2500



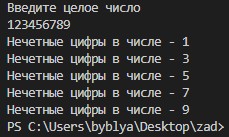
**Задание №2. Определить, имеются ли в записи целого числа n нечетные цифры.**

Листинг программы выглядит следующим образом:



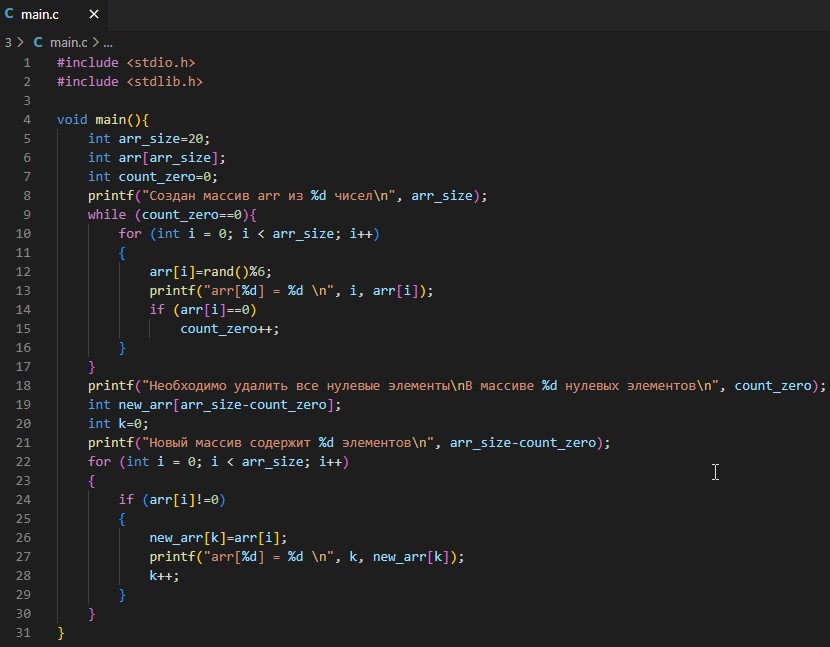
* Сначала мы объявляем переменную-массив «Х» типа char длинной в 60
* Далее, просим пользователя ввести целое число, например, **123456789**
* Заносим это число в переменную «Х».
* Объявляем переменную «intLen» типа int и присваиваем ей значение функции strlen(x) – длину строки. Например, мы введи число, состоящее из 9 цифр, следовательно, переменная «intLen» имеет значение 9.
* Далее объявляем цикл от 0 до значения «intLen» и в цикле создаем условие «Если i цифра целого числа не делится на 2 без остатка, то эта цифра нечетная и мы выводим ее на экран.

Результат работы программы:



**Задание №3. Удалить из массива все нулевые элементы.**

Листинг программы выглядит следующим образом:

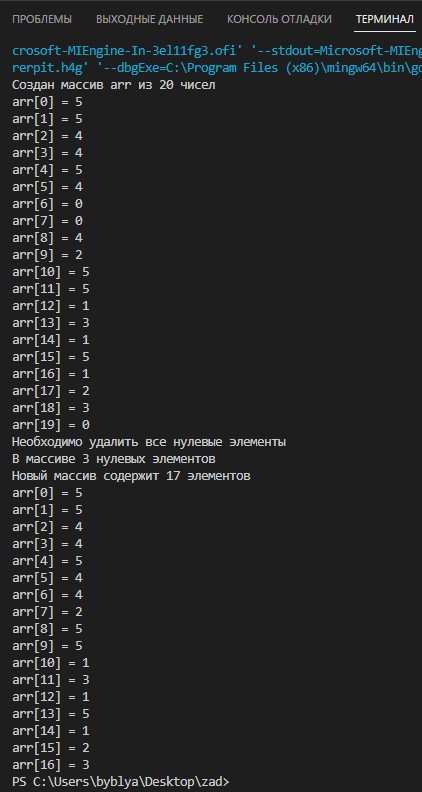


* Объявляются переменные:

1. Arr\_size – размер массива, равен 20;
2. Arr[arr\_size] – сам массив с размером 20;
3. Count\_zero – число, которое будет считать нули в массиве. Равен 0.

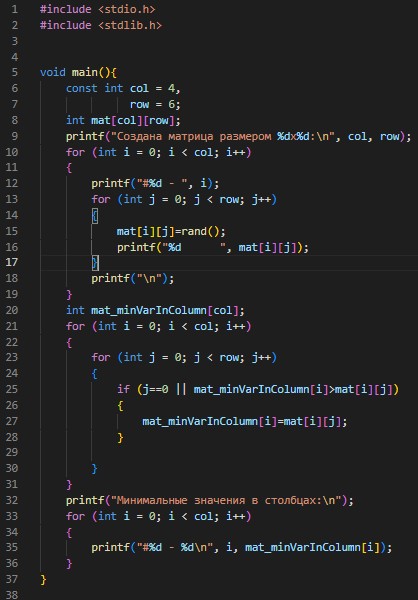
* Через цикл заполняется массив случайными числами от 0 до 5. Если в процессе заполнения, элементу массива было присвоено значение 0, то счетчик Count\_zero увеличивается.
* Далее создается новый массив размером (разница между изначальным размером массива (Arr\_size = 20) и кол-вом найденных нулей (Count\_zero)). Так же, создается вспомогательная переменная k.
* Далее объявляем цикл от 0 до Arr\_size. Значения старого массива, не считая нулей, присваиваем в новый массив. Таким образом, новый массив имеет те же значения, что и старый, только без нулевых элементов.

Результат работы программы:



**Задание №4. Найти минимальное значение в каждом столбце матрицы.**

Листинг программы выглядит следующим образом:



* Объявляем контстанты типа int

1. Col – кол-во столбцов матрицы (4)
2. Row – кол-во строк матрицы (6)

* Создаем матрицу mat на основе значений col (4) и row (6).
* Заполняем матрицу случайными числами через двойной цикл (Первый цикл отвечает за столбцы - col, второй за строки - row)
* Объявляем переменную mat\_minVarInColumn типа int длинной col (4)
* Далее, через двойной цикл перебираем нашу матрицу mat и присваиваем переменной значения в случаях если:

1. В i столбце первое число;
2. Число в i столбце j строке меньше чем в переменной минимальных чисел

* Тем самым, в переменной mat\_minVarInColumn будут лежать минимальные числа столбцов матрицы.

Результат работы программы:

