



МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ
УНИВЕРСИТЕТ имени М. В. ЛОМОНОСОВА
Факультет вычислительной математики
и кибернетики



И. А. Волкова, А. А. Вылиток, Л. Е. Карпов

ТЕСТЫ ПО КУРСУ «СИСТЕМЫ ПРОГРАММИРОВАНИЯ»

Объектно-ориентированное программирование на Си++
и методы трансляции на основе формальных грамматик
и автоматов

Учебное пособие для студентов II курса

```
struct B {  
    virtual operator int () { return 7; }  
};  
struct D : B {  
    operator int () { return 12; }  
};  
B & f(B & b) {  
    if ( typeid ( b ) == typeid ( D ) )  
        cout << 1;  
    else  
        cout << 2;  
    return b;  
}  
int main() {  
    B b;  
    D d;  
    cout << f( d ) - f( b ) << endl;  
}
```

МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
имени М. В. ЛОМОНОСОВА



Факультет
вычислительной математики
и кибернетики



И. А. Волкова, А. А. Вылиток, Л. Е. Карпов

Тесты
по курсу «Системы программирования»

**Объектно-ориентированное программирование на Си++
и методы трансляции на основе формальных грамматик
и автоматов**

Учебное пособие для студентов II курса



МОСКВА – 2024

УДК 004.43 (075.8)
ББК 32.973-018.1я73
В67



<https://elibrary.ru/cthesi>

*Печатается по решению Редакционно-издательского совета
факультета вычислительной математики и кибернетики
МГУ имени М. В. Ломоносова*

Рецензенты:

И.В. Машечкин – д.ф.-м.н.;

Н.В. Лукашевич – д.т.н.

Волкова И. А., Вылиток А. А., Карпов Л. Е.

В67

Тесты по курсу «Системы программирования» (Объектно-ориентированное программирование на Си++ и методы трансляции на основе формальных грамматик и автоматов) : учебное пособие для студентов II курса / И.А. Волкова, А.А. Вылиток, Л.Е. Карпов. – Москва : Издательский отдел факультета ВМиК МГУ имени М.В. Ломоносова (лицензия ИД N 05899 от 24.09.2001 г.); МАКС Пресс, 2024. – 92 с.

e-ISBN: 978-5-89407-645-4 (ВМК МГУ имени М.В. Ломоносова)

ISBN 978-5-317-07274-2 (МАКС Пресс)

<https://doi.org/10.29003/m4204.978-5-317-07274-2>

В учебном пособии представлены тесты по языку Си++, основам трансляции и формальным грамматикам, рекомендуемые студентам 2-го курса для подготовки к коллоквиуму и экзамену в рамках курса «Системы программирования». В книге собраны тесты, проверяющие понимание студентами основных понятий ООП и систем программирования, основных механизмов объектно-ориентированных языков программирования, а также основных методов трансляции на основе формальных грамматик и автоматов. Тесты предлагались студентам в дистанционных проверочных работах в 2020–2022 гг.

Для студентов факультета ВМК в поддержку основного лекционного курса «Системы программирования», а также для преподавателей, ведущих практические занятия по этому курсу. Авторы выражают благодарность преподавателям кафедры алгоритмических языков за участие в обсуждении и составлении задач.

УДК 004.43(075.8)

ББК 32.973-018.1я73

e-ISBN: 978-5-89407-645-4
ISBN 978-5-317-07274-2

© Волкова И.А., Вылиток А.А., Карпов Л.Е., 2024

© Факультет ВМК МГУ имени М.В. Ломоносова, 2024

© Оформление. ООО «МАКС Пресс», 2024

Оглавление

Тесты	4
Перегрузка функций. Алгоритм выбора наилучшим образом подходящей функции (best matching)	4
Перегрузка операций	12
Конструкторы, деструкторы, исключения в C++	15
Механизм динамического полиморфизма в C++	26
Механизм RTTI в C++	30
STL, шаблоны, typename в C++	38
Классификация грамматик и языков по Хомскому	45
Вывод цепочки по заданной грамматике. Сентенциальные формы	50
Конечные автоматы. Преобразование НКА к ДКА	56
Множества first и follow для нетерминалов КС-грамматики	61
Перевод в ПОЛИЗ	63
Синтаксически управляемый перевод	65
Основные понятия ООП и систем программирования, применение основных механизмов C++	70
Ответы к тестам	78
Ответы к тестам по теме «Перегрузка функций. Алгоритм выбора наилучшим образом подходящей функции (best matching)».....	78
Ответы к тестам по теме «Перегрузка операций»	79
Ответы к тестам по теме «Конструкторы, деструкторы, исключения в C++».....	80
Ответы к тестам по теме «Механизм динамического полиморфизма в C++».....	81
Ответы к тестам по теме «Механизм RTTI в C++»	82
Ответы к тестам по теме «STL, шаблоны, typename в C++»	83
Ответы к тестам по теме «Классификация грамматик и языков по Хомскому»	84
Ответы к тестам по теме «Вывод цепочки по заданной грамматике. Сентенциальные формы».....	85
Ответы к тестам по теме «Конечные автоматы. Преобразование НКА к ДКА»	86
Ответы к тестам по теме «Множества first и follow для нетерминалов КС-грамматики».....	87
Ответы к тестам по теме «Перевод в ПОЛИЗ».....	88
Ответы к тестам по теме «Синтаксически управляемый перевод».....	89
Ответы к тестам по теме «Основные понятия ООП и систем программирования».....	90

Тесты

Перегрузка функций. Алгоритм выбора наилучшим образом подходящей функции (best matching)

1. Три функции объявлены в одной области видимости:

```
void f (char * , long);
```

```
void f (double, int);
```

```
void f ( ? , short);
```

Что можно вставить вместо знака вопроса в описании параметра одной из функций, чтобы вызов $f(0, 5L)$ вызвал именно эту функцию (с вопросом)? Укажите все правильные варианты.

- 1) ничего из перечисленного нельзя вставить
- 2) *char*
- 3) *double*
- 4) многоточие
- 5) *int*

2. Три функции объявлены в одной области видимости:

```
void g (long, char *);
```

```
void g (int, double);
```

```
void g (short, ? );
```

Что можно вставить вместо знака вопроса в описании параметра одной из функций, чтобы вызов $g(5L, 0)$ вызвал именно эту функцию (с вопросом)? Укажите все правильные варианты.

- 1) ничего из перечисленного нельзя вставить
- 2) *char*
- 3) *double*
- 4) многоточие
- 5) *int*

3. Три функции объявлены в одной области видимости:

```
void f (char * , short);
```

```
void f (double, int);
```

```
void f ( ? , short);
```

Что можно вставить вместо знака вопроса в описании параметра одной из функций, чтобы вызов $f(0, (\text{short})1)$ вызвал именно эту функцию (с вопросом)? Укажите все правильные варианты.

- 1) ничего из перечисленного нельзя вставить
- 2) *char* *
- 3) *double*
- 4) многоточие
- 5) *int*

4. Три функции объявлены в одной области видимости:

```
void g (long *, short);  
void g (short , double);  
void g (short *, ? );
```

Что можно вставить вместо знака вопроса в описании параметра одной из функций, чтобы вызов $g(0, 1)$ вызвал именно эту функцию (с вопросом)?
Укажите все правильные варианты.

- 1) ничего из перечисленного нельзя вставить
- 2) *char **
- 3) *double*
- 4) многоточие
- 5) *int*

5. Три функции объявлены в одной области видимости:

```
void f (short , int);  
void f ( ? , long);  
void f ( int, int);
```

Что можно вставить вместо знака вопроса в описании параметра одной из функций, чтобы вызов $f(5L, 1)$ вызвал именно эту функцию (с вопросом)?
Укажите все правильные варианты.

- 1) ничего из перечисленного нельзя вставить
- 2) *char*
- 3) *double*
- 4) многоточие
- 5) *int*

6. Три функции объявлены в одной области видимости:

```
void f (short , string);  
void f ( long, ? );  
void f ( int, char );
```

Что можно вставить вместо знака вопроса в описании параметра одной из функций, чтобы вызов $f(1.2, "a")$ вызвал именно эту функцию (с вопросом)?
Укажите все правильные варианты.

- 1) ничего из перечисленного нельзя вставить
- 2) *char*
- 3) *double*
- 4) многоточие
- 5) *char **

7. Три функции объявлены в одной области видимости:

```
void f ( short, std::string);  
void f ( long, ? );  
void f ( int, ... );
```

Что можно вставить вместо знака вопроса в описании параметра одной из функций, чтобы вызов `f(1.2, "a")` вызвал именно эту функцию (с вопросом)? Укажите все правильные варианты.

- 1) ничего из перечисленного нельзя вставить
- 2) `char`
- 3) `double`
- 4) многоточие
- 5) `char*`

8. Три функции объявлены в одной области видимости:

```
void f ( long double, char * );  
void f ( long, ? );  
void f ( int, ... );
```

Что можно вставить вместо знака вопроса в описании параметра одной из функций, чтобы вызов `f(1.2, "a")` вызвал именно эту функцию (с вопросом)? Укажите все правильные варианты.

- 1) ничего из перечисленного нельзя вставить
- 2) `char`
- 3) `double`
- 4) многоточие
- 5) `std::string`

9. Три функции объявлены в одной области видимости:

```
void f ( long double, char * );  
void f ( long, ? );  
void f ( int, std::string );
```

Что можно вставить вместо знака вопроса в описании параметра одной из функций, чтобы вызов `f(1.2, std::list())` вызвал именно эту функцию (с вопросом)? Укажите все правильные варианты.

- 1) ничего из перечисленного нельзя вставить
- 2) `char`
- 3) `double`
- 4) многоточие
- 5) `std::string`

10. Три функции объявлены в одной области видимости:

```
void g ( char *, long double );  
void g ( ? , int );  
void g ( std::list, short );
```

Что можно вставить вместо знака вопроса в описании параметра одной из функций, чтобы вызов `g(std::vector(), 1.2)` вызвал именно эту функцию (с вопросом)? Укажите все правильные варианты.

- 1) ничего из перечисленного нельзя вставить
- 2) *std::vector*
- 3) *std::list*
- 4) многоточие
- 5) *double*

11. Дана заготовка трех функций, описанных в одной области видимости:

```
void g (const char * , long long ) { }
void g(double, int ) { }
void g ( ? , short) { }
```

Что можно вставить вместо знака вопроса в описании параметра третьей функции, чтобы вызов *g(0, 5L)* вызвал именно эту функцию (с вопросом)? Укажите все правильные варианты.

- 1) ничего из перечисленного нельзя вставить
- 2) *char*
- 3) *double*
- 4) многоточие
- 5) *int*

12. Дана заготовка трех функций, описанных в одной области видимости:

```
void h (const char * , long ) { }
void h(double, unsigned int ) { }
void h ( ? , signed char) { }
```

Что можно вставить вместо знака вопроса в описании параметра третьей функции, чтобы вызов *h(0, 2L)* вызвал именно эту функцию (с вопросом)? Укажите все правильные варианты.

- 1) ничего из перечисленного нельзя вставить
- 2) *char*
- 3) *double*
- 4) многоточие
- 5) *int*

13. Дана заготовка трех функций, описанных в одной области видимости:

```
void f (short , unsigned short int ) { }
void f (unsigned long , long double ) { }
void f ( ? , const char*) { }
```

Что можно вставить вместо знака вопроса в описании параметра третьей функции, чтобы вызов *f(5u, 2L)* вызвал именно эту функцию (с вопросом)? Укажите все правильные варианты.

- 1) ничего из перечисленного нельзя вставить
- 2) *int*
- 3) *unsigned*
- 4) *float*
- 5) *unsigned short*

14. Дана заготовка трех функций, описанных в одной области видимости:

```
void x (float , double ) { }
void x (double , float ) { }
void x ( ? , short ) { }
```

Что можно вставить вместо знака вопроса в описании параметра третьей функции, чтобы вызов $x(5L, 5.0)$ вызвал именно эту функцию (с вопросом)? Укажите все правильные варианты.

- 1) ничего из перечисленного нельзя вставить
- 2) *unsigned int*
- 3) *char**
- 4) *char*
- 5) *long*

15. Дана заготовка трех функций, описанных в одной области видимости:

```
void g (char , long ) { }
void g (double , int ) { }
void g (long long, ? ) { }
```

Что можно вставить вместо знака вопроса в описании параметра третьей функции, чтобы вызов $g(0, 0)$ вызвал именно эту функцию (с вопросом)? Укажите все правильные варианты.

- 1) ничего из перечисленного нельзя вставить
- 2) *unsigned int*
- 3) *unsigned char*
- 4) *char**
- 5) *short*

16. Дана заготовка трех функций, описанных в одной области видимости:

```
void f (short , int ) { }
void f (char , int ) { }
void f ( ? , unsigned ) { }
```

Что можно вставить вместо знака вопроса в описании параметра третьей функции, чтобы вызов $f(0, 5U)$ вызвал именно эту функцию (с вопросом)? Укажите все правильные варианты.

- 1) ничего из перечисленного нельзя вставить
- 2) *unsigned short*
- 3) *signed short*
- 4) *unsigned long*
- 5) *signed long*

17. Дана заготовка трех функций, описанных в одной области видимости:

```
void h (bool , long) { }  
void h (float , int) { }  
void h (double , ? ) { }
```

Что можно вставить вместо знака вопроса в описании параметра третьей функции, чтобы вызов $h(0, 0.5)$ вызвал именно эту функцию (с вопросом)? Укажите все правильные варианты.

- 1) ничего из перечисленного нельзя вставить
- 2) *char*
- 3) *double*
- 4) многоточие
- 5) *int*

18. Дана заготовка трех функций, описанных в одной области видимости:

```
void y (int , int) { }  
void y (const char* , const char*) { }  
void y (char , ? ) { }
```

Что можно вставить вместо знака вопроса в описании параметра третьей функции, чтобы вызов $y(1, "abc")$ вызвал именно эту функцию (с вопросом)? Укажите все правильные варианты.

- 1) ничего из перечисленного нельзя вставить
- 2) *char*
- 3) *double*
- 4) многоточие
- 5) *int*

19. Дана заготовка трех функций, описанных в одной области видимости:

```
void f (int& , const char * ) { }  
void f (char , void * ) { }  
void f ( ? , const char * ) { }
```

Что можно вставить вместо знака вопроса в описании параметра третьей функции, чтобы вызов $f(0, 0)$ вызвал именно эту функцию (с вопросом)? Укажите все правильные варианты.

- 1) ничего из перечисленного нельзя вставить
- 2) *char*
- 3) *double*
- 4) многоточие
- 5) *int*

20. Дана заготовка трех функций, описанных в одной области видимости:

```
void w (const char * , const int & ) { }  
void w (const char * , char ) { }  
void w ( void * , ? ) { }
```

Что можно вставить вместо знака вопроса в описании параметра третьей функции, чтобы вызов $w(0, 2)$ вызвал именно эту функцию (с вопросом)? Укажите все правильные варианты.

- 1) ничего из перечисленного нельзя вставить
- 2) *char*
- 3) *double*
- 4) многоточие
- 5) *int*

21. Дана заготовка трех функций, описанных в одной области видимости:

```
struct A {  
    operator double() {return 1.0;}  
};  
void f(const char * , int);  
void f(char * , A);  
void f(double, ? );
```

Что можно вставить вместо знака вопроса в описании параметра третьей функции, чтобы вызов $f(0, 5.F)$ вызвал именно эту функцию (с вопросом)? Укажите все правильные варианты.

- 1) ничего из перечисленного нельзя вставить
- 2) *int*
- 3) *char*
- 4) *double*
- 5) А

22. Дана заготовка трех функций, описанных в одной области видимости:

```

struct A {
    operator double() { return 1.0; }
};
struct B : A {
    operator double() { return 1.0; }
};
void f( const char *, int );
void f( char *, double );
void f( double, ? );

```

Что можно вставить вместо знака вопроса в описании параметра третьей функции, чтобы вызов $f(0, B())$ вызвал именно эту функцию (с вопросом)? Укажите все правильные варианты.

- 1) ничего из перечисленного нельзя вставить
- 2) *int*
- 3) *char*
- 4) *double*
- 5) А

23. Дана заготовка трех функций, описанных в одной области видимости:

```

void f ( const char, unsigned long );
void f ( ?, long );
void f ( short, unsigned );

```

Что можно вставить вместо знака вопроса в описании параметра второй функции, чтобы вызов $f(5u, true)$ вызвал именно эту функцию (с вопросом)? Укажите все правильные варианты.

- 1) ничего из перечисленного нельзя вставить
- 2) *unsigned char*
- 3) *unsigned short*
- 4) *int*
- 5) *unsigned int*
- 6) *unsigned long*

Перегрузка операций

24. Вставьте в класс A самую короткую строку кода (без лишних пробелов) так, чтобы программа скомпилировалась без ошибок. Предупреждение компилятора (*warning*) при пустом теле функции (нет возвращаемого значения) не считаем ошибкой.

```
class A {  
public:  
    _____  
};  
int main(){  
    A a, b, c;  
    c = a + b;  
}
```

25. Вставьте в класс A самую короткую строку кода (без лишних пробелов) так, чтобы программа скомпилировалась без ошибок. Предупреждение компилятора (*warning*) при пустом теле функции (нет возвращаемого значения) не считаем ошибкой.

```
class A {  
public:  
    _____  
};  
int main(){  
    const A a, b;  
    A c;  
    c = a + b;  
}
```

26. Вставьте в класс A самую короткую строку кода (без лишних пробелов) так, чтобы программа скомпилировалась без ошибок. Предупреждение компилятора (*warning*) при пустом теле функции (нет возвращаемого значения) не считаем ошибкой.

```
class A {  
public:  
    _____  
};  
int main(){  
    A a, b, c;  
    c = a * b;  
}
```

27. Вставьте в класс A самую короткую строку кода (без лишних пробелов) так, чтобы программа скомпилировалась без ошибок. Предупреждение компилятора (*warning*) при пустом теле функции (нет возвращаемого значения) не считаем ошибкой.

```

class A {
public:
    _____
};
int main(){
    const A b;
    A c;
    c = -b;
}

```

28. Вставьте в класс A самую короткую строку кода (без лишних пробелов) так, чтобы программа скомпилировалась без ошибок. Предупреждение компилятора (*warning*) при пустом теле функции (нет возвращаемого значения) не считаем ошибкой.

```

class A {
public:
    _____
};
int main(){
    A c;
    c = --c;
}

```

29. Вставьте в класс A самую короткую строку кода (без лишних пробелов) так, чтобы программа скомпилировалась без ошибок. Предупреждение компилятора (*warning*) при пустом теле функции (нет возвращаемого значения) не считаем ошибкой.

```

class A {
public:
    _____
};
int main(){
    A c;
    c = c++;
}

```

30. Вставьте в класс A самую короткую строку кода (без лишних пробелов) так, чтобы программа скомпилировалась без ошибок. Предупреждение компилятора (*warning*) при пустом теле функции (нет возвращаемого значения) не считаем ошибкой.

```

class A {
public:
    _____
};
int main(){
    A c;
    ~c;
}

```

31. Вставьте в класс A самую короткую строку кода (без лишних пробелов) так, чтобы программа скомпилировалась без ошибок.

```
class T {  
public:  
    int a;  
};  
class A {  
public:  
    T * p;  
};  
int main(){  
    A c;  
    C -> a;  
}
```

32. Вставьте в класс A самую короткую строку кода (без лишних пробелов) так, чтобы программа скомпилировалась без ошибок. Предупреждение компилятора (*warning*) при пустом теле функции (нет возвращаемого значения) не считаем ошибкой.

```
class A {  
public:  
};  
int main(){  
    A c;  
    (int)c;  
}
```

33. Вставьте в класс A самую короткую строку кода (без лишних пробелов) так, чтобы программа скомпилировалась без ошибок. Предупреждение компилятора (*warning*) при пустом теле функции (нет возвращаемого значения) не считаем ошибкой.

```
class A {  
public:  
};  
int main(){  
    A c;  
    c();  
}
```

Конструкторы, деструкторы, исключения в C++

34. Вставьте (без лишних пробелов) пропущенный параметр операции **throw** так, чтобы при выполнении данного фрагмента программы на экран напечаталось
"A() A() A(const A&) ~A() ~A() ~A()"
при условии, что компилятор **ничего не оптимизирует**.

```
class A {
public:
    A() { cout<< "A() ";}
    A(const A & s) { cout<< "A(const A &) ";}
    ~A() { cout<< "~A() ";}
};
int main(){
    A a;
    try { throw _____; }
    catch(A &) { }
    catch(string) {cout<< "~A()";}
    catch(...) {cout<< "A(const A&) ~A() ~A() ~A()";}
}
```

35. Вставьте (без лишних пробелов) пропущенный параметр операции **throw** так, чтобы при выполнении данного фрагмента программы на экран напечаталось
"A() A(const A&) ~A() ~A()"
при условии, что компилятор **ничего не оптимизирует**.

```
class A {
public:
    A() { cout<< "A() ";}
    A(const A & s) { cout<< "A(const A &) ";}
    ~A() { cout<< "~A() ";}
};
int main(){
    try { throw _____; }
    catch(A &) { }
    catch(string) {cout<< "~A()";}
    catch(...) {cout<< "A(const A&) ~A() ~A()";}
}
```

36. Вставьте (без лишних пробелов) пропущенный параметр операции **throw** так, чтобы при выполнении данного фрагмента программы на экран напечаталось
"A() A(const A&) A(const A&) ~A() ~A() ~A()"
при условии, что компилятор **ничего не оптимизирует**.

```
class A {
public:
    A() { cout<< "A() ";}
    A(const A & s) { cout<< "A(const A &) ";}
    ~A() { cout<< "~A() ";}
};
```



```

int main(){
    A a;
    try { throw _____; }
    catch(A a) { }
    catch(string) {cout<< "~A()";}
    catch(...) {cout<< "A(const A&) ~A() ~A() ~A()";}
}

```

37. Вставьте (без лишних пробелов) пропущенный параметр операции **throw** так, чтобы при выполнении данного фрагмента программы на экран напечаталось
"A() A(const A&) ~A() A(const A&) ~A() ~A()"
 при условии, что компилятор **ничего не оптимизирует**.

```

class A {
public:
    A() { cout<< "A() ";}
    A(const A & s) { cout<< "A(const A &) ";}
    ~A() { cout<< "~A() ";}
};
int main(){
    try { throw _____ ; }
    catch(A a) { }
    catch(string) {cout<< "~A()";}
    catch(...) {cout<< "A(const A&) ~A() ~A()";}
}

```

38. Вставьте (без лишних пробелов) пропущенный параметр операции **throw** так, чтобы при выполнении данного фрагмента программы на экран напечаталось
"A() A(const A&) ~A() A() ~A() ~A()"
 при условии, что компилятор **ничего не оптимизирует**.

```

class A {
public:
    A() { cout<< "A() ";}
    A(const A & s) { cout<< "A(const A &) ";}
    ~A() { cout<< "~A() ";}
};
int main(){
    try { throw _____; }
    catch(A &) { A b; }
    catch(int) {cout<< "~A()";}
    catch(...) {cout<< "A() ~A() ~A()";}
}

```

39. Вставьте (без лишних пробелов) пропущенный параметр операции **throw** так, чтобы при выполнении данного фрагмента программы на экран напечаталось

"A() A(const A&) ~A() A(const A&) A() ~A() ~A() ~A()" при условии, что компилятор **ничего не оптимизирует**.

```
class A {
    public:
        A() { cout<< "A() ";}
        A(const A & s) { cout<< "A(const A &) ";}
        ~A() { cout<< "~A() ";}
};
int main(){
    try { throw _____; }
    catch(A a) { A b; }
    catch(int) {cout<< "~A() ";}
    catch(...) {cout<< "A() ~A() ~A() ~A()";}
}
```

40. Вставьте (без лишних пробелов) пропущенный параметр операции **throw** так, чтобы при выполнении данного фрагмента программы на экран напечаталось "A() A(const A&) A() ~A() ~A() ~A()" при условии, что компилятор **ничего не оптимизирует**.

```
class A {
    public:
        A() { cout<< "A() ";}
        A(const A & s) { cout<< "A(const A &) ";}
        ~A() { cout<< "~A() ";}
};
int main(){
    A a;
    try { throw _____; }
    catch(A &) { A b; }
    catch(int) {cout<< "~A() ";}
    catch(...) {cout<< "~A() ~A() ~A()";}
}
```

41. Вставьте (без лишних пробелов) пропущенный параметр операции **throw** так, чтобы при выполнении данного фрагмента программы на экран напечаталось "R() R(const R&) R() ~R() ~R() ~R()" при условии, что компилятор **ничего не оптимизирует**.

```
class R {
    public:
        R() { cout<< "R() ";}
        R(const R & s) { cout<< "R(const R &) ";}
        ~R() { cout<< "~R() ";}
};
```

```

int main(){

    R r;
    try { throw _____; }
    catch(R &) { R b; }
    catch(int) {cout<< "~R() ";}
    catch(...) {cout<< "~R() ~R() ~R()";}
}

```

42. Вставьте (без лишних пробелов) пропущенный параметр операции **throw** так, чтобы при выполнении данного фрагмента программы на экран напечаталось

"B() B(const B&) ~B() B(const B&) B() ~B() ~B() ~B()"

при условии, что компилятор **ничего не оптимизирует**.

```

class B {
public:
    B() { cout<< "B() ";}
    B(const B & s) { cout<< "B(const B &) ";}
    ~B() { cout<< "~B() ";}
};
int main(){
    try { throw _____; }
    catch(B a) { B b; }
    catch(int) {cout<< "~B() ";}
    catch(...) {cout<< "B() ~B() ~B() ~B()";}
}

```

43. Вставьте (без лишних пробелов) пропущенный параметр оператора **throw** так, чтобы при выполнении данного фрагмента программы на экран напечаталось

"B() B(const B&) B(const B&) ~B() ~B() ~B()"

при условии, что компилятор **ничего не оптимизирует**.

```

class B {
public:
    B() { cout<< "B() ";}
    B(const B & s) { cout<< "B(const B &) ";}
    ~B() { cout<< "~B() ";}
};
int main(){

    B b;
    try { throw _____; }
    catch(B a) { }
    catch(string) {cout<< "~B() ";}
    catch(...) {cout<< "B(const B&) ~B() ~B() ~B()";}
}

```

44. Вставьте в заготовку программы пропущенное целое число так, чтобы в результате работы полученной программы напечаталось число 4. (Считаем, что компилятор ничего не оптимизирует и все необходимые файлы и директивы включены)

```
class X {
public:
    static int k;
    X() { k+=2; }
    X( const X& ) { k += ____; }
    ~X() { k -= 1; }
};

void f( X& x, int n ) {
    try { if ( n == 0 ) throw a; else throw 1; }
    catch(int x) { f( a, x - 1 ); }
    catch(X & x) { throw ; }
}

int X::k = 0;
int main() {
    X a;
    try { f( a, 10 ); }
    catch ( ... ) { }
    cout << a.k;
}
```

45. Вставьте в заготовку программы пропущенное целое число так, чтобы в результате работы полученной программы напечаталось число 5. (Считаем, что компилятор ничего не оптимизирует и все необходимые файлы и директивы включены)

```
class B {
public:
    static int k;
    B() { k +=2; }
    B( const B& ) { k +=____; }
    ~B() { k -= 1; }
};

void f( B& b, int n ) {
    try { if ( n == 0 ) throw B (); else throw 1; }
    catch (int x) { f(b, x-1); }
    catch (B & b) { throw ; }
}

int B::k = 0;
int main () {
    B b;
    try { f( b, 10 ); }
    catch (...) { }
    cout << b.k;
}
```

46. Вставьте в заготовку программы пропущенное целое число так, чтобы в результате работы полученной программы напечаталось число 4. (Считаем, что компилятор ничего не оптимизирует и все необходимые файлы и директивы включены)

```
class A {
public:
    static int n;
    A() { n += 2; }
    A( const A& ) { n += ____; }
    ~A() { n -= 1; }
};
void f( A& a, int n ) {
    try { if ( n == 0 ) throw A (); else throw 1; }
    catch ( int x ) { f( a, x - 1 ); }
    catch ( A & a ) { throw ; }
}
int A::n = 0;
int main() {
    A a;
    try { f( a, 10 ); }
    catch ( ... ) { }
    cout << A::n;
}
```

47. Вставьте в заготовку программы пропущенное целое число так, чтобы в результате работы полученной программы напечаталось число 4. (Считаем, что компилятор ничего не оптимизирует и все необходимые файлы и директивы включены)

```
class C {
public:
    static int k;
    C() { k += 2; }
    C( const C& ) { k += ____; }
    ~C() { k -= 1; }
};
void f( C& a, int n ) {
    try { if ( n == 0 ) throw a; else throw 1; }
    catch ( int x ) { f( a, x - 1 ); }
    catch ( C & a ) { throw ; }
}
int C::k = 0;
int main () {
    C a;
    try { f( a, 10 ); }
    catch ( ... ) { }
    cout << C::k;
}
```

48. Вставьте в заготовку программы пропущенное целое число так, чтобы в результате работы полученной программы напечаталось число 8. (Считаем, что компилятор ничего не оптимизирует и все необходимые файлы и директивы включены)

```

struct A {
    static int n;
    virtual ~A() {}
};
struct D:A {
    D() { n++; }
    D( const D & d ) { n = n + ____;}
    ~D() { --n; }
};
void f(A & b) {
    D d1 = dynamic_cast ( b );
    throw d1;
}
int A::n = 1;
int main() {
    try {
        D d;
        f(d);
    }
    catch ( A & ) { cout << A::n; }
}

```

49. Вставьте в заготовку программы пропущенное целое число так, чтобы в результате работы полученной программы напечаталось число 5. (Считаем, что компилятор ничего не оптимизирует и все необходимые файлы и директивы включены)

```

struct C {
    static int n;
    virtual ~C(){}
};
struct D:C {
    D() { n++; }
    D( const D & d ){ n = n + ____;}
    ~D() { --n;}
};
void f( D d ) {
    D* pd = &d;
    throw pd;
}
int C::n = 1;
int main () {
    try {
        D d;
        f(d);
    }
}

```

```

    }
    catch(C * ) { cout << C::n << '\n'; }
}

```

50. Вставьте в заготовку программы пропущенное целое число так, чтобы в результате работы полученной программы напечаталось число 1. (Считаем, что компилятор ничего не оптимизирует и все необходимые файлы и директивы включены)

```

struct E {
    static int n;
    E(){ --n;}
    E( const E & ){ n += ____;}
    virtual ~E(){}
};
struct D : E {
    D(){ --n; }
    ~D(){}
};
void f( E & b ) {
    E b1 = b;
    throw b1;
}
int E::n = 1;
int main () {
    D d;
    try {
        f( d );
    }
    catch(E &) { cout << d.n << '\n'; }
}

```

51. Вставьте в заготовку программы пропущенное целое число так, чтобы в результате работы полученной программы напечаталось число 9. (Считаем, что компилятор ничего не оптимизирует и все необходимые файлы и директивы включены)

```

struct Bs {
    static int n;
    virtual ~Bs() {}
};
struct Drv : Bs {
    Drv() { ++n; }
    Drv( const Drv & d ) { n = n + ____; }
    ~Drv() { --n; }
};
void f( Bs & b ) {
    Drv d = dynamic_cast <Drv &> ( b );
    throw d;
}

```

```

int Bs::n;
int main () {
    int m;
    try {
        Drv d;
        f( d );
    }
    catch( Bs &      ) { m = Bs::n; }
    cout << m + Bs::n;
}

```

52. Вставьте в заготовку программы пропущенное целое число так, чтобы в результате работы полученной программы напечаталось число 2. Считаем, что (Считаем, что компилятор ничего не оптимизирует и все необходимые файлы и директивы включены)

```

class B {
public:
    static int k;
    B() { k += 2; }
    B( const B& ) { k += ____; }
    ~B() { k -= 1; }
};
void f ( B & a, int n ) {
    try { if ( n == 0 ) throw B (); else throw 1; }
    catch( int x ) { f ( a, x - 1 ); }
    catch( B & a ) { throw; }
}
int B::k = -1;
int main () {
    try { B a; f ( a, 10 ); }
    catch ( ... ) { cout << B::k; }
}

```

53. Вставьте в заготовку программы пропущенное число так, чтобы в результате работы полученной программы напечаталось число 55. (Считаем, что компилятор ничего не оптимизирует и все необходимые файлы и директивы включены)

```

class A {
public:
    static int k;
    A() { k += 2; }
    A( const A& ) { k += ____; }
    ~A() { k -= 1; }
};
void f( A & a, int n ) {
    try { if ( n == 0 ) throw A (); else throw 1; }
    catch ( int x ) { f( a, x - 1 ); }
    catch ( A & a ) { throw; }
}

```



```

int A::k = -1;
int main () {
    try { A a; f(a, 10); }
    catch ( A a ) { cout << A::k * 10 + a.k; }
}

```

54. Дана программа на языке C++. Есть ли в ней ошибки?

Что будет выведено на печать в случае отсутствия ошибок? Компилятор **ничего не оптимизирует**.

(Считаем, что все необходимые файлы и директивы включены. Если есть ошибка компиляции, то ошибки выполнения нет, так как исполняемый файл не сгенерируется)

```

class A {
    A(const A&) { cout << 2; }
public:
    A() { cout << 1; }
};
void f( A& a, int n ) {
    try {
        if ( n == 0 ) throw a;
        else throw 1;
    }
    catch ( int x ) { cout << "catch (int x)\n" ; }
    catch ( A & a ) { cout << "catch (A& a)\n" ; }
}
int main () {
    A a;
    f ( a, 10 );
}

```

- 1) есть ошибка компиляции
- 2) есть ошибка выполнения
- 3) 1catch (int x)
- 4) 1catch (A& a)
- 5) 12catch (A& a)

55. Дана программа на языке C++. Есть ли в ней ошибки?

Что будет выведено на печать в случае отсутствия ошибок? Компилятор **ничего не оптимизирует**.

(Считаем, что все необходимые файлы и директивы включены. Если есть ошибка компиляции, то ошибки выполнения нет, так как исполняемый файл не сгенерируется)

```

class A {
public:
    A ( const A& ra = A ( 1 ) ) { cout << 2; }
    A ( int ) { cout << 1; }
};

```

```

void f( int n ) {
    try {
        if ( n == 0 ) throw A ();
        else throw 1;
    }
    catch( int x ) { cout << "catch (int x)\n" ; }
    catch( A & a ) { cout << "catch (A& a)\n" ; }
}
int main () {
    f (0);
}

```

- 1) есть ошибка компиляции
- 2) есть ошибка выполнения
- 3) 12catch (int x)
- 4) 122catch (A& a)
- 5) 12catch (A& a)

56. Дана программа на языке С++. Есть ли в ней ошибки?

Что будет выведено на печать в случае отсутствия ошибок? Компилятор **ничего не оптимизирует**.

(Считаем, что все необходимые файлы и директивы включены. Если есть ошибка компиляции, то ошибки выполнения нет, так как исполняемый файл не сгенерируется)

```

class A {
    A() { cout << 1; }
public:
    A(const A &) { cout << 2; }
};
void f( int n ) {
    try {
        if ( n == 0 ) throw A();
        else throw 1;
    }
    catch ( int x ) { cout << "catch (int x)\n"; }
    catch ( A & a ) { cout << "catch (A& a)\n"; }
}
int main() {
    A a;
    f( 15 );
}

```

- 1) есть ошибка компиляции
- 2) есть ошибка выполнения
- 3) 12catch (int x)
- 4) 122catch (A& a)
- 5) 12catch (A& a)

Механизм динамического полиморфизма в C++

57. Заполните пропуск (наиболее кратко) так, чтобы в результате работы фрагмента программы напечаталось "B::f()".

```
class A {
public:
    virtual void f() const {cout << "A::f() ";}
};
class B: public A {
public:
    void f() const {cout << "B::f() ";}
};
int main(){
    const A & a= B();
    a _____;
}
```

58. Заполните пропуск (наиболее кратко) так, чтобы в результате работы фрагмента программы напечаталось "A::f()".

```
class A {
public:
    virtual void f() const {cout<< "A::f() ";}
};
class B: public A {
public:
    void f() const {cout<< "B::f() ";}
};
int main(){
    const A & a= B();
    a _____;
}
```

59. Заполните пропуск (наиболее кратко) так, чтобы в результате работы фрагмента программы напечаталось "C::f()".

```
class C {
public:
    void f() const {cout<< "C::f() ";}
};
class B: public C {
public:
    virtual void f() const {cout<< "B::f() ";}
};
int main(){
    const C & c= B();
    c _____;
}
```

60. Заполните пропуск (наиболее кратко) так, чтобы в результате работы фрагмента программы напечаталось "B::f()".

```
class A {
public:
    virtual void f() const {cout<< "A::f() ";}
};
class B: public A {
public:
    virtual void f() const {cout<< "B::f() ";}
};
int main(){
    B b;
    A &a =b;
    a _____;
}
```

61. Заполните пропуск (наиболее кратко) так, чтобы в результате работы фрагмента программы напечаталось "A::f()".

```
class A {
public:
    virtual void f() {cout << "A::f() ";}
};
class B: public A {
public:
    virtual void f() const {cout<< "B::f() ";}
};
int main(){
    B b;
    A & a =b;
    a _____;
}
```

62. Заполните пропуск (наиболее кратко) так, чтобы в результате работы фрагмента программы напечаталось "A::f()".

```
class A {
public:
    virtual void f() {cout<< "A::f() ";}
};
class B: public A {
public:
    virtual void f() const {cout<< "B::f() ";}
};
int main(){
    B b;
    A * a = &b;
    a _____;
}
```

63. Заполните пропуск (наиболее кратко) так, чтобы в результате работы фрагмента программы напечаталось "A::f()".

```
class A {
public:
    void f() {cout << "A::f() "};
};
class B: public A {
public:
    virtual void f() {cout<< "B::f() "};
};
int main(){
    B b;
    A * a = &b;
    a _____;
}
```

64. Заполните пропуск (наиболее кратко) так, чтобы в результате работы фрагмента программы напечаталось "A::f()".

```
class A {
public:
    virtual void f() {cout<< "A::f() "};
};
class B: public A {
public:
    virtual void f() {cout<< "B::f() "};
};
int main(){
    B b;
    A * a = &b;
    a _____;
}
```

65. Заполните пропуск (наиболее кратко) так, чтобы в результате работы фрагмента программы напечаталось "A::g()".

```
class A {
public:
    virtual void g() {cout<< "A::g() "};
};
class B: public A {
public:
    void g() {cout<< "B::g() "};
};
int main(){
    B b;
    A * a = &b;
    a _____;
}
```

66. Заполните пропуск (наиболее кратко) так, чтобы в результате работы фрагмента программы напечаталось "B::g()=1".

```
class A {
public:
    virtual void g(int i=1){cout<< "A::g()=" << i;}
};
class B: public A {
public:
    void g(int i=2){cout << "B::g()=" << i;}
};
int main(){
    B b;
    A * a = &b;
    a _____;
}
```

67. Дана программа на языке C++. Есть ли в ней ошибки? Что будет выведено на печать в случае отсутствия ошибок? (Считаем, что все необходимые файлы и директивы включены)

```
struct A {
    virtual void f() { cout << "A::f()\n"; }
};
struct B : A {
private:
    void f() { cout << "B::f()\n"; }
};
int main () {
    B b;
    A* pa = &b;
    pa->f();
}
```

- 1) есть ошибка компиляции
- 2) A::f()
- 3) B::f()
- 4) A::f()
- 5) B::f()

Механизм RTTI в C++

68. Что будет напечатано следующей программой?
(Считаем, что все необходимые файлы и директивы включены)

```
class spa {
public:
    spa * this;
    spa () { this = this ; }
    virtual spa* operator ->() { cout << 1; return this; }
};
class spb: public spa {
};
int main() {
    spa s1; spb s2;

    if ( dynamic_cast ( s1 -> this ) ) cout << 2;

    else cout << 3;

}
```

- 1) 13
- 2) 12
- 3) 3
- 4) 2
- 5) ошибка при выполнении

69. Что будет напечатано следующей программой?
(Считаем, что все необходимые файлы и директивы включены)

```
class spa {
public:
    spa * this;
    spa() { this = this; }
    virtual spa* operator ->() { cout << 1; return this; }
};
class spd: public spa {
};
int main() {
    spa s1; spd s2;
    if (dynamic_cast <spd*>(s2 -> this)) cout << 2;
    else cout << 3;
}
```

- 1) 13
- 2) 12
- 3) 3
- 4) 2
- 5) ошибка при выполнении

70. Что будет напечатано следующей программой?
(Считаем, что все необходимые файлы и директивы включены)

```
class spa {  
public:  
    spa * this;  
    spa() { this = this; }  
    virtual spa * operator () ( spa & ) {  
        cout << 1; return this;  
    }  
};  
class spd : public spa { public: };  
int main () {  
    spa s1; spd s2;  
    if ( dynamic_cast <spd*> ( s2 ( s2 ) ) ) cout << 2;  
    else cout << 3;  
}
```

- 1) 13
- 2) 12
- 3) 3
- 4) 2
- 5) ошибка при выполнении

71. Что будет напечатано следующей программой?
(Считаем, что все необходимые файлы и директивы включены)

```
class spc {  
public:  
    spc * this;  
    spc() { this = this; }  
    virtual spc * operator () ( spc & ) {  
        cout << 1; return this;  
    }  
};  
class spd : public spc { public: };  
  
int main () {  
    spc s1; spd s2;  
    if ( dynamic_cast <spd*> ( s1 (s2)) ) cout << 2;  
    else cout << 3;  
}
```

- 1) 13
- 2) 12
- 3) 3
- 4) 2
- 5) ошибка при выполнении

72. Что будет напечатано следующей программой?

(Считаем, что все необходимые файлы и директивы включены)

```
class spe {
public:
    spe * this;
    spe() { this = this; }
    virtual spe * operator [] ( spe & ) { cout << 1; return this; }
};

class spd : public spe { public: };

int main () {
    spe s1; spd s2;
    if ( dynamic_cast <spd*> ( s2 [s2]) ) cout << 2;
    else cout << 3;
}
```

- 1) 13
- 2) 12
- 3) 3
- 4) 2
- 5) ошибка при выполнении

73. Что будет напечатано следующей программой?

(Считаем, что все необходимые файлы и директивы включены)

```
class spa {
public:
    spa * this;
    spa() { this = this; }
    virtual spa * operator [] ( spa & ) { cout << 1; return this; }
};

class spd : public spa { public: };

int main () {
    spa s1; spd s2;
    if ( dynamic_cast <spd*> ( s1 [s2]) ) cout << 2;
    else cout << 3;
}
```

- 1) 13
- 2) 12
- 3) 3
- 4) 2
- 5) ошибка при выполнении

74. Что будет напечатано следующей программой?

(Считаем, что все необходимые файлы и директивы включены)

```
struct A {
    int n = 2;
    virtual A& operator * ( const A& b ) { n *= b.n; return *this; }
};
struct D : A {
    D & operator* ( const D& d ) { n *= d.n; return *this; }
};
A & f ( A & b ) {
    if ( typeid ( b ) == typeid ( D ) )
        cout << 1;
    else
        cout << 2;
    return b;
}
int main () {
    A b;
    D d;
    cout << ( f( d ) * f( b ) ).n << endl;
}
```

- 1) 124
- 2) 224
- 3) 128
- 4) 228
- 5) ошибка при выполнении

75. Что будет напечатано следующей программой?

(Считаем, что все необходимые файлы и директивы включены)

```
struct B {
    int n = 2;
    B & operator* ( const B& b ) {n *= b.n; return *this; }
};
struct D : B {
    D & operator* ( const D & d ) {n *= d.n; return *this; }
};
B & f( B & b ) {
    if ( typeid( b ) == typeid( D ) )
        cout << 1;
    else
        cout << 2;
    return b;
}
int main() {
    B b;
    D d;
    cout << ( f( d ) * f b ) ).n << endl; }
}
```

- 1) 124
- 2) 224
- 3) 128
- 4) 228
- 5) ошибка при выполнении

76. Что будет напечатано следующей программой?
(Считаем, что все необходимые файлы и директивы включены)

```

struct E {
    int n = 10;
    virtual E* operator- ( const E * b ) { n = b -> n - 3; return this; }
};
struct D : E {
    D * operator - ( const D * d ) { n = d -> n - 5; return this; }
};

E & f(E & b) {
    if ( typeid ( b ) == typeid ( D ) )
        cout << 1;
    else
        cout << 2;
    return b;
}
int main () {
    E b;
    D d;
    cout << ( f( d ) - &f( b ) )-> n << endl;
}

```

- 1) 121
- 2) 221
- 3) 127
- 4) 227
- 5) ошибка при выполнении

77. Что будет напечатано следующей программой?
(Считаем, что все необходимые файлы и директивы включены)

```

struct A {
    int n = 10;
    A* operator- ( const A * b ) {
        n = b -> n - 3;
        return this;
    }
};
struct D : A {
    D* operator- ( const D * d ) { n = d -> n - 5; return this; }
};

```

```

A & f( A & b ) {
    if ( typeid ( b ) == typeid ( D ) )
        cout << 1;
    else
        cout << 2;
    return b;
}
int main () {
    A b;
    D d;
    cout << ( f( d ) - &f( b ) )-> n << endl;
}

```

- 1) 121
- 2) 221
- 3) 127
- 4) 227
- 5) ошибка при выполнении

78. Что будет напечатано следующей программой?
(Считаем, что все необходимые файлы и директивы включены)

```

struct B {
    virtual operator int () { return 7; }
};
struct D : B {
    operator int() { return 12; }
};

B & f( B & b ) {
    if ( typeid ( b ) == typeid ( D ) )
        cout << 1;
    else
        cout << 3;
    return b;
}

int main () {
    B b;
    D d;
    cout << f( d ) - f( b ) << endl;
}

```

- 1) 110
- 2) 130
- 3) 135
- 4) 115
- 5) 335
- 6) ошибка компиляции

79. Что будет напечатано следующей программой?
(Считаем, что все необходимые файлы и директивы включены)

```
struct B {
    virtual operator int () { return 7; }
};
struct D : B {
    operator int () { return 12; }
};
B & f(B & b) {
    if ( typeid ( b ) == typeid ( D ) )
        cout << 1;
    else
        cout << 2;
    return b;
}
int main() {
    B b;
    D d;
    cout << f( d ) - f( b ) << endl;
}
```

- 1) 110
- 2) 120
- 3) 125
- 4) 115
- 5) 225
- 6) ошибка компиляции

80. Что будет напечатано следующей программой?
(Считаем, что все необходимые файлы и директивы включены)

```
struct B {
    virtual operator int () { return 5; }
};
struct D : B {
    operator int () { return 13; }
};
B & g(B & b) {
    if ( typeid ( b ) == typeid ( B () ) )
        cout << 1;
    else
        cout << 2;
    return b;
}
int main () {
    B b;
    D d;
    cout << g ( d ) - g ( b ) << endl;
}
```

- 1) 120
- 2) 210
- 3) 218
- 4) 118
- 5) 228
- 6) ошибка компиляции

STL, шаблоны, typename в C++

81. Есть ли ошибки в данном фрагменте программы?

Считаем, что все необходимые файлы и директивы включены.

(Если есть ошибки компиляции, то ошибок выполнения нет, так как исполняемый файл программы не создается)

```
template < class Y >
typename Y::const_iterator f(const Y & x, typename Y::value_type v ){
    typename Y :: const_iterator p = x.end();
    while (p != x.begin() )
        if ( * -- p == v ) return p;
    return x.end();
}
int main() {
    list g;
    g.push_back(1);
    f(g,7);
}
```

- 1) нет ошибок
- 2) есть ошибки компиляции
- 3) есть ошибки выполнения

82. Есть ли ошибки в данном фрагменте программы?

Считаем, что все необходимые файлы и директивы включены.

(Если есть ошибки компиляции, то ошибок выполнения нет, так как исполняемый файл программы не создается)

```
template < class X >
X::const_iterator f( const X & x, X::value_type v ){
    X :: const_iterator p = x.end();
    while (p != x.begin() )
        if ( * -- p == v ) return p;
    return x.end();
}
int main() {
    list g;
    g.push_back(1);
    f(g,7);
}
```

- 1) нет ошибок
- 2) есть ошибки компиляции
- 3) есть ошибки выполнения

83. Есть ли ошибки в данном фрагменте программы?

Считаем, что все необходимые файлы и директивы включены.

(Если есть ошибки компиляции, то ошибок выполнения нет, так как исполняемый файл программы не создается)

```
template < class S, int n > S f(){
    S s = 0;
    for ( ; n ; --n) { s += n; }
    return s;
}
int main() {
    cout << f <int, 10> () << endl;
}
```

- 1) нет ошибок
- 2) есть ошибки компиляции
- 3) есть ошибки выполнения

84. Есть ли ошибки в данном фрагменте программы?

Считаем, что все необходимые файлы и директивы включены.

(Если есть ошибки компиляции, то ошибок выполнения нет, так как исполняемый файл программы не создается)

```
template < class S, double d > S f() {
    S s = 0;
    for ( int i = 0; i < 10 ; ++i ) { s += d; }
    return s;
}
int main () {
    double d = 1.5;
    cout << f < int, d > () << endl;
}
```

- 1) нет ошибок
- 2) есть ошибки компиляции
- 3) есть ошибки выполнения

85. Есть ли ошибки в данном фрагменте программы?

Считаем, что все необходимые файлы и директивы включены.

(Если есть ошибки компиляции, то ошибок выполнения нет, так как исполняемый файл программы не создается)

```
template < class S > S f( list <S> & l ) {
    typename list <S>::iterator i = l.begin();
    i = i+1;
    return *i;
}
```



```

int main () {
    list <int> l = { 1, 2, 3 };
    cout << f< int>( l ) << endl;
}

```

- 1) нет ошибок
- 2) есть ошибки компиляции
- 3) есть ошибки выполнения

86. Есть ли ошибки в данном фрагменте программы?

Считаем, что все необходимые файлы и директивы включены.

(Если есть ошибки компиляции, то ошибок выполнения нет, так как исполняемый файл программы не создается)

```

template < class A > A f( vector <A> & v ) {
    typename vector <A>::iterator i = v.begin();
    i = i + 1;
    return *i;
}
int main() {
    vector <int> v = { 1, 2, 3 };
    cout << f<int>( v ) << endl;
}

```

- 1) нет ошибок
- 2) есть ошибки компиляции
- 3) есть ошибки выполнения

87. Есть ли ошибки в данном фрагменте программы?

Считаем, что все необходимые файлы и директивы включены.

(Если есть ошибки компиляции, то ошибок выполнения нет, так как исполняемый файл программы не создается)

```

template < class Y >
typename Y::const_iterator f( const Y & x, typename Y::value_type v ){
    typename Y::const_iterator p = x.end();
    while ( p != x.begin() )
    if ( * ( p-- ) == v )
        return p;
    return x.end();
}
int main(){
    list g;
    g.push_back ( 1 );
    f( g, 7 );
}

```

- 1) нет ошибок
- 2) есть ошибки компиляции
- 3) есть ошибки выполнения

88. Есть ли ошибки в данном фрагменте программы?

Считаем, что все необходимые файлы и директивы включены.

(Если есть ошибки компиляции, то ошибок выполнения нет, так как исполняемый файл программы не создается)

```

template < class Z >
typename Z::const_iterator f( const Z & x, typename Z::value_type v ){
    typename Z::const_iterator p = x.end ( );
    while ( p != x.begin ( ) ){
        cout << "q";
        if ( * ( p -- ) == v )
            return p;
    }
    return x.end();
}
int main(){
    vector g;
    g.push_back( 1 );
    f( g, 7 );
}

```

- 1) нет ошибок
- 2) есть ошибки компиляции
- 3) есть ошибки выполнения

89. Есть ли ошибки в данном фрагменте программы?

Считаем, что все необходимые файлы и директивы включены.

(Если есть ошибки компиляции, то ошибок выполнения нет, так как исполняемый файл программы не создается)

```

template < class B >
typename B::iterator f(const B & x, B * v ){
    typename B :: iterator p = x.end ( );
    while ( p != x.begin ( ) )
        if ( * -- p == * v ) return p;
    return x.end ( );
}
int main(){
    vector g;
    g.push_back ( 1 ) ;
    f( g, &g );
}

```

- 1) нет ошибок
- 2) есть ошибки компиляции
- 3) есть ошибки выполнения

90. Есть ли ошибки в данном фрагменте программы?

Считаем, что все необходимые файлы и директивы включены.

(Если есть ошибки компиляции, то ошибок выполнения нет, так как исполняемый файл программы не создается)

```

template < class Y >
typename Y::reverse_iterator f(const Y & x, typename Y::value_type v ){
    typename Y::const_reverse_iterator p = x.rbegin ();
    while ( p != x.rend () )
        if ( * -- p == x.front () ) return p;
    return x.end ();
}
int main(){
    list g;
    g.push_back ( 1 );
    f( g, 7 );
}

```

- 1) нет ошибок
- 2) есть ошибки компиляции
- 3) есть ошибки выполнения

91. Есть ли ошибки в данном фрагменте программы? Считаем, что все необходимые файлы и директивы включены.

(Если есть ошибки компиляции, то ошибок выполнения нет, поскольку программа не может быть выполнена)

```

template < class C >
int find_num_even_last( const C & c, typename C::value_type v ) {
    int nl = 0, j = 0;
    typename C :: const_iterator p = c.begin ();
    while ( p != c.end () ) {
        if ( *( p += 2 ) == v )
            nl = j;
        j++;
    }
    return nl;
}
int main () {
    vector <int> v = { 1, 2, 3 };
    cout << find_num_even_last < vector<int> > ( v, 2 ) << endl;
}

```

- 1) Нет ошибок
- 2) Есть ошибки компиляции функции *find_num_even_last*
- 3) Есть ошибки компиляции функции *main*
- 4) Есть ошибки выполнения

92. Есть ли ошибки в данном фрагменте программы? Считаем, что все необходимые файлы и директивы включены.
(Если есть ошибки компиляции, то ошибок выполнения нет, поскольку программа не может быть выполнена)

```

template < class C >
int find_num_last( const C & c, typename C::value_type v ) {
    int nl = 0, j = 0;
    typename C::iterator p = c.begin ();
    while ( p != c.end () ) {
        if ( *p++ == v ) nl = j;
        j++;
    }
    return nl;
}

int main () {
    vector <int> v = { 1, 2, 3, 1 };
    cout << find_num_last < vector<int> > ( v, 2 ) << endl;
}

```

- 1) Нет ошибок
- 2) Есть ошибки компиляции функции *find_num_last*
- 3) Есть ошибки компиляции функции *main*
- 4) Есть ошибки выполнения

93. Есть ли ошибки в данном фрагменте программы? Считаем, что все необходимые файлы и директивы включены.
(Если есть ошибки компиляции, то ошибок выполнения нет, поскольку программа не может быть выполнена)

```

template < class X >
X::const_iterator f(const X & x, X::value_type v ) {
    X::const_iterator p = x.end();
    while ( p != x.begin() )
        if ( * -- p == v ) return p;
    return x.end ();
}

int main() {
    list <int> g;
    g.push_back(1);
    f( g, 7 );
}

```

- 1) Нет ошибок
- 2) Есть ошибки компиляции функции *f*
- 3) Есть ошибки компиляции функции *main*
- 4) Есть ошибки выполнения

Классификация грамматик и языков по Хомскому

94. К каким классам относится язык, порождаемый грамматикой? Перечислить все варианты.

$$S \rightarrow AS \mid \varepsilon$$

$$A \rightarrow a$$

- 1) контекстно-свободный
- 2) контекстно-зависимый
- 3) рекурсивно-перечислимый (тип 0)
- 4) регулярный

95. К каким классам относится язык, порождаемый грамматикой? Перечислить все варианты.

$$S \rightarrow ASb \mid \varepsilon$$

$$A \rightarrow a$$

- 1) контекстно-свободный
- 2) контекстно-зависимый
- 3) рекурсивно-перечислимый (тип 0)
- 4) регулярный

96. К каким классам относится язык, порождаемый грамматикой? Перечислить все варианты.

$$S \rightarrow ASa \mid \varepsilon$$

$$A \rightarrow a$$

- 1) контекстно-свободный
- 2) контекстно-зависимый
- 3) рекурсивно-перечислимый (тип 0)
- 4) регулярный

97. К каким классам относится язык, порождаемый грамматикой? Перечислить все варианты.

$$S \rightarrow AbSc \mid Ab$$

$$bA \rightarrow Ab$$

$$Ab \rightarrow ab$$

$$Aa \rightarrow aa$$

- 1) контекстно-свободный
- 2) контекстно-зависимый
- 3) рекурсивно-перечислимый (тип 0)
- 4) регулярный

98. К каким классам относится язык, порождаемый грамматикой? Перечислить все варианты.

$$S \rightarrow ASbS \mid \varepsilon$$

$$A \rightarrow a$$

- 1) контекстно-свободный
- 2) контекстно-зависимый
- 3) рекурсивно-перечислимый (тип 0)
- 4) регулярный

99. К каким классам относится язык, порождаемый грамматикой? Перечислить все варианты.

$$S \rightarrow BSaS \mid \varepsilon$$

$$B \rightarrow b$$

- 1) контекстно-свободный
- 2) контекстно-зависимый
- 3) рекурсивно-перечислимый (тип 0)
- 4) регулярный

100. К каким классам относится язык, порождаемый грамматикой? Перечислить все варианты.

$$S \rightarrow SASb \mid \varepsilon$$

$$A \rightarrow a$$

- 1) контекстно-свободный
- 2) контекстно-зависимый
- 3) рекурсивно-перечислимый (тип 0)
- 4) регулярный

101. К каким классам относится язык, порождаемый грамматикой? Перечислить все варианты.

$$S \rightarrow aSbS \mid bSaS \mid \varepsilon$$

- 1) контекстно-свободный
- 2) контекстно-зависимый
- 3) рекурсивно-перечислимый (тип 0)
- 4) регулярный

102. К каким классам относится язык, порождаемый грамматикой? Перечислить все варианты.

$$S \rightarrow \varepsilon$$

- 1) контекстно-свободный
- 2) контекстно-зависимый
- 3) рекурсивно-перечислимый (тип 0)
- 4) регулярный

103. К каким классам относится язык, порождаемый грамматикой? Перечислить все варианты.

$$S \rightarrow aSaS \mid aSa \mid \varepsilon$$

- 1) контекстно-свободный
- 2) контекстно-зависимый
- 3) рекурсивно-перечислимый (тип 0)
- 4) регулярный

104. К каким классам относится язык, порождаемый грамматикой? Перечислить все варианты.

$$S \rightarrow AA \mid SA \mid \varepsilon$$

$$AA \rightarrow a$$

- 1) контекстно-свободный
- 2) контекстно-зависимый
- 3) рекурсивно-перечислимый (тип 0)
- 4) регулярный

105. Пусть $L = \{b^n \mid n \geq 0\}$.

Укажите такое целое k , что выполняются три условия:

- 1) L удовлетворяет определению языка типа k ;
- 2) L удовлетворяет определению языка типа $k + 1$;
- 3) L не является языком типа $k + 2$.

(В ответе указать одно целое число)

106. Пусть $L = \{b^n ab^n \mid n \geq 0\}$.

Укажите такое целое k , что выполняются три условия:

- 1) L удовлетворяет определению языка типа k ;
- 2) L удовлетворяет определению языка типа $k - 1$;
- 3) L не является языком типа $k + 1$.

(В ответе указать одно целое число)

107. Пусть $L = \{c^n ccc^n \mid n \geq 0\}$.

Укажите такое целое k , что выполняются три условия:

- 1) L удовлетворяет определению языка типа k ;
- 2) L удовлетворяет определению языка типа $k - 1$;
- 3) L не является языком типа $k + 1$.

(В ответе указать одно целое число)

108. Пусть $L = \{a^n cd^n \mid 0 \leq n \leq 5\}$.

Укажите такое целое k , что выполняются три условия:

- 1) L удовлетворяет определению языка типа k ;
- 2) L удовлетворяет определению языка типа $k - 1$;
- 3) L не является языком типа $k + 1$.

(В ответе указать одно целое число)

109. Пусть $L = \{c^n b^n c^n \mid n \geq 0\}$.

Укажите такое целое k , что выполняются три условия:

- 1) L удовлетворяет определению языка типа k ;
- 2) L удовлетворяет определению языка типа $k - 1$;
- 3) L не является языком типа $k + 1$.

(В ответе указать одно целое число)

110. Дана грамматика G :

$$S \rightarrow xSy \mid \varepsilon$$

Укажите такое целое k , что выполняются три условия:

- 1) G не удовлетворяет определению грамматики типа k ;
- 2) G удовлетворяет определению грамматики типа $k - 1$;
- 3) G удовлетворяет определению грамматики типа $k + 1$.

(В ответе указать одно целое число)

111. Дана грамматика G :

$$S \rightarrow cSc \mid \varepsilon \mid a$$

Укажите такое целое $k > 0$, что выполняются три условия:

- 1) G удовлетворяет определению грамматики типа k ;
- 2) G не удовлетворяет определению грамматики типа $k - 1$;
- 3) G не удовлетворяет определению грамматики типа $k + 1$.

(В ответе указать одно целое число)

112. Дана грамматика G :

$$S \rightarrow xSx \mid \varepsilon \mid y$$

Укажите такое целое k , что выполняются три условия:

- 1) G удовлетворяет определению грамматики типа k ;
- 2) G не удовлетворяет определению грамматики типа $k + 1$;
- 3) G удовлетворяет определению грамматики типа $k + 2$.

(В ответе указать одно целое число)

113. Дана грамматика G :

$$S \rightarrow aB \mid Ba \mid \varepsilon$$

$$B \rightarrow Bc \mid c$$

Укажите такое целое k , что выполняются три условия:

- 1) G не удовлетворяет определению грамматики типа $k + 1$;
- 2) G удовлетворяет определению грамматики типа $k - 1$;
- 3) G удовлетворяет определению грамматики типа k .

(В ответе указать одно целое число)

114. Дана грамматика G :

$$S \rightarrow dSdd \mid \varepsilon \mid a$$

Укажите такое целое k , что выполняются три условия:

- 1) G не удовлетворяет определению грамматики типа k ;
- 2) G удовлетворяет определению грамматики типа $k - 1$;
- 3) G удовлетворяет определению грамматики типа $k + 1$.

(В ответе указать одно целое число)

115. Пусть $L = \{ a^n b^{n+1} a^{n+2} \mid n \geq 1 \}$.

Укажите такое целое k , что выполняются три условия:

- 1) L удовлетворяет определению языка типа k ;
- 2) L удовлетворяет определению языка типа $k - 1$;
- 3) L не является языком типа $k + 1$.

(В ответе указать одно целое число)

116. Пусть $L = \{ a^n b^m a^m b^i \mid n, m, i \geq 1 \}$.

Укажите такое целое k , что выполняются три условия:

- 1) L удовлетворяет определению языка типа k ;
- 2) L удовлетворяет определению языка типа $k - 1$;
- 3) L не является языком типа $k + 1$.

(В ответе указать одно целое число)

117. Дана грамматика G :

$$S \rightarrow AB$$

$$A \rightarrow BA \mid \varepsilon$$

$$B \rightarrow \varepsilon$$

Укажите такое целое k , что выполняются три условия:

- 1) G удовлетворяет определению грамматики типа k ;
- 2) G не удовлетворяет определению грамматики типа $k + 1$;
- 3) G удовлетворяет определению грамматики типа $k + 2$.

(В ответе указать одно целое число)

Вывод цепочки по заданной грамматике. Сентенциальные формы

118. Найти самую длинную сентенциальную форму, выводимую в данной грамматике:

$$S \rightarrow aAb \mid Bab$$

$$A \rightarrow a \mid BabB$$

$$B \rightarrow \varepsilon$$

119. Найти самую длинную сентенциальную форму, выводимую в данной грамматике:

$$S \rightarrow AaB \mid b$$

$$A \rightarrow BaB \mid ab$$

$$B \rightarrow \varepsilon$$

120. Найти самую длинную сентенциальную форму, выводимую в данной грамматике:

$$S \rightarrow Dm \mid nb$$

$$D \rightarrow Cab \mid aC$$

$$C \rightarrow \varepsilon$$

121. Найти самую длинную сентенциальную форму, выводимую в данной грамматике:

$$S \rightarrow CaD \mid b$$

$$C \rightarrow DaD \mid ab$$

$$D \rightarrow \varepsilon$$

122. Найти самую длинную сентенциальную форму, выводимую в данной грамматике:

$$S \rightarrow Am \mid nbn$$

$$A \rightarrow Babn \mid aC$$

$$B \rightarrow \varepsilon$$

123. Найти самую длинную сентенциальную форму, выводимую в данной грамматике:

$$S \rightarrow AA \mid nbn$$

$$A \rightarrow BB \mid B \mid b$$

$$B \rightarrow \varepsilon$$

124. Найти самую длинную сентенциальную форму, выводимую в данной грамматике:

$$S \rightarrow BA \mid \varepsilon$$

$$A \rightarrow BB \mid B \mid b$$

$$B \rightarrow \varepsilon$$

125. Найти самую длинную сентенциальную форму, выводимую в данной грамматике:

$$S \rightarrow Em \mid nb$$

$$E \rightarrow Hab \mid aH$$

$$H \rightarrow \varepsilon$$

126. Найти самую длинную сентенциальную форму, выводимую в данной грамматике:

$$S \rightarrow Dm \mid nb$$

$$D \rightarrow Eab \mid aC$$

$$E \rightarrow \varepsilon$$

127. Дана КС-грамматика G. Выпишите цепочку минимальной длины из множества $L(G)$. Если таких цепочек несколько, укажите любую из них. (Пробелов в ответе не должно быть, все буквы латинские, пустая цепочка вводится словом eps)

$$S \rightarrow Xa \mid Cb \mid Fb \mid BD$$

$$X \rightarrow Ca$$

$$B \rightarrow Ca \mid Cb \mid \varepsilon$$

$$C \rightarrow a \mid Da$$

$$D \rightarrow Da \mid Dc$$

$$E \rightarrow b$$

$$F \rightarrow Fb \mid Eb$$

128. Дана КС-грамматика G. Выпишите цепочку минимальной длины из множества $L(G)$. Если таких цепочек несколько, укажите любую из них. (Пробелов в ответе не должно быть, все буквы латинские, пустая цепочка вводится словом eps)

$$S \rightarrow AS \mid bY \mid aaaC$$

$$A \rightarrow Aa \mid Ab$$

$$Y \rightarrow AY \mid CDb$$

$$C \rightarrow AYC \mid aaC \mid \varepsilon$$

$$D \rightarrow b$$

$$E \rightarrow Aa \mid a \mid \varepsilon$$

129. Дана КС-грамматика G . Выпишите цепочку минимальной длины из множества $L(G)$. Если таких цепочек несколько, укажите любую из них. (Пробелов в ответе не должно быть, все буквы латинские, пустая цепочка вводится словом ϵ)

$$S \rightarrow A0 \mid AD \mid Z2 \mid E1$$

$$A \rightarrow A0 \mid 0D$$

$$B \rightarrow A0 \mid \epsilon$$

$$Z \rightarrow Z2 \mid ZD$$

$$D \rightarrow D0 \mid 0 \mid \epsilon$$

$$E \rightarrow E1 \mid 1D$$

130. Дана КС-грамматика G . Выпишите цепочку минимальной длины из множества $L(G)$. Если таких цепочек несколько, укажите любую из них. (Пробелов в ответе не должно быть, все буквы латинские, пустая цепочка вводится словом ϵ)

$$S \rightarrow 1A \mid AB \mid W1$$

$$A \rightarrow 1A \mid 1$$

$$B \rightarrow 0AB \mid 2AB \mid E$$

$$W \rightarrow W0 \mid W1$$

$$D \rightarrow 1A \mid \epsilon$$

$$E \rightarrow 1WE \mid 0E \mid \epsilon$$

131. Дана КС-грамматика G . Выпишите цепочку минимальной длины из множества $L(G)$. Если таких цепочек несколько, укажите любую из них. (Пробелов в ответе не должно быть, все буквы латинские, пустая цепочка вводится словом ϵ)

$$S \rightarrow KCD \mid AC \mid AE$$

$$A \rightarrow aA \mid bA$$

$$K \rightarrow bA \mid Cc$$

$$C \rightarrow aaC \mid \epsilon$$

$$D \rightarrow aA \mid c$$

$$E \rightarrow aD \mid \epsilon$$

132. Дана КС-грамматика G . Выпишите цепочку минимальной длины из множества $L(G)$. Если таких цепочек несколько, укажите любую из них. (Пробелов в ответе не должно быть, все буквы латинские, пустая цепочка вводится словом ϵ)

$$S \rightarrow AC \mid UA \mid dFB \mid dBF$$

$$A \rightarrow ABC \mid AB \mid aB$$

$$B \rightarrow \varepsilon$$

$$C \rightarrow Cc \mid C$$

$$D \rightarrow DBC \mid BC$$

$$U \rightarrow UA \mid B$$

$$F \rightarrow b \mid a$$

133. Дана КС-грамматика G. Выпишите цепочку минимальной длины из множества L(G). Если таких цепочек несколько, укажите любую из них. (Пробелов в ответе не должно быть, все буквы латинские, пустая цепочка вводится словом eps)

$$S \rightarrow ABC \mid aaCD \mid EF \mid bDH$$

$$A \rightarrow aAB \mid bEb \mid \varepsilon$$

$$B \rightarrow bDH$$

$$C \rightarrow bDB \mid A$$

$$D \rightarrow aH \mid b$$

$$E \rightarrow bBA \mid bAC \mid b$$

$$H \rightarrow bEb \mid aCa$$

134. Дана КС-грамматика G. Выпишите цепочку минимальной длины из множества L(G). Если таких цепочек несколько, укажите любую из них. (Пробелов в ответе не должно быть, все буквы латинские, пустая цепочка вводится словом eps)

$$S \rightarrow 0W1 \mid 1Q \mid WA \mid 0RQ$$

$$W \rightarrow 1R \mid RT \mid \varepsilon$$

$$Q \rightarrow TR \mid 0R \mid W$$

$$R \rightarrow 1T \mid 0Q \mid 00$$

$$T \rightarrow 1PQ \mid 0RQ \mid 1T$$

135. Дана КС-грамматика G. Выпишите цепочку минимальной длины из множества L(G). Если таких цепочек несколько, укажите любую из них. (Пробелов в ответе не должно быть, все буквы латинские, пустая цепочка вводится словом eps)

$$S \rightarrow Rt \mid TqR \mid PT \mid VFы$$

$$Q \rightarrow ppP \mid RtT$$

$$P \rightarrow pP \mid t$$

$$T \rightarrow qR \mid tT \mid Pp$$

$$V \rightarrow VV \mid W$$

136. Дана КС-грамматика G. Выпишите цепочку минимальной длины из множества L(G). Если таких цепочек несколько, укажите любую из них. (Пробелов в ответе не должно быть, все буквы латинские, пустая цепочка вводится словом eps)

$$S \rightarrow Kq \mid Mm \mid QQ$$

$$K \rightarrow Nn$$

$$M \rightarrow Mq \mid Om \mid QQ$$

$$N \rightarrow nN \mid mM \mid m \mid \varepsilon$$

$$Q \rightarrow Qq \mid Mm$$

$$P \rightarrow \varepsilon \mid m$$

137. Дана КС-грамматика G. Выпишите цепочку минимальной длины из множества L(G). Если таких цепочек несколько, укажите любую из них. (Пробелов в ответе не должно быть, все буквы латинские, пустая цепочка вводится словом eps)

$$S \rightarrow aA \mid CB$$

$$A \rightarrow aA \mid a \mid D$$

$$B \rightarrow \varepsilon$$

$$C \rightarrow cC \mid cB$$

$$D \rightarrow d \mid \varepsilon$$

138. Дана КС-грамматика G. Выпишите цепочку минимальной длины из множества L(G). Если таких цепочек несколько, укажите любую из них. (Пробелов в ответе не должно быть, все буквы латинские, пустая цепочка вводится словом eps).

$$S \rightarrow aA \mid C$$

$$A \rightarrow aA \mid a$$

$$B \rightarrow C$$

$$C \rightarrow cC \mid cB$$

$$D \rightarrow d \mid \varepsilon$$

139. Дана КС-грамматика G . Выпишите цепочку минимальной длины из множества $L(G)$. Если таких цепочек несколько, укажите любую из них. (Пробелов в ответе не должно быть, все буквы латинские, пустая цепочка вводится словом *eps*).

$$S \rightarrow dA \mid CB$$

$$A \rightarrow dA \mid a \mid D$$

$$B \rightarrow \varepsilon \mid D$$

$$C \rightarrow cC \mid B$$

$$D \rightarrow a \mid \varepsilon$$

Конечные автоматы. Преобразование НКА к ДКА

140. Сколько состояний будет иметь конечный автомат, полученный детерминизацией диаграммы состояний для данной регулярной грамматики?

$$S \rightarrow aA \mid aB$$

$$B \rightarrow bB \mid b$$

$$A \rightarrow aA \mid a$$

141. Сколько состояний будет иметь конечный автомат, полученный детерминизацией диаграммы состояний для данной регулярной грамматики?

$$S \rightarrow aA \mid aB$$

$$B \rightarrow bB \mid \varepsilon$$

$$A \rightarrow aA \mid a$$

142. Сколько состояний будет иметь конечный автомат, полученный детерминизацией диаграммы состояний для данной регулярной грамматики?

$$S \rightarrow aD \mid aB$$

$$B \rightarrow bS \mid aD$$

$$D \rightarrow aD \mid a$$

143. Сколько состояний будет иметь конечный автомат, полученный детерминизацией диаграммы состояний для данной регулярной грамматики?

$$S \rightarrow aA \mid aD$$

$$D \rightarrow bS \mid aA$$

$$A \rightarrow aA \mid \varepsilon$$

144. Сколько состояний будет иметь конечный автомат, полученный детерминизацией диаграммы состояний для данной регулярной грамматики?

$$S \rightarrow aA \mid aB$$

$$B \rightarrow aS$$

$$A \rightarrow aS \mid \varepsilon$$

145. Сколько состояний будет иметь конечный автомат, полученный детерминизацией диаграммы состояний для данной регулярной грамматики?

$$S \rightarrow dD \mid bB$$

$$B \rightarrow bS$$

$$D \rightarrow dS \mid \varepsilon$$

146. Сколько состояний будет иметь конечный автомат, полученный детерминизацией диаграммы состояний для данной регулярной грамматики?

$$S \rightarrow Sd \mid a \mid Xa$$

$$X \rightarrow Sd \mid \varepsilon$$

147. Сколько состояний будет иметь конечный автомат, полученный детерминизацией диаграммы состояний для данной регулярной грамматики?

$$S \rightarrow Sd \mid Ba \mid \varepsilon$$

$$B \rightarrow Sd$$

148. Сколько состояний будет иметь конечный автомат, полученный детерминизацией диаграммы состояний для данной регулярной грамматики?

$$S \rightarrow Sd \mid Ya \mid \varepsilon$$

$$Y \rightarrow Sd$$

149. Сколько состояний будет иметь конечный автомат, полученный детерминизацией диаграммы состояний для данной регулярной грамматики?

$$S \rightarrow Bc \mid \varepsilon$$

$$B \rightarrow Sa \mid Ab$$

$$A \rightarrow Sa$$

150. Сколько состояний будет иметь конечный автомат, полученный детерминизацией диаграммы состояний для заданной регулярной грамматики?

Для построений использовать соответствующие алгоритмы из лекций. ДКА может иметь несколько заключительных состояний, сводить их в одно с помощью нового символа (маркера конца цепочки) не нужно. (Тупиковое состояние ошибки ERR не учитываем)

$$S \rightarrow aS \mid aY$$

$$Y \rightarrow aY \mid aB \mid \varepsilon$$

$$B \rightarrow aB \mid aS$$

151. Сколько состояний будет иметь конечный автомат, полученный детерминизацией диаграммы состояний для заданной регулярной грамматики?

Для построений использовать соответствующие алгоритмы из лекций. ДКА может иметь несколько заключительных состояний, сводить их в одно с помощью нового символа (маркера конца цепочки) не нужно. (Тупиковое состояние ошибки ERR не учитываем)

$$S \rightarrow Sa \mid Xb$$

$$X \rightarrow Xa \mid Ba \mid \varepsilon$$

$$B \rightarrow Bb \mid Sb$$

152. Сколько состояний будет иметь конечный автомат, полученный детерминизацией диаграммы состояний для заданной регулярной грамматики?

Для построений использовать соответствующие алгоритмы из лекций. ДКА может иметь несколько заключительных состояний, сводить их в одно с помощью нового символа (маркера конца цепочки) не нужно. (Тупиковое состояние ошибки *ERR* не учитываем)

$$S \rightarrow aS \mid aA \mid aB$$

$$A \rightarrow aA \mid bB$$

$$B \rightarrow aB \mid a$$

153. Сколько состояний будет иметь конечный автомат, полученный детерминизацией диаграммы состояний для заданной регулярной грамматики?

Для построений использовать соответствующие алгоритмы из лекций. ДКА может иметь несколько заключительных состояний, сводить их в одно с помощью нового символа (маркера конца цепочки) не нужно. (Тупиковое состояние ошибки *ERR* не учитываем)

$$S \rightarrow Sa \mid Ab \mid Za$$

$$A \rightarrow a \mid Sa \mid \varepsilon$$

$$Z \rightarrow Zb \mid Sa$$

154. Сколько состояний будет иметь конечный автомат, полученный детерминизацией диаграммы состояний для заданной регулярной грамматики?

Для построений использовать соответствующие алгоритмы из лекций. ДКА может иметь несколько заключительных состояний, сводить их в одно с помощью нового символа (маркера конца цепочки) не нужно. (Тупиковое состояние ошибки *ERR* не учитываем)

$$S \rightarrow a \mid aA \mid aC$$

$$A \rightarrow aA \mid aC$$

$$C \rightarrow aC \mid a$$

155. Сколько состояний будет иметь конечный автомат, полученный детерминизацией диаграммы состояний для заданной регулярной грамматики?

Для построений использовать соответствующие алгоритмы из лекций. ДКА может иметь несколько заключительных состояний, сводить их в одно с помощью нового символа (маркера конца цепочки) не нужно. (Тупиковое состояние ошибки *ERR* не учитываем)

$$S \rightarrow Sa \mid Zb$$

$$Z \rightarrow a \mid Ba$$

$$B \rightarrow Ba \mid Sb$$

156. Сколько состояний будет иметь конечный автомат, полученный детерминизацией диаграммы состояний для заданной регулярной грамматики?

Для построений использовать соответствующие алгоритмы из лекций. ДКА может иметь несколько заключительных состояний, сводить их в одно с помощью нового символа (маркера конца цепочки) не нужно. (Тупиковое состояние ошибки *ERR* не учитываем)

$$S \rightarrow vT \mid uU$$

$$T \rightarrow vT \mid vS \mid u$$

$$U \rightarrow uU \mid uS \mid v$$

157. Сколько состояний будет иметь конечный автомат, полученный детерминизацией диаграммы состояний для заданной регулярной грамматики?

Для построений использовать соответствующие алгоритмы из лекций. ДКА может иметь несколько заключительных состояний, сводить их в одно с помощью нового символа (маркера конца цепочки) не нужно. (Тупиковое состояние ошибки *ERR* не учитываем)

$$S \rightarrow tT \mid uZ$$

$$T \rightarrow tW \mid uS$$

$$Z \rightarrow uW \mid tS$$

$$W \rightarrow uT \mid tZ \mid u \mid t$$

158. Сколько состояний будет иметь конечный автомат, полученный детерминизацией диаграммы состояний для заданной регулярной грамматики?

Для построений использовать соответствующие алгоритмы из лекций. ДКА может иметь несколько заключительных состояний, сводить их в одно с помощью нового символа (маркера конца цепочки) не нужно. (Тупиковое состояние ошибки *ERR* не учитываем)

$$S \rightarrow Pa \mid Rd \mid Sa$$

$$P \rightarrow Sd \mid a$$

$$R \rightarrow Rd \mid Sd$$

159. Сколько состояний будет иметь конечный автомат, полученный детерминизацией диаграммы состояний для заданной регулярной грамматики?

Для построений использовать соответствующие алгоритмы из лекций. ДКА может иметь несколько заключительных состояний, сводить их в одно с помощью нового символа (маркера конца цепочки) не нужно. (Тупиковое состояние ошибки ERR не учитываем)

$$S \rightarrow X0 \mid 0$$

$$X \rightarrow X0 \mid Y1$$

$$Y \rightarrow S0 \mid Y1 \mid 1$$

160. Сколько состояний будет иметь конечный автомат, полученный детерминизацией диаграммы состояний для заданной регулярной грамматики? Для построений использовать соответствующие алгоритмы из лекций. ДКА может иметь несколько заключительных состояний, сводить их в одно с помощью нового символа (маркера конца цепочки) не нужно. (Тупиковое состояние ошибки ERR не учитываем)

$$S \rightarrow Sb \mid Aa \mid a \mid b$$

$$A \rightarrow Aa \mid Sb \mid a$$

161. Сколько состояний будет иметь конечный автомат, полученный детерминизацией диаграммы состояний для заданной регулярной грамматики? Для построений использовать соответствующие алгоритмы из лекций. ДКА может иметь несколько заключительных состояний, сводить их в одно с помощью нового символа (маркера конца цепочки) не нужно. (Тупиковое состояние ошибки ERR не учитываем)

$$S \rightarrow a \mid bA$$

$$A \rightarrow aA \mid a \mid \varepsilon$$

162. Сколько состояний будет иметь конечный автомат, полученный детерминизацией диаграммы состояний для заданной регулярной грамматики? Для построений использовать соответствующие алгоритмы из лекций. ДКА может иметь несколько заключительных состояний, сводить их в одно с помощью нового символа (маркера конца цепочки) не нужно. (Тупиковое состояние ошибки ERR не учитываем)

$$S \rightarrow Sa \mid Sb \mid Sc \mid \varepsilon$$

Множества *first* и *follow* для нетерминалов КС-грамматики

163. Для данной грамматики выписать в алфавитном порядке (без пробелов и повторов) символы множества *follow* (*B*).

$$S \rightarrow dDS \mid cBA \mid a$$

$$D \rightarrow dD \mid bc \mid \varepsilon$$

$$A \rightarrow aADc \mid bB$$

$$B \rightarrow cB \mid \varepsilon$$

164. Для данной грамматики выписать в алфавитном порядке (без пробелов и повторов) символы множества *follow* (*S*).

$$S \rightarrow aSc \mid bA \mid \varepsilon$$

$$A \rightarrow cSNbA \mid d$$

$$N \rightarrow daN \mid \varepsilon$$

165. Для данной грамматики выписать в алфавитном порядке (без пробелов и повторов) символы множества *follow* (*S*).

$$S \rightarrow aSc \mid dA \mid \varepsilon$$

$$A \rightarrow cSNdA \mid e$$

$$N \rightarrow eaN \mid \varepsilon$$

166. Для данной грамматики выписать в алфавитном порядке (без пробелов и повторов) символы множества *first* (*S*).

$$S \rightarrow aSb \mid Yb \mid a$$

$$Y \rightarrow cSY \mid dd \mid \varepsilon$$

167. Для данной грамматики выписать в алфавитном порядке (без пробелов и повторов) символы множества *first* (*S*).

$$S \rightarrow aSb \mid Yb \mid ab$$

$$Y \rightarrow cSY \mid Bm \mid \varepsilon$$

$$B \rightarrow d \mid \varepsilon$$

168. Для данной грамматики выписать в алфавитном порядке (без пробелов и повторов) символы множества *first* (*S*).

$$S \rightarrow aAc \mid Bc \mid \varepsilon$$

$$A \rightarrow aS$$

$$B \rightarrow bB \mid \varepsilon$$

169. Для данной грамматики выписать в алфавитном порядке (без пробелов и повторов) символы множества *first* (*B*).

$$S \rightarrow aAc \mid Bc \mid \varepsilon$$

$$A \rightarrow aS$$

$$B \rightarrow bB \mid \varepsilon$$

170. Для данной грамматики выписать в алфавитном порядке (без пробелов и повторов) символы множества *first* (*Y*).

$$S \rightarrow aSb \mid Yb \mid ab$$

$$Y \rightarrow cSY \mid Bm \mid \varepsilon$$

$$B \rightarrow d \mid \varepsilon$$

171. Для данной грамматики выписать в алфавитном порядке (без пробелов и повторов) символы множества *first* (*S*).

$$S \rightarrow aSb \mid BYb \mid ab$$

$$Y \rightarrow cSY \mid Bm \mid \varepsilon$$

$$B \rightarrow d \mid \varepsilon$$

172. Для данной грамматики выписать в алфавитном порядке (без пробелов и повторов) символы множества *first* (*S*).

$$S \rightarrow aSb \mid ab \mid AZb$$

$$Z \rightarrow cSZ \mid Am \mid \varepsilon$$

$$A \rightarrow d \mid \varepsilon$$

Перевод в ПОЛИЗ

173. Какова длина ПОЛИЗа (т.е. сколько элементов будет в ПОЛИЗе) для следующего фрагмента программы на Си?

```
if (a > b) a--, b++;
```

174. Какова длина ПОЛИЗа (т.е. сколько элементов будет в ПОЛИЗе) для следующего фрагмента программы на Си?

```
if (a > b) a--; else b++;
```

175. Какова длина ПОЛИЗа (т.е. сколько элементов будет в ПОЛИЗе) для следующего фрагмента программы на Си?

```
if (a++ > b) b = a; else b++;
```

176. Какова длина ПОЛИЗа (т.е. сколько элементов будет в ПОЛИЗе) для следующего фрагмента программы на Си?

```
if (a > b) ; else b++;
```

177. Какова длина ПОЛИЗа (т.е. сколько элементов будет в ПОЛИЗе) для следующего фрагмента программы на Си?

```
if (a > b) a--; else b++;
```

178. Какова длина ПОЛИЗа (т.е. сколько элементов будет в ПОЛИЗе) для следующего фрагмента программы на Си?

```
if (a > b) a--; else ;
```

179. Какова длина ПОЛИЗа (т.е. сколько элементов будет в ПОЛИЗе) для следующего фрагмента программы на Си?

```
while (a > b) a--;
```

180. Какова длина ПОЛИЗа (т.е. сколько элементов будет в ПОЛИЗе) для следующего фрагмента программы на Си?

```
while (a-- > b) b++;
```

181. Какова длина ПОЛИЗа (т.е. сколько элементов будет в ПОЛИЗе) для следующего фрагмента программы на Си?

```
while (a-- > b) b;
```

182. Какова длина ПОЛИЗа (т.е. сколько элементов будет в ПОЛИЗе) для следующего фрагмента программы на Си?

```
if (a) b;
```

183. Какова длина ПОЛИЗа (т.е. сколько элементов будет в ПОЛИЗе) для данного фрагмента программы на языке Си?

```
while (k --) c += k ;
```

184. Какова длина ПОЛИЗа (т.е. сколько элементов будет в ПОЛИЗе) для данного фрагмента программы на языке Си?

```
if (d) ; else c++, b = c ;
```


185. Какова длина ПОЛИЗа (т.е. сколько элементов будет в ПОЛИЗе) для данного фрагмента программы на языке Си?

```
if (x) while (x != c) x--;
```

186. Какова длина ПОЛИЗа (т.е. сколько элементов будет в ПОЛИЗе) для данного фрагмента программы на языке Си?

```
if (!x) c = --x; else x = 5;
```

187. Какова длина ПОЛИЗа (т.е. сколько элементов будет в ПОЛИЗе) для данного фрагмента программы на языке Си?

```
if(z) while ( a != z) z--;
```

188. Какова длина ПОЛИЗа (т.е. сколько элементов будет в ПОЛИЗе) для данного фрагмента программы на языке Си?

```
do x = b = ++c; while (x < 4);
```

189. Какова длина ПОЛИЗа (т.е. сколько элементов будет в ПОЛИЗе) для данного фрагмента программы на языке Си?

```
do x = z = y; --y; while (x);
```

190. Какова длина ПОЛИЗа (т.е. сколько элементов будет в ПОЛИЗе) для данного фрагмента программы на языке Си?

```
for ( ; k < 10; ++k );
```

191. Какова длина ПОЛИЗа (т.е. сколько элементов будет в ПОЛИЗе) для данного фрагмента программы на языке Си?

```
c = c < b ? c : b;
```

192. Какова длина ПОЛИЗа (т.е. сколько элементов будет в ПОЛИЗе) для данного фрагмента программы на языке Си?

```
if (e) e = a + b + c;
```

193. Какова длина ПОЛИЗа (т.е. сколько элементов будет в ПОЛИЗе) для данного фрагмента программы на языке Си?

```
while ( * a * b * c ) d = e+++f;
```

194. Какова длина ПОЛИЗа (т.е. сколько элементов будет в ПОЛИЗе) для данного фрагмента программы на языке Си?

```
if ((a = b)) a = -b;
```

195. Какова длина ПОЛИЗа (т.е. сколько элементов будет в ПОЛИЗе) для данного фрагмента программы на языке Си?

```
while (a = b--, ++c);
```

Синтаксически управляемый перевод

196. Какие из приведенных ниже схем формального перевода задают перевод

$$\tau = \{ \langle a^n b^n, 0^{n-2} 1^{2n} \rangle \mid n \geq 2 \}$$

и при этом реализуемы методом рекурсивного спуска?

Укажите все правильные варианты.

1) $S \rightarrow a \langle 0 \rangle Sb \langle 11 \rangle \mid aab \langle 11 \rangle b \langle 11 \rangle$

2) $S \rightarrow aaCb \langle 11 \rangle b \langle 11 \rangle$
 $C \rightarrow a \langle 0 \rangle Cb \langle 11 \rangle \mid \varepsilon$

3) $S \rightarrow aaA$
 $A \rightarrow a \langle 0 \rangle A \mid C$
 $C \rightarrow b \langle 11 \rangle C \mid \varepsilon$

197. Какие из приведенных ниже схем формального перевода задают перевод

$$\tau = \{ \langle a^n c^n, 0^{n+2} 1^{\lfloor n/2 \rfloor} \rangle \mid n \geq 0 \}$$

и при этом реализуемы методом рекурсивного спуска?

Укажите все правильные варианты.

Целая часть числа x обозначается как $\lfloor x \rfloor$

1) $S \rightarrow a \langle 00 \rangle Sc \langle 1 \rangle \mid \varepsilon \langle 00 \rangle$

2) $S \rightarrow a \langle 0 \rangle Ac \mid \varepsilon \langle 00 \rangle$
 $A \rightarrow a \langle 0 \rangle Sc \langle 1 \rangle \mid \varepsilon \langle 00 \rangle$

3) $S \rightarrow a \langle 0 \rangle Ac \langle 1 \rangle \mid \varepsilon \langle 00 \rangle$
 $A \rightarrow a \langle 0 \rangle Sc \mid \varepsilon \langle 00 \rangle$

198. Какие из приведенных ниже схем формального перевода задают перевод

$$\tau = \{ \langle \beta d \beta^n, a^n b^m \rangle \mid n = |\beta d \beta^R|_a, m = |\beta d \beta^R|_b, \beta \in \{a, b\}^+ \}$$

и при этом реализуемы методом рекурсивного спуска?

Укажите все правильные варианты.

1) $S \rightarrow a \langle aa \rangle Sa \mid bSb \langle bb \rangle \mid a \langle aa \rangle da \mid b \langle bb \rangle db$

2) $S \rightarrow a \langle aa \rangle Ba \mid bBb \langle bb \rangle$
 $B \rightarrow a \langle aa \rangle Ba \mid bBb \langle bb \rangle \mid d$

3) $S \rightarrow a \langle a \rangle A \langle b \rangle a \mid b \langle a \rangle Ab \langle b \rangle$
 $A \rightarrow a \langle a \rangle A \langle b \rangle a \mid b \langle a \rangle Ab \langle b \rangle \mid d$

199. Какие из приведенных ниже схем формального перевода задают перевод

$$\tau = \{ \langle \beta c \beta^R, (ac)^n \rangle \mid n = |\beta|, \beta \in \{a, b\}^+ \}$$

и при этом реализуемы методом рекурсивного спуска?
Укажите все правильные варианты.

1) $S \rightarrow a \langle ac \rangle Sa \mid bSb \langle ac \rangle \mid a \langle ac \rangle ca \mid b \langle ac \rangle cb$

2) $S \rightarrow a \langle ac \rangle Aa \mid b \langle ac \rangle Ab$
 $A \rightarrow a \langle ac \rangle Aa \mid b(ac)Ab \mid c$

3) $S \rightarrow a \langle a \rangle A \langle c \rangle a \mid b \langle a \rangle Ab \langle c \rangle$
 $A \rightarrow a \langle a \rangle A \langle c \rangle a \mid b \langle a \rangle Ab \langle c \rangle \mid c$

200. Какие из приведенных ниже схем формального перевода задают перевод

$$\tau = \{ \langle \beta c \beta^R, ac^n b \rangle \mid n = |\beta|, \beta \in \{a, b\}^+ \}$$

и при этом реализуемы методом рекурсивного спуска?
Укажите все правильные варианты.

1) $S \rightarrow a \langle a \rangle S \langle c \rangle a \mid bSb \langle c \rangle \mid a \langle cb \rangle ca \mid b \langle cb \rangle cb$

2) $S \rightarrow a \langle ac \rangle Ba \mid b \langle ac \rangle Bb$
 $B \rightarrow a \langle c \rangle Ba \mid b \langle c \rangle Bb \mid c \langle b \rangle$

3) $S \rightarrow a \langle ac \rangle A \langle b \rangle a \mid b \langle ac \rangle Ab \langle b \rangle$
 $A \rightarrow a \langle c \rangle Aa \mid b \langle c \rangle Ab \mid c$

201. Какие из приведенных ниже схем формального перевода задают перевод

$$\tau = \{ \langle \beta c \beta^R, b^n ac^n \rangle \mid n = |\beta|, \beta \in \{a, b\}^+ \}$$

и при этом реализуемы методом рекурсивного спуска?
Укажите все правильные варианты.

1) $S \rightarrow a \langle b \rangle S \langle c \rangle a \mid b \langle b \rangle Sb \langle c \rangle \mid a \langle bac \rangle ca \mid b \langle bac \rangle cb$

2) $S \rightarrow a \langle b \rangle A \langle c \rangle a \mid b \langle b \rangle A \langle c \rangle b$
 $A \rightarrow a \langle b \rangle A \langle c \rangle a \mid b \langle b \rangle A \langle c \rangle b \mid c \langle a \rangle$

3) $S \rightarrow a \langle a \rangle A \langle bc \rangle a \mid b \langle a \rangle Ab \langle bc \rangle$
 $A \rightarrow aA \langle bc \rangle a \mid bAb \langle bc \rangle \mid c$

202. Какие из приведенных ниже схем формального перевода задают перевод

$$\tau = \{ \langle a^m b^n 0^{n+2m} 1^{m+2n} \rangle \mid n, m \geq 0 \}$$

и при этом реализуемы методом рекурсивного спуска?
Укажите все правильные варианты.

$$1) S \rightarrow a \langle 00 \rangle S \langle 1 \rangle \mid b \langle 0 \rangle E \langle 11 \rangle \mid \varepsilon \\ E \rightarrow b \langle 0 \rangle E \langle 11 \rangle \mid \varepsilon$$

$$2) S \rightarrow a \langle 00 \rangle ASbB \langle 11 \rangle \mid \varepsilon \\ A \rightarrow a \langle 0 \rangle A \mid \varepsilon \\ B \rightarrow bB \langle 1 \rangle \mid \varepsilon$$

$$3) S \rightarrow \langle 0 \rangle ASB \langle 1 \rangle \mid A \mid B \mid \varepsilon \\ A \rightarrow a \langle 0 \rangle AB \mid \varepsilon \\ B \rightarrow AbB \langle 1 \rangle B \mid \varepsilon$$

203. Какие из приведенных ниже схем формального перевода задают перевод

$$\tau = \{ \langle a^m b^n, 0^{m-2} 1^{n+2} \rangle \mid m > 1, n \geq 0 \}$$

и при этом реализуемы методом рекурсивного спуска?
Укажите все правильные варианты.

$$1) S \rightarrow aaXSB \langle 11 \rangle \mid \varepsilon \\ X \rightarrow a \langle 0 \rangle X \mid \varepsilon \\ B \rightarrow bB \langle 1 \rangle X \mid \varepsilon$$

$$2) S \rightarrow aaA \langle 11 \rangle \\ A \rightarrow a \langle 0 \rangle A \mid bB \langle 1 \rangle \mid \varepsilon \\ B \rightarrow bB \langle 1 \rangle \mid \varepsilon$$

$$3) S \rightarrow aaASB \langle 1 \rangle \mid A \mid B \\ A \rightarrow a \langle 0 \rangle AB \mid \varepsilon \\ B \rightarrow AbB \langle 1 \rangle B \mid \varepsilon$$

204. Какие из приведенных ниже схем формального перевода задают перевод

$$\tau = \{ \langle 0^m 1^n, (*)^{m+n} (-)^{2n+m} \rangle \mid n, m \geq 0 \}$$

и при этом реализуемы методом рекурсивного спуска?
Укажите все правильные варианты.

(Терминальный алфавит состоит из звездочки и тире)

$$1) S \rightarrow D \mid Q \\ D \rightarrow 0 \langle * \rangle D \langle - \rangle \mid 1 \langle * \rangle Q \langle - \rangle \mid \varepsilon \\ Q \rightarrow 1 \langle * \rangle Q \langle - \rangle \mid \varepsilon$$

$$2) S \rightarrow 0 \langle * \rangle S \langle - \rangle \mid 1 \langle * \rangle P \langle - \rangle \mid \varepsilon \\ P \rightarrow 1 \langle * \rangle P \langle - \rangle \mid \varepsilon$$

$$\begin{aligned}
3) \quad & S \rightarrow 0 \langle * \rangle D \mid 1P \langle - \rangle \\
& D \rightarrow 0 \langle * \rangle D \mid P \\
& P \rightarrow 1 \langle * \rangle P \langle - - \rangle \mid \varepsilon
\end{aligned}$$

205. Какие из приведенных ниже схем формального перевода задают перевод

$$\tau = \{ \langle t^m u^{n+m}, p^n q^{2m+n} \rangle \mid m \geq 0, n > 0 \}$$

и при этом реализуемы методом рекурсивного спуска?
Укажите все правильные варианты.

$$\begin{aligned}
1) \quad & S \rightarrow Q \mid T \mid \varepsilon \\
& Q \rightarrow t \langle p \rangle T \langle qq \rangle \mid uT \langle q \rangle \\
& T \rightarrow uT \langle q \rangle \mid \varepsilon \\
2) \quad & S \rightarrow P \mid uT \langle q \rangle \\
& P \rightarrow t \langle p \rangle P \mid T \mid \varepsilon \\
& T \rightarrow uT \langle qq \rangle \mid \varepsilon \\
3) \quad & S \rightarrow tSu \langle qq \rangle \mid u \langle p \rangle T \langle q \rangle \\
& T \rightarrow u \langle p \rangle T \langle q \rangle \mid \varepsilon
\end{aligned}$$

206. Какие из приведенных ниже схем формального перевода задают перевод

$$\tau = \{ \langle a^n b^m a^m c^k, a^{n+m} b^{m+k} \rangle \mid n \geq 1, m, k \geq 0 \}$$

и при этом реализуемы методом рекурсивного спуска?
Укажите все правильные варианты.

$$\begin{aligned}
1) \quad & S \rightarrow aA \langle a \rangle BC \\
& A \rightarrow aA \langle a \rangle \mid \varepsilon \\
& B \rightarrow b \langle a \rangle Ba \langle b \rangle \mid \varepsilon \\
& C \rightarrow c \langle b \rangle C \mid \varepsilon \\
2) \quad & S \rightarrow a \langle a \rangle S \mid a \langle a \rangle AB \\
& A \rightarrow b \langle a \rangle Aa \langle b \rangle \mid \varepsilon \\
& B \rightarrow cB \langle b \rangle \mid \varepsilon \\
3) \quad & S \rightarrow ABC \mid abab \langle aabb \rangle \\
& A \rightarrow a \langle a \rangle A \mid \varepsilon \\
& B \rightarrow b \langle a \rangle Ba \langle b \rangle \mid \varepsilon \\
& C \rightarrow cC \langle b \rangle \mid \varepsilon
\end{aligned}$$

207. Какие из приведенных ниже схем формального перевода задают перевод

$$\tau = \{ \langle a^n b^m a^m b^k, a^{n+m} b^{m+k} \rangle \mid n, m, k \geq 1 \}$$

и при этом реализуемы методом рекурсивного спуска?
Укажите все правильные варианты.

- 1) $S \rightarrow a(a) Ab(a) BabC(bb)$
 $A \rightarrow a(a) A \mid \varepsilon$
 $B \rightarrow b(a) Ba(b) \mid \varepsilon$
 $C \rightarrow bC(b) \mid \varepsilon$
- 2) $S \rightarrow a(a) S \mid a(a) AB(b)$
 $A \rightarrow b(a) Aa(b) \mid ba(ab)$
 $B \rightarrow bB(b) \mid \varepsilon$
- 3) $S \rightarrow ABC \mid abab(aabb)$
 $A \rightarrow a(a) A \mid \varepsilon$
 $B \rightarrow b(a) Ba(b) \mid \varepsilon$
 $C \rightarrow bC(b) \mid \varepsilon$

208. Какие из приведенных ниже схем формального перевода задают перевод

$$\tau = \{ \langle a^n b^k a^{n+k}, a^k b^n \rangle \mid n, k \geq 0 \}$$

и при этом реализуемы методом рекурсивного спуска?
Укажите все правильные варианты.

- 1) $S \rightarrow a(a)Sa(b) \mid A$
 $A \rightarrow b(a)Aa \mid \varepsilon$
- 2) $S \rightarrow aSa(b) \mid A$
 $A \rightarrow b(a)Ba \mid \varepsilon$
 $B \rightarrow bBa(a) \mid \varepsilon$
- 3) $S \rightarrow aSa(b) \mid A \mid C$
 $A \rightarrow bBa(a) \mid \varepsilon$
 $B \rightarrow b(a)Ba \mid \varepsilon$
 $C \rightarrow \varepsilon$
- 4) *нет подходящего варианта*

Основные понятия ООП и систем программирования, применение основных механизмов C++

209. Дано определение термина. Назовите сам термин (строчными буквами, одно слово).

«Специальный метод класса, вызываемый при уничтожении объекта».

210. Дано определение термина. Назовите сам термин (строчными буквами, одно слово).

«Специальный метод класса, вызываемый при создании объекта».

211. Дано определение термина. Назовите сам термин (строчными буквами, два слова через один пробел).

«Класс, содержащий хотя бы одну чистую виртуальную функцию».

212. Дано определение термина. Назовите сам термин (строчными буквами, одно слово).

«Механизм, связывающий вместе код и данные, которыми он манипулирует, и одновременно защищающий их от произвольного доступа со стороны другого кода, внешнего по отношению к рассматриваемому».

213. Дано определение термина. Назовите сам термин (строчными буквами, одно слово).

«Механизм, с помощью которого один объект приобретает свойства другого объекта».

214. Дано определение термина. Назовите сам термин (строчными буквами, одно слово).

«Механизм, позволяющий использовать один и тот же интерфейс для общего класса действий».

215. Дано определение термина. Назовите сам термин (строчными буквами, два слова через один пробел).

«Программа, которую можно использовать отчужденно от ее автора».

216. Дано определение термина. Назовите сам термин (строчными буквами, два слова через один пробел).

«Набор инструментальных средств для поддержки процесса разработки программного продукта в течение всего его жизненного цикла.»

217. Дано определение термина. Назовите сам термин (строчными буквами, одно слово).

«Проверка соответствия программного продукта требованиям спецификации»

218. Дано определение термина. Назовите сам термин (строчными буквами, одно слово).

«Программа для автоматизации поиска ошибок в коде».

219. Дано определение термина из курса «Системы программирования», при этом сам термин не называется:

«набор инструментальных средств для поддержки процесса разработки программного продукта в течение всего его жизненного цикла».

Определите этот термин и вставьте в следующую программу минимальный по длине (по количеству символов) фрагмент кода (без лишних пробелов) так, чтобы этот термин был напечатан.

(Считаем, что все необходимые файлы и директивы включены)

```
class A {
public:
    virtual void h() { cout << "отладчик"; }
    void g() { cout << "редактор связей"; }
};
class B: public A {
public:
    void h() { cout << "система программирования"; }
    void g() { cout << "система автоматизированного тестирования"; }
};
int main() {
    B b;
    A *p = &b;
    p->_____ ;
}
```

220. Дано определение термина из курса «Системы программирования», при этом сам термин не называется:

«набор инструментальных средств для поддержки процесса разработки программного продукта в течение всего его жизненного цикла».

Определите этот термин и вставьте в следующую программу минимальный по длине (по количеству символов) фрагмент кода (без лишних пробелов) так, чтобы этот термин был напечатан.

(Считаем, что все необходимые файлы и директивы включены)

```
class A {
public:
    virtual void f() { cout << "система программирования"; }
    virtual void h() { cout << "отладчик"; }
};
class B: public A {
public:
    void f() { cout << "редактор связей"; }
    virtual void h() const { cout << "система программирования"; }
};
int main() {
    B b;
    A &r=b;
    r._____;
}
```

221. Дано определение термина из курса «Системы программирования», при этом сам термин не называется:

«набор инструментальных средств для поддержки процесса разработки программного продукта в течение всего его жизненного цикла».

Определите этот термин и вставьте в следующую программу минимальный по длине (по количеству символов) фрагмент кода (без лишних пробелов) так, чтобы этот термин был напечатан.

(Считаем, что все необходимые файлы и директивы включены)

```
class A {
public:
    virtual void f() { cout << "система программирования"; }
    virtual void p() { cout << "отладчик"; }
};
class B: public A {
public:
    void f() const { cout << "редактор связей"; }
    virtual void p() { cout << "система программирования"; }
};
int main() {
    B b;
    A * p = & b;
    p ->_____;
}
```

222. Дано определение термина из курса «Системы программирования», при этом сам термин не называется:

«набор инструментальных средств для поддержки процесса разработки программного продукта в течение всего его жизненного цикла».

Определите этот термин и вставьте в следующую программу минимальный по длине (по количеству символов) фрагмент кода (без лишних пробелов) так, чтобы этот термин был напечатан.

(Считаем, что все необходимые файлы и директивы включены)

```
class A {
public:
    virtual void f() const { cout << "отладчик"; }
    virtual void h() { cout << "редактор связей"; }
};
class B: public A {
public:
    void f() { cout << "система программирования"; }
    void h() { cout << "система автоматизированного тестирования"; }
};
int main() {
    B b;
    A & r = b;
    _____; // вставить операнды операции '.'
}
```

223. Дано определение термина из курса «Системы программирования», при этом сам термин не называется:

«набор инструментальных средств для поддержки процесса разработки программного продукта в течение всего его жизненного цикла».

Определите этот термин и вставьте в следующую программу минимальные по длине (по количеству символов) фрагменты кода (без лишних пробелов) так, чтобы этот термин был напечатан.

(Считаем, что все необходимые файлы и директивы включены)

```
class A {
public:
    virtual void h() { cout << "редактор связей"; }
    virtual void g() { cout << "тестирование"; }
};
class B: public A {
public:
    void h() { cout << "отладка"; }
    void g() { cout << "система программирования"; }
};
int main() {
    B b;
    A *p = &b;
    _____->_____ ; // вставить операнды операции '->'
}
```

224. Дано определение термина из курса «Системы программирования», при этом сам термин не называется:

«набор инструментальных средств для поддержки процесса разработки программного продукта в течение всего его жизненного цикла».

Определите этот термин и вставьте в следующую программу минимальный по длине (по количеству символов) фрагмент кода (без лишних пробелов) так, чтобы этот термин был напечатан.

(Считаем, что все необходимые файлы и директивы включены)

```
class A {
public:
    virtual void f() { cout << "отладчик"; }
    void q() { cout << "редактор связей"; }
};
class B: public A {
public:
    void f() { cout << "система программирования"; }
    void q() { cout << "система автоматизированного тестирования"; }
};
int main() {
    B b;
    A *p = &b;
    p->_____;
}
```

225. Дано определение термина из курса «Системы программирования», при этом сам термин не называется:

«набор инструментальных средств для поддержки процесса разработки программного продукта в течение всего его жизненного цикла».

Определите этот термин и вставьте в следующую программу минимальный по длине (по количеству символов) фрагмент кода (без лишних пробелов) так, чтобы этот термин был напечатан.

(Считаем, что все необходимые файлы и директивы включены)

```
class A {
public:
    virtual void q() { cout << "отладчик"; }
    void t() { cout << "редактор связей"; }
};
class B: public A {
public:
    void q() { cout << "система программирования"; }
    void t() { cout << "система автоматизированного тестирования"; }
};
int main() {
    B b;
    A *p = &b;
    p->_____;
}
```

226. Дано определение термина из курса «Системы программирования», при этом сам термин не называется:

«набор инструментальных средств для поддержки процесса разработки программного продукта в течение всего его жизненного цикла».

Определите этот термин и вставьте в следующую программу минимальный по длине (по количеству символов) фрагмент кода (без лишних пробелов) так, чтобы этот термин был напечатан.

(Считаем, что все необходимые файлы и директивы включены)

```
class A{
public:
    virtual void f() { cout << "отладчик"; }
    void t() { cout << "редактор связей"; }
};
class D: public A {
public:
    void f() { cout << "система программирования"; }
    void t() { cout << "система автоматизированного тестирования"; }
};
int main() {
    D d;
    A *p = &d;
    p->_____ ;
}
```

227. Дано определение термина из курса «Системы программирования», при этом сам термин не называется:

«набор инструментальных средств для поддержки процесса разработки программного продукта в течение всего его жизненного цикла».

Определите этот термин и вставьте в следующую программу минимальный по длине фрагмент кода (без лишних пробелов) так, чтобы этот термин был напечатан.

(Считаем, что все необходимые файлы и директивы включены)

```
class D {
public:
    virtual void h() { cout << "отладчик"; }
    void g() { cout << "редактор связей"; }
};
class B: public D {
public:
    void h() { cout << "система программирования"; }
    void g() { cout << "система автоматизированного тестирования"; }
};
int main() {
    B b;
    D *p = &b;
    p->_____ ;
}
```

228. Дано определение термина из курса «Системы программирования», при этом сам термин не называется:

«набор инструментальных средств для поддержки процесса разработки программного продукта в течение всего его жизненного цикла».

Определите этот термин и вставьте в следующую программу минимальный по длине (по количеству символов) фрагмент кода (без лишних пробелов) так, чтобы этот термин был напечатан.

(Считаем, что все необходимые файлы и директивы включены)

```
class A {
public:
    virtual void f() { cout << "отладчик"; }
    void g() { cout << "редактор связей"; }
};
class C: public A {
public:
    void f() { cout << "система программирования"; }
    void g() { cout << "система автоматизированного тестирования"; }
};
int main() {
    C c;
    A *p = &c;
    p->_____ ;
}
```

229. Дано определение термина из курса «Системы программирования», при этом сам термин не называется:

«оптимизация программы, проводимая одновременно с генерацией объектного кода или после нее».

Определите этот термин и вставьте в следующую программу минимальный по длине (по количеству символов) фрагмент кода (без лишних пробелов) так, чтобы этот термин был напечатан.

(Считаем, что все необходимые файлы и директивы включены)

```
struct D {
public:
    virtual void h() { cout << "машинно-независимая оптимизация"; }
};

struct B : public D {
public:
    void h() { cout << "машинно-зависимая оптимизация"; }
};
int main(){
    B b;
    D *p = &b;
    p->_____ ;
}
```

230. Дано определение термина из курса «Системы программирования», при этом сам термин не называется:

«набор инструментальных средств для поддержки процесса разработки программного продукта в течение всего его жизненного цикла».

Определите этот термин и вставьте в следующую программу минимальный по длине (по количеству символов) фрагмент кода (без лишних пробелов) так, чтобы этот термин был напечатан.

(Считаем, что все необходимые файлы и директивы включены)

```
struct A {  
public:  
    void f() { cout << "справочная система"; }  
    virtual void g() { cout << "система программирования"; }  
};  
struct B : public A {  
public:  
    void f() { cout << "профилировщик"; }  
    void g() { cout << "система управления версиями"; }  
};  
int main(){  
    B b;  
    A *p = &b;  
    b._____;  
}
```

ОТВЕТЫ К ТЕСТАМ

Ответы к тестам по теме «Перегрузка функций. Алгоритм выбора наилучшим образом подходящей функции (best matching)»

Номер теста	Ответ
1	1)
2	1)
3	5)
4	5)
5	1)
6	5)
7	5)
8	1)
9	4)
10	2)
11	5)
12	1)
13	1)
14	1)
15	1)
16	2), 3), 4), 5)
17	3)
18	4)
19	5)
20	1)
21	4)
22	5)
23	4)

Ответы к тестам по теме «Перегрузка операций»

Номер теста	Ответ
24	<i>A operator + (A) {}</i>
25	<i>A operator + (A) const {}</i>
26	<i>A operator * (A) {}</i>
27	<i>A operator - () const {}</i>
28	<i>A operator -- () {}</i>
29	<i>A operator ++ (int) {}</i>
30	<i>A operator -- () {}</i>
31	<i>T * operator -> () { return p; }</i>
32	<i>operator int () {}</i>
33	<i>A operator () () {}</i>

Ответы к тестам по теме «Конструкторы, деструкторы, исключения в C++»

Номер теста	Ответ
34	<i>throw A();</i>
35	<i>throw A();</i>
36	<i>throw a;</i>
37	<i>throw A();</i>
38	<i>throw A();</i>
39	<i>throw A();</i>
40	<i>throw a;</i>
41	<i>throw r;</i>
42	<i>throw B();</i>
43	<i>throw b;</i>
44	<i>k += <u>3</u>;</i>
45	<i>k += <u>3</u>;</i>
46	<i>n += <u>2</u>;</i>
47	<i>k += <u>3</u>;</i>
48	<i>n = n + <u>4</u>;</i>
49	<i>n = n + <u>5</u>;</i>
50	<i>n += <u>1</u>;</i>
51	<i>n = n + <u>3</u>;</i>
52	<i>k += <u>1</u>;</i>
53	<i>k += <u>2</u>;</i>
54	1)
55	4)
56	1)

Ответы к тестам по теме «Механизм динамического полиморфизма в C++»

Номер теста	Ответ
57	$a.\underline{f}()$;
58	$a.\underline{A::f}()$;
59	$c.\underline{f}()$;
60	$a.\underline{f}()$;
61	$a.\underline{f}()$;
62	$a \rightarrow \underline{f}()$;
63	$a \rightarrow \underline{f}()$;
64	$a \rightarrow \underline{A::f}()$;
65	$a \rightarrow \underline{A::g}()$;
66	$a \rightarrow \underline{g}()$;
67	3)

Ответы к тестам по теме «Механизм RTTI в C++»

Номер теста	Ответ
68	1)
69	2)
70	2)
71	1)
72	2)
73	1)
74	1)
75	2)
76	3)
77	4)
78	3)
79	3)
80	5)

Ответы к тестам по теме «STL, шаблоны, турепате в C++»

Номер теста	Ответ
81	1)
82	2)
83	2)
84	2)
85	2)
86	1)
87	2)
88	3)
89	2)
90	2)
91	4)
92	2)
93	2)

Ответы к тестам по теме «Классификация грамматик и языков по Хомскому»

Номер теста	Ответ
94	1), 2), 3), 4)
95	1), 2), 3)
96	1), 2), 3), 4)
97	2), 3)
98	1), 2), 3)
99	1), 2), 3)
100	1), 2), 3)
101	1), 2), 3)
102	1), 2), 3), 4)
103	1), 2), 3), 4)
104	1), 2), 3), 4)
105	2
106	2
107	3
108	3
109	1
110	1
111	2
112	0
113	2
114	1
115	1
116	2
117	0

**Ответы к тестам по теме «Вывод цепочки по заданной грамматике.
Сентенциальные формы»**

Номер теста	Ответ
118	<i>aBabBb</i>
119	<i>BaBaB</i>
120	<i>CaBm</i>
121	<i>DaDaD</i>
122	<i>Babnm</i>
123	<i>BBBB</i>
124	<i>BBB</i>
125	<i>Habm</i>
126	<i>Eabm</i>
127	<i>ab</i>
128	<i>aaa <или> bbb</i>
129	<i>0</i>
130	<i>1</i>
131	<i>cc</i>
132	<i>a</i>
133	<i>aab</i>
134	<i>1</i>
135	<i>ttp</i>
136	<i>nq</i>
137	<i>a <или> c</i>
138	<i>aa</i>
139	<i>eps</i>

Ответы к тестам по теме «Конечные автоматы. Преобразование НКА к ДКА»

Номер теста	Ответ
140	4
141	4
142	3
143	3
144	2
145	3
146	3
147	2
148	2
149	3
150	3
151	3
152	5
153	6
154	2
155	5
156	8
157	6
158	6
159	6
160	3
161	4
162	1

**Ответы к тестам по теме «Множества first и follow для нетерминалов
КС-грамматики»**

Номер теста	Ответ
163	<i>Abcd</i>
164	<i>Bcd</i>
165	<i>Cde</i>
166	<i>Abcd</i>
167	<i>Abcdm</i>
168	<i>Abc</i>
169	<i>B</i>
170	<i>Cdm</i>
171	<i>Abcdm</i>
172	<i>Abcdm</i>

Ответы к тестам по теме «Перевод в ПОЛИЗ»

Номер теста	Ответ
173	11
174	13
175	15
176	10
177	13
178	10
179	10
180	11
181	10
182	5
183	10
184	12
185	13
186	15
187	13
188	14
189	14
190	14
191	12
192	11
193	17
194	10
195	11

Ответы к тестам по теме «Синтаксически управляемый перевод»

Номер теста	Ответ
196	2)
197	2)
198	2)
199	2)
200	2), 3)
201	2)
202	1)
203	2)
204	2)
205	3)
206	1)
207	1)
208	4)

Ответы к тестам по теме «Основные понятия ООП и систем программирования»

Номер теста	Ответ
209	деструктор
210	конструктор
211	абстрактный класс
212	инкапсуляция
213	наследование
214	полиморфизм
215	программный продукт
216	система программирования
217	верификация
218	отладчик
219	$p \rightarrow \underline{h()};$
220	$r . \underline{A::f()};$
221	$p \rightarrow \underline{f()} <или> p();$
222	$b . \underline{f()};$
223	$p \rightarrow \underline{q()};$
224	$p \rightarrow \underline{f()};$
225	$p \rightarrow \underline{q()};$
226	$p \rightarrow \underline{f()};$
227	$p \rightarrow \underline{h()};$
228	$p \rightarrow \underline{f()};$
229	$p \rightarrow \underline{h()};$
230	$b . \underline{A::q()};$

Учебное издание

ВОЛКОВА Ирина Анатольевна
ВЫЛИТОК Алексей Александрович
КАРПОВ Леонид Евгеньевич

ТЕСТЫ ПО КУРСУ «СИСТЕМЫ ПРОГРАММИРОВАНИЯ»
(Объектно-ориентированное программирование на Си++
и методы трансляции на основе формальных грамматик и автоматов)

Учебное пособие для студентов II курса

Издательство «МАКС Пресс» Главный редактор:
Е.М. Бугачева Обложка: *А.В. Кононова*

Напечатано с готового оригинал-макета Подписано
в печать 15.10.2024 г. Формат 70x100 1/16.
Усл.печ.л. 7,5. Тираж 50 экз. Заказ 166.

Издательство ООО «МАКС Пресс» Лицензия
ИД N 00510 от 01.12.99 г.

119992, ГСП-2, Москва, Ленинские горы,
МГУ им. М.В. Ломоносова, 2-й учебный корпус, 527 к. Тел.
8(495)939-3890/91. Тел./Факс 8(495)939-3891.

Отпечатано в полном соответствии с качеством предоставленных
материалов в ООО «Фотоэксперт» 109316, г. Москва, Волгоградский
проспект, д. 42, корп. 5, эт. 1, пом. I, ком. 6.3-23Н