

1.

Создайте пустой вектор (`std::vector`) целых чисел. Считайте из терминала количество элементов N . Считайте N чисел из терминала, запоминая их в вектор используя метод `push_back`. После заполнения, выведите в терминал следующую информацию:

- Количество элементов в векторе
- Под сколько элементов выделена память в векторе

Далее, используя только один проход по вектору, рассчитайте и выведите в терминал следующую информацию:

- Сумму всех чисел в векторе
- Сумму четных чисел в векторе
- Количество нечетных чисел в векторе
- Минимальное и максимальное число в векторе

2.

Повторите задание 1., но используя итераторы, и автоматическое итерирование.

3.

Считайте из терминала количество слов N . Далее считывая слова и используя `std::map`, посчитайте сколько раз было введено каждое слово. Далее, используя `std::set`, получите и выведите какие слова встречались при вводе.

4.1

Создайте структуру `Data`, которая имеет 3 поля: `char sym`, `double x`, `double y`.

Создайте функцию `Data read()`, которая внутри считывает один символ и два вещественных числа, и возвращает объект `Data`, заполненный данными значениями.

Создайте функцию `void change(Data& obj)`, которая меня местами значение полей `x` и `y` в переменной `obj`, если `sym` равен ‘`a`’ или ‘`A`’.

Создайте функцию `void print(const Data& obj)`, которая выводит значения полей `sym`, `x`, `y` в переменной `obj` через пробел и в конце переносит на новую строку.

В функции `main`. Создайте список (`std::list`) хранящий элементы этих структур. Далее считайте целое число N , и введите N троек значений (`char`,

`double, double)`, используя функцию `read`. Полученные объекты типа `Data`, записывайте в начало списка, используя `push_forward`.

Потом используя `for_each`, примените функцию `change`, а затем выведите полученный результат, используя `for_each` и `print`.

4.2 - используется код 4.1

Напишите предикат `isCapitalLetter` для `Data`, который проверяет, что в структуре в поле `sym` записана заглавная буква (от ‘A’ до ‘Z’).

После последнего пункта 4.1, добавьте проверку, используя `all_of`, что в списке все структуры имеют заглавную букву. Результат выведите в терминал.

4.3 - используется код 4.2

Напишите предикат `isInRange` для `Data`, который проверяет, что в структуре оба числа `x` и `y` лежат в диапазоне `[-3.5 3.5]`.

После пункта 4.2 в коде, используя `find_if`, найдите и выведите все элементы, для которых предикат `isInRange` равен `true`.

4.4 - используется код 4.3

Напишите функцию `int makeSym(const Data& obj)`, которая возвращает сумму `obj.x` и `obj.y`, приведенную к `int`.

После пункта 4.3, создайте вектор целых чисел (`std::vector`), и запишите в него результат функции `makeSym` для каждого элемента списка, используя `transform`. Для вставки в вектор используйте `back_inserter`.

Выведите все элементы вектора в одной строчке.

4.5 - используется код 4.4

Напишите предикат `isOdd` для целых чисел, который вернет `true`, если число нечетное.

Допишите под 4.4, чтобы из полученного вектора, при помощи `remove_if` убрать все нечетные числа, из вектора.

Затем, используя `copy`, запишите оставшиеся четные числа в файл ‘out’, используя `ofstream` и `ostream_iterator`.

4.6 - используется код 4.5

Осуществите заполнение списка из пункта 4.1, используя функцию generate и написанную функцию read.