Ф.И.О._____№ группы_____

NB! **Во всех задачах, где это требуется, предполагается наличие необходимых включений .**

1	2	3	4	5	6	7	8	9	10

1. В приведенной ниже программе есть ошибки компиляции. ОБЪЯСНИТЬ, в чем они заключаются. Внести исправления **двумя различными способами**, модифицируя **только класс A**.

```
struct A {
      A(A && rvr) { std::cout << "Move_Constr\n"; }
      ~A() { std::cout << "Destr "; }
};
int main() {
      try { A a, b(a);
            a = b;
            throw a;
      }
      catch(...) { std::cout << "Catch(...) \n"; }
}
```

2. Указать, какие ошибки компиляции есть в приведенной ниже программе. Модифицировать **только** функцию **main**(), **ничего не удаляя** каким-либо образом, чтобы ошибок не было.
Что будет напечатано в результате работы получившейся правильной программы?

```
void f(int i, long int li) { std::cout << "::f()\n"; }
int x = 0;
namespace NS1 {   int x = 1;
                  void f(bool b, long double ld) { std::cout << "NS1::f()\n"; }
}
namespace NS2 { int x = 2;
                void f(char c, double d) { std::cout << "NS2::f()\n"; }
}
using namespace NS2;
int main() {
      using namespace NS2;
      f(1L, 1.0F);
      f(x, 3.5L);
}
```

3. Дана иерархия классов и функции ***handle*** и ***main.*** Всего в классах B и D должны быть определены **ровно шесть** методов, **два** из которых – деструкторы. Каждый метод печатает на экран свою уникальную цифру (1, 2, 3, 4, 5 или 6), например { cout << "1 "; }. Результирующая программа должна корректно компилироваться, и в результате выполнения печатать на экран: **1 2 3**
4 2 5 6 3

```
class B {
public:




};
class D : public B {
public:




};
```

```
void handle (B* pb) {
    pb -> a();
    pb -> b();
    D* pd = dynamic_cast<D*>(pb);
    if (pd) { pd -> c(); }
    delete pb;
    std::cout << std::endl;
}

int main() {
    handle (new B()); // 1 2 3
    handle (new D()); // 4 2 5 6 3
}
```

4. Описать функцию, возвращающую итератор максимального элемента непустого контейнера STL, хранящего числовые данные. **Ответ:**

5. Может ли класс, производный от **не**абстрактного класса (не путать с АТД) быть абстрактным? Ответ **обосновать** (да/нет – не засчитывается). **Ответ:**

6. Дана заготовка грамматики $G_{optimize}$ с действиями по переводу подмножества логических выражений над алфавитом $\{ x, t, f, \vee \}$ в эквивалентное выражение из одного символа, где t и f означают истину и ложь соответственно, x – логическая переменная, $\vee$ – дизъюнкция.

$S \to f \vee S \mid x \vee A \mid t \ \langle cout << ' t '; \rangle$
$A \to x \vee A \mid f \mid t$

Вставить в $G_{optimize}$ недостающие действия вида $\langle cout << 'символ'; \rangle$ так, чтобы в процессе рекурсивного спуска печаталось односимвольное выражение, эквивалентное исходному (т.е. оптимальный по длине ПОЛИЗ).
Примеры: $t \to t$; $f \vee x \vee f \to x$; $f \vee f \vee t \to t$; $x \vee f \to x$; $x \vee x \vee t \to t$. **Ответ:**

7. Дана КС-грамматика G :

$S \to AB \mid AC$
$A \to Aa \mid c$
$B \to b$
$C \to c$

(**а**) Почему метод рекурсивного неприменим к G или, по-другому – почему не выполняется критерий *LL(1)*-грамматики ? **Ответ:**

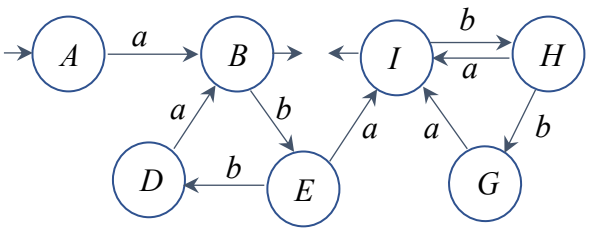
(**б**) Построить эквивалентную КС-грамматику $G_{LL(1)}$, к которой метод рекурсивного применим (выполняется критерий *LL(1)*-грамматики). **Ответ:**

8. Построить грамматику G_0, порождающую язык $\{ \varepsilon \}$ и удовлетворяющую трем условиям:
1) G_0 не является грамматикой типа 1;
2) G_0 не является грамматикой типа 2;
3) количество правил вывода в G_0 равно двум.

Ответ:

9. По заданному конечному автомату $\mathcal{A}$ (на рисунке) построить эквивалентный ДКА $\mathcal{A}_{min}$ с минимальным числом состояний (вершин). Пояснить, каким способом построено решение.
Ответ:

$\mathcal{A}$:



10. Что такое *система программирования* ? **Ответ:**