

## ВАРИАНТ 1

1. Преобразовать следующее выражение в ПОЛИЗ:  $a - (b+c*d) / e$
2. Написать рекурсивную функцию проверки, является ли линейный односвязный список упорядоченным по возрастанию.
3. В чем разница между шаблоном функции и шаблонной функцией.
4. Для чего в библиотеку STL введены итераторы?
5. В чем различие между контейнерами `list` и `vector`?
6. Для чего предназначены итераторы вывода?
7. Написать функцию, которая в векторе целых чисел определяет, сколько в векторе пар, стоящих рядом одинаковых элементов.
8. Какие типы алгоритмов присутствуют в STL?
9. Что должен делать следующий фрагмент программы

```
int array[3];  
.  
.  
.  
replace_if (array, array + 3, bind2nd (greater<int> (), 2), 4);
```

## ВАРИАНТ 2

1. Преобразовать следующее выражение в ПОЛИЗ:  $a / (d+b) - c*e$
2. Написать рекурсивную функцию определения количества листьев в бинарном дереве.
3. Для чего в коллекции, имеющей итератор, нужно описывать метод `end()`.
4. Что означает понятие контейнера в STL?
5. Почему в библиотеке STL не используются указатели вместо итераторов?
6. Когда требуется использовать последовательные итераторы?
7. Написать функцию, которая проверяет, состоит ли вектор целых чисел из одинаковых элементов.
8. Что такое функциональный объект?
9. Что должен делать следующий фрагмент программы

```
int numbers[6];  
.  
.  
.  
reverse (numbers, numbers + 6);
```

### ВАРИАНТ 3

1. Преобразовать следующее выражение в ПОЛИЗ:  $a/b/(c+d) - e*f$
2. Написать рекурсивную функцию определения максимального значения в бинарном дереве.
3. Для чего в итераторе нужно хранить указатель коллекции, с которой работает итератор.
4. Можно ли организовать контейнер, хранящий элементы пользовательского типа?
5. Что определяется в алгоритмах через диапазон итераторов?
6. В чем разница между двунаправленными итераторами и итераторами произвольного доступа?
7. Написать функцию, которая для вектора символов проверяет, является ли последовательность символов, содержащихся в векторе палиндромом (одинаково читается с начала и с конца).
8. По какому принципу определяется, какие методы должен иметь контейнер в STL?
9. Что должен делать следующий фрагмент программы

```
vector<int> v (10);  
ostream_iterator<int> iter (cout, " ");  
. . .  
copy (v.begin (), v.end (), iter);
```

### ВАРИАНТ 4

1. Преобразовать следующее выражение в ПОЛИЗ:  $a - (b+c) / (d-e) * f$
2. Написать рекурсивную функцию сравнения двух линейных односвязных списков на равенство.
3. Когда создается шаблонная функция.
4. Могут ли контейнеры содержать данные разных типов?
5. Итераторы какого типа используются при работе с векторами?
6. Для каких целей используются итераторы ввода?
7. Описать функцию, которая для вектора символов проверяет состоит ли вектор из одинаковых символов.
8. Для чего в классах библиотеки STL определены методы `begin()` и `end()`?
9. Что должен делать следующий фрагмент программы

```
int numbers[6];  
. . .  
remove (numbers, numbers + 6, 1);
```

## ВАРИАНТ 5

1. Преобразовать следующее выражение в ПОЛИЗ:  $a * (b / (c + d) - f) + e$
2. Написать рекурсивную функцию определения количества положительных элементов в бинарном дереве.
3. Какие операции следует перегружать для итераторов и почему.
4. Можно ли в алгоритмах библиотеки STL работать с контейнерами, созданными пользователем?
5. Каким требованиям должны удовлетворять все итераторы библиотеки STL?
6. Какие из методов не реализованы в шаблоне `vector` и почему: `push_back()`, `front()`, `pop_back()`, `size()`, `operator[]`, `empty()`, `push_front()`, `pop_front()`?
7. Написать функцию, которая для вектора целых чисел проверяет, является ли последовательность чисел в векторе симметричной относительно середины.
8. В чем различие между контейнерами `list` и `vector`?
9. Что должен делать следующий фрагмент программы

```
vector<int>v;  
.  
.  
.  
sort (v.begin (), v.end (), greater<int> ());
```

## ВАРИАНТ 6

1. Преобразовать следующее выражение в ПОЛИЗ:  $a - (b + c) * d / (e - f)$
2. Написать рекурсивную функцию построения копии линейного односвязного списка.
3. Какие проблемы возникают при использовании рекурсивно определенного класса «Бинарное дерево»?
4. Как взаимосвязаны контейнеры и итераторы?
5. Для чего при работе с итераторами нужен метод `end()`?
6. Какие требования к итераторам определены для алгоритмов в STL?
7. Написать функцию, которая проверяет, является ли вектор целых чисел упорядоченным по возрастанию.
8. Что такое функциональный объект?
9. Что должен делать следующий фрагмент программы

```
int array[3];  
.  
.  
.  
replace_if (array, array + 3, bind1st (greater<int> (), 2), 4);
```

## ВАРИАНТ 7

1. Преобразовать следующее выражение в ПОЛИЗ:  $(a+b*c) / (d-e*f)$
2. Написать рекурсивную функцию проверки, является ли линейный список положительным (не содержит отрицательных элементов).
3. Как связаны между собой шаблоны функций и перегрузка имени функций.
4. Если алгоритм проходит по элементам коллекции один раз, не изменяя их, какой тип итератора он должен использовать?
5. Какие адаптеры итераторов имеются в STL, для чего они нужны?
6. Какие операции должны быть определены для итератора вывода?
7. Написать функцию, которая для вектора символов определяет, сколько в векторе пар стоящих рядом одинаковых символов.
8. В каких случаях следует использовать контейнер `vector`, а в каких – `list`?
9. Что делает следующий фрагмент программы

```
int array[3];  
.  
.  
.  
int* p = remove_if (array, array + 3, bind1st(less<int> (), 2));
```

## ВАРИАНТ 8

1. Преобразовать следующее выражение в ПОЛИЗ:  $a/b - (c+d)*f+e$
2. Написать рекурсивную функцию вычисления суммы положительных элементов в бинарном дереве.
3. Почему бывает необходимо создавать явно экземпляр шаблонной функции и как это сделать.
4. Как определяется диапазон контейнера при вызове обобщенных алгоритмов?
5. Какие способы реализации функциональных объектов есть в C++?
6. Когда необходимы последовательные итераторы?
7. Написать функцию, которая для вектора целых чисел определяет, сколько раз в векторе меняется знак.
8. Почему в алгоритмах библиотеки STL можно использовать массивы?
9. Что делает следующий фрагмент программы

```
vector<int>v;  
.  
.  
.  
sort (v.begin (), v.end (), greater<int> ());
```