

Знакомство с итераторами и алгоритмами

Для всех исходных наборов данных вначале указывается их размер (т. е. количество элементов), а затем значения элементов.

STL1Iter1. Дан набор целых чисел. Найти количество нулей в исходном наборе. Использовать итератор `ptin_iterator` и алгоритм `count`.

STL1Iter2. Дан текстовый файл с именем *name*, содержащий строковые представления вещественных чисел. Найти количество положительных чисел в исходном файле. Использовать итератор `istream_iterator` и алгоритм `count_if`.

STL1Iter3. Дан текстовый файл с именем *name*, содержащий английские слова. Найти количество слов длины 6. Использовать итератор `istream_iterator` и алгоритм `count_if`.

STL1Iter4. Дан набор вещественных чисел. Вывести все элементы этого набора в том же порядке. Использовать итераторы `ptin_iterator`, `ptout_iterator` и алгоритм `copy`.

STL1Iter5. Дан текстовый файл с именем *name*, содержащий строковые представления целых чисел. Вывести все числа из файла в том же порядке. Использовать итераторы `istream_iterator`, `ptout_iterator` и алгоритм `copy`.

STL1Iter6. Дана строка *name* и набор символов. Записать в текстовый файл с именем *name* исходный набор символов в том же порядке, добавляя после каждого символа пробел. Использовать итераторы `ptin_iterator`, `ostream_iterator` и алгоритм `copy`.

STL1Iter7. Дан текстовый файл с именем *name1*, содержащий строковые представления целых чисел, и строка *name2*. Записать в текстовый файл с именем *name2* все ненулевые числа из исходного файла в том же порядке, располагая каждое число на новой строке. Использовать итераторы `istream_iterator`, `ostream_iterator` и алгоритм `remove_copy`.

STL1Iter8. Дан текстовый файл с именем *name1*, содержащий английские слова, и строка *name2*. Записать в текстовый файл с именем *name2* все слова из исходного файла, которые начинаются и оканчиваются одной и той же буквой, сохраняя порядок их следования и располагая каждое слово на новой строке. Использовать итераторы `istream_iterator`, `ostream_iterator` и алгоритм `remove_copy_if`.

STL1Iter9. Дан текстовый файл с именем *name*, содержащий строковые представления целых чисел. Вывести числа из исходного файла с нечетными порядковыми номерами (т. е. первое число, третье число и т. д.). Использовать итераторы `istream_iterator`, `ptout_iterator` и алгоритм `remove_copy_if`.

STL1Iter10. Дан набор вещественных чисел, содержащий не менее двух элементов. Вывести числа из исходного набора с четными порядковыми номерами (т. е. второе число, четвертое число и т. д.). Использовать итераторы `ptin_iterator`, `ptout_iterator` и алгоритм `remove_copy_if`.

STL1Iter11. Решить задачу STL1Iter7, используя вместо алгоритма `remove_copy` цикл `for` с параметром-итератором.

Указание. Опишите итератор `out` типа `ostream_iterator` и организуйте цикл `for` с параметром-итератором `in` типа `istream_iterator` и операцией инкремента `in++`. В цикле выполняйте проверку введенного числа `*in` и в случае, если оно удовлетворяет требуемому условию, выполняйте оператор `out = *in`. Выражение `*in` можно использовать многократно; оно всегда будет соответствовать последнему прочитанному элементу данных (чтение очередного элемента выполняется при создании объекта `istream_iterator` и при выполнении операции инкремента).

STL1Iter12. Решить задачу STL1Iter8, используя вместо алгоритма `remove_copy_if` цикл `for` с параметром-итератором.

Указание. Ср. с задачей STL1Iter11.

STL1Iter13. Решить задачу STL1Iter9, используя вместо алгоритма `remove_copy_if` цикл `for` с параметром-итератором.

Указание. Опишите итератор `out` типа `ptout_iterator` и организуйте цикл `for` с параметром-итератором `in` типа `istream_iterator` и операцией инкремента `in++`. В цикле выполняйте оператор `out = *in++`.

STL1Iter14. Решить задачу STL1Iter10, используя вместо алгоритма `remove_copy_if` цикл `for` с параметром-итератором.

Указание. Ср. с задачей STL1Iter13. В данном случае надо использовать цикл с параметром `in` типа `ptin_iterator`.

STL1Iter15. Дана строка *name* и набор целых чисел. Записать в текстовый файл с именем *name* все числа из исходного набора в том же порядке, заменяя каждое число 0 на число 10 и добавляя после каждого числа два пробела. Использовать итераторы `ptin_iterator`, `ostream_iterator` и алгоритм `replace_copy`.

STL1Iter16. Дан набор символов. Вывести все символы из исходного набора в том же порядке, заменяя цифровые символы на символ подчеркивания. Использовать итераторы `ptin_iterator`, `ptout_iterator` и алгоритм `replace_copy_if`.

STL1Iter17. Дана строка *name* и набор символов. Записать в текстовый файл с именем *name* удвоенные кодовые значения всех символов из исходного набора в том же порядке, добавляя после каждого числа один пробел. Использовать итераторы `ptin_iterator`, `ostream_iterator` и алгоритм `transform`.

STL1Iter18. Дана строка *name* и целое число $K (> 0)$. Записать в текстовый файл с именем *name* K символов «*». Использовать итератор `ostream_iterator` и алгоритм `fill_n`.

STL1Iter19. Даны вещественные числа A , D и целое число N . Вывести N первых членов арифметической прогрессии с первым элементом A и разностью D . Использовать итератор `ptout_iterator` и алгоритм `generate_n`.

Указание. Если компилятор поддерживает стандарт C++11, то в качестве параметра-генератора алгоритма используйте лямбда-выражение с внешней переменной D , захваченной по значению, и вспомогательной вещественной переменной, захваченной по ссылке.

STL1Iter20. Дана строка *name* и целое число $N (1 \leq N \leq 26)$. Записать в текстовый файл с именем *name* N начальных прописных букв латинского алфавита. Использовать итератор `ostream_iterator` и алгоритм `generate_n`.

Указание. Если компилятор поддерживает стандарт C++11, то в качестве параметра-генератора алгоритма используйте лямбда-выражение с внешней вспомогательной символьной переменной, захваченной по ссылке.

STL1Iter21. Даны два текстовых файла с именами *name1* и *name2*, содержащие строковые представления целых чисел, причем в каждом файле числа располагаются по возрастанию. Вывести все числа из исходных файлов в виде единой последовательности, упорядоченной по возрастанию. Использовать итераторы `istream_iterator`, `ptout_iterator` и алгоритм `merge`.

STL1Iter22. Даны два текстовых файла с именами *name1* и *name2*, содержащие английские слова, причем в каждом файле слова располагаются по возрастанию длины, а слова равной длины — в лексикографическом порядке. Вывести все слова из исходных файлов в виде единой последовательности, упорядоченной таким же образом, как и исходные файлы. Использовать итераторы `istream_iterator`, `ptout_iterator` и алгоритм `merge` с параметром — функциональным объектом.

STL1Iter23. Даны два текстовых файла с именами *name1* и *name2*, содержащие одинаковое количество строковых представлений вещественных чисел. Вывести разности $B_i - A_i$,

$B_2 - A_2, \dots, B_N - A_N$, где N — количество чисел в каждом файле, A_1, A_2, \dots, A_N — числа из файла *name1*, а B_1, B_2, \dots, B_N — числа из файла *name2*. Использовать итераторы `istream_iterator`, `ptout_iterator` и алгоритм `transform`.

STL11ter24. Дан текстовый файл с именем *name1*, содержащий английские слова, строка *name2* и набор английских слов, размер которого не превосходит количество слов, содержащихся в файле *name1*. Записать в текстовый файл с именем *name2* все слова из исходного набора, дополнив каждое слово символом «-» (дефис) и словом из файла *name1* с тем же порядковым номером (если файл *name1* содержит больше слов, чем исходный набор, то лишние слова в файле игнорируются). Каждое дополненное слово записывать в файл с новой строки. Использовать итераторы `ptin_iterator`, `istream_iterator`, `ostream_iterator` и алгоритм `transform`.