

Необходимо реализовать соответствующую функцию, используя средства библиотеки STL, и продемонстрировать её работоспособность на конкретных примерах.

Вариант 1.

Напишите функцию, которая читает текстовый файл, состоящий из строк (слова в строках разделены произвольным количеством пробельных символов), и печатает самое длинное начало каждой строки, являющееся палиндромом. Каждое слово выводить на отдельной строке стандартного потока вывода. Использовать класс string, потоки, и стандартный алгоритм reverse. Не использовать доступ к индивидуальным символам строки [], или at или доступ через итераторы. Реверсирование строки s реализовывать с помощью стандартного алгоритма reverse.

Вариант 2.

Напишите функцию, которая принимает контейнер-вектор элементов типа int, удаляет каждый элемент с четным индексом, а затем распечатывает элементы вектора в обратном порядке. Для доступа к элементам вектора использовать только итераторы.

Вариант 3.

Написать функцию, которая принимает два параметра:

- неизменяемый вектор целых чисел v1;
- вектор целых чисел v2.

Функция проходит по всем элементам векторов v1 и v2. Если i-й элемент вектора v1 оказался больше i-ого элемента вектора v2, он копируется в конец вектора v2. Добавленные в конец вектора v2 элементы не учитываются при прохождении вектора v2. Затем все элементы вектора v2 выводятся на стандартный поток вывода в прямом порядке. Для доступа к элементам массива и списка использовать только итераторы.

Вариант 4.

Напишите функцию, которая принимает три параметра:

- неизменяемый вектор целых чисел;
- список целых чисел;
- целый положительный параметр: шаг по вектору.

Функция проходит элементы вектора целых чисел с заданным шагом и копирует их в последовательные элементы списка (список проходится с шагом 1). Затем элементы списка выводятся на стандартный поток вывода в обратном порядке. Для доступа к элементам массива и списка использовать только итераторы.

Вариант 5.

Описать функцию с двумя параметрами: контейнер-список целых элементов и контейнер-вектор указателей на целые. Функция должна, последовательно проходя по элементам контейнеров от начала к концу вектора и от конца к началу списка, менять местами элементы (целые значения) контейнеров, после чего распечатать в прямом порядке целые значения сначала для списка, затем для вектора. Функция возвращает количество переставленных элементов контейнеров.

Вариант 6/.

Написать функцию с параметром, представляющим собой контейнер-список указателей на элементы длинного целого типа. Функция, просматривает контейнер от конца к началу и меняет знак значения, на которое указывает текущий указатель, если следующее за ним значение отрицательно. Например, список (-1,-2,3,4) превратится в (-1,2,-3,4). Затем функция печатает значения, на которые указывают элементы контейнера в прямом порядке. Функция возвращает число измененных элементов.

Вариант 7.

Напишите шаблонную функцию `f`, которая принимает два двунаправленных итератора (`begin` и `end`), класс-предикат и значение. Функция проходит по коллекции и все элементы, для которых выполняется предикат, заменяет на переданное значение. Значение является необязательным параметром, в случае его отсутствия используется значение по умолчанию соответствующего типа. Функция возвращает число измененных элементов. Стандартными алгоритмами пользоваться запрещено. Допустимые использования функции `f`:

```
struct Pred1 { bool operator()(int v) const { return v > 10; } };
```

```
// ...
```

```
vector<int> vec; f(vec.begin(), vec.end(), Pred1());
```

```
int val = f(vec.begin(), vec.end(), Pred1(), 5);
```

Вариант 8.

Написать функцию с тремя параметрами: непустой и неизменяемый контейнер-список типа `list<long int>`, непустой контейнер-вектор типа `vector<long int>`, целое число – шаг по второму контейнеру. Функция должна копировать отрицательные элементы списка с шагом, равным 1, в уже имеющийся контейнер-вектор, от его начала к концу с шагом, равным третьему параметру, а затем распечатывать элементы вектора в прямом порядке. Функция возвращает количество измененных элементов вектора.

Вариант 9.

Написать функцию, принимающую три параметра:

- вектор целых чисел `v1`;
- неизменяемый вектор целых чисел `v2`;
- целое число `k`.

Функция должна умножить на `k` те элементы вектора `v1`, номера которых записаны в векторе `v2`. Номера элементов могут повторяться. В этом случае умножать элемент нужно несколько раз. Возможное переполнение игнорируйте. Элементы нумеруются от 1. Если в векторе `v2` задан недопустимый номер элемента, он игнорируется. Затем элементы вектора `v1` должны быть напечатаны на стандартный поток вывода в прямом порядке. Для доступа к элементам массива и списка использовать только итераторы. Для вывода элементов вектора `v1` используйте итератор вывода `ostream_iterator`.

Вариант 10.

Написать функцию с тремя параметрами: непустой и неизменяемый контейнер-вектор типа `vector<double>`, непустой контейнер-список типа `list<double>`, целое число – шаг по первому контейнеру. Функция должна сравнивать элементы списка, выбираемыми от его начала с шагом, равным 1, с элементами вектора, выбираемыми от начала с шагом, равным третьему параметру. Если обнаруживается несовпадение очередной выбранной пары, то в список в текущем месте вставляется отсутствующий элемент. Изменённый список распечатывается в обратном порядке. Функция возвращает количество элементов, вставленных в список.

Вариант 11.

Даны описания:

```
typedef vector<double> V;
```

```
struct Signif_t {
```

```
    V::size_type Index; // индекс элемента вектора
```

```
    bool Signif; // значимость элемента (true – да, false – нет)
```

```
};
```

```
typedef list<Signif_t> S;
```

Написать функцию, которая по заданному вектору типа `V` и соответствующему ему списку типа `S` вычисляет сумму значащих элементов вектора, выдавая её в выходной поток, а также индексы и значения суммируемых элементов.

Вариант 12.

Написать функцию `g()` с параметром, представляющим собой контейнер-вектор указателей на элементы вещественного типа. Функция просматривает контейнер в

прямом порядке и обнуляет текущее указываемое значение, если следующее значение отрицательно. Например, [1, -2, 4, -6, -4, 5] -> [0, -2, 0, 0, -4, 5]. Далее функция в обратном порядке печатает значения, на которые указывают элементы контейнера. Функция возвращает число измененных значений.

Вариант 13.

Написать функцию f, которая для своего аргумента-контейнера целочисленных элементов возвращает сумму первого и последнего элементов (если таковые имеются) или 0. Использование f() должно быть корректно в следующей функции:

```
int g ( vector <int> & V, const list <int> & L ) {  
  
    return f (V) + f (L);  
  
}
```

Вариант 14.

Написать функцию, читающую текстовый файл, состоящий из слов, разделённых произвольным количеством пробельных символов, и разбивающую слова на группы так, чтобы в одну группу попали все слова, составленные из одного и того же множества символов. То есть в каждом слове группы должны быть использованы все символы этой группы (может быть, несколько раз). Например, слова «двор» и «ров» не входят в одну группу, а слова «окно» и «кон» - входят. Функция должна подсчитать количество найденных групп и выдать результат на стандартный поток вывода.

Вариант 15.

Напишите шаблонную функцию f, которая получает на вход два итератора произвольного доступа (begin и end) и переворачивает соответствующий контейнер, то есть первый элемент становится последним и наоборот. Стандартными алгоритмами пользоваться запрещено. Допустимо использование функции: vector<int> vec; f(vec.begin(), vec.end());

Вариант 16.

Даны описания:

```
typedef vector<int> V;
```

```
struct Weight_t {
```

```
    V::size_type Index;           // индекс элемента вектора
```

```
    float Weight;                // вес элемента вектора
```

```
};
```

```
typedef list <Weight_t> L;
```

Написать функцию, которая по заданному вектору типа V и соответствующему ему списку типа L вычисляет средневзвешенное значение элементов вектора (средний результат умножения элементов вектора на их веса, которые надо брать из списка), выдавая в выходной поток значения и веса элементов.

Вариант 17.

Написать функцию, принимающую два параметра:

- неизменяемый вектор целых чисел v ;
- список целых чисел lst .

Функция должна удалить из списка lst элементы с номерами, заданными в векторе v . Элементы списка нумеруются от 1. Номера элементов списка отражают позиции элементов на момент начала работы программы. Если номер повторяется в массиве более одного раза, все вхождения, кроме первого, игнорируются. Если число в массиве не является допустимым номером элемента в списке, оно игнорируется. Для доступа к элементам массива и списка использовать только итераторы.