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() { std::cout << "Constr "; }
      A& operator=(A && a) { return *this; }
};
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 a = 0;
namespace NS1 {   void f(bool b, long double ld) { std::cout << "NS1::f()\n"; }
                  int a = 1;
}
namespace NS2 {   void f(long int la, const char * cstr) { std::cout << "NS2::f()\n";}
                  int a = 2;
}
using namespace NS2;
int main() {
      using namespace NS1;
      f(a, "abc");
      f(true, 1.5f);
}
```

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**
1 4 5 6 3

```
class B {
public:




};
class D : public B {
public:




};
```

```
void handle(const B& rb) {
  rb.a();
  rb.b();
  if (typeid(rb) == typeid(D)) {
    dynamic_cast<const D&> (rb).c();
  }
}

int main() {
  handle(B()); // 1 2 3
  std::cout << std::endl;
  handle(D()); // 1 4 5 6 3
  std::cout << std::endl;
}
```

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

5. Может ли класс, производный от **не**полиморфного класса быть полиморфным? 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,* ∧ } в эквивалентное выражение из одного символа, где *t* и *f* означают истину и ложь соответственно, *x* – логическая переменная, ∧ – конъюнкция.

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

$A \to x \wedge A \mid f \mid t$

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

7. Дана КС-грамматика *G* :

$S \to ABC$
$A \to Ca \mid C$
$B \to Bb \mid b$
$C \to c$

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

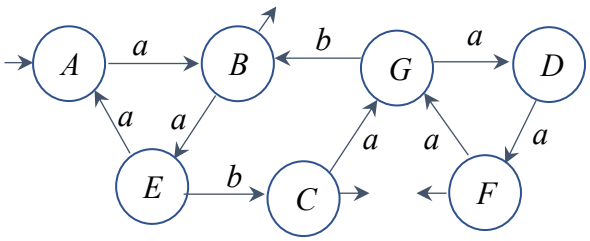
(**б**) Построить эквивалентную КС-грамматику $G_{LL(1)}$, к которой метод рекурсивного применим (выполняется критерий *LL(1)*-грамматики). **Ответ:**

8. Построить грамматику G_1, порождающую язык { *aa* } и удовлетворяющую трем условиям:
1) G_1 не является грамматикой типа 2;
2) G_1 не является грамматикой типа 3;
3) количество правил вывода в G_1 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 *программный продукт* ? **Ответ:**