

1	2	3	4	5		6	7	8	9	10

1. Основные задачи лексического анализатора. Привести два-три примера ситуаций, в которых лексический анализатор выдаст ошибку.
2. Библиотека STL: Дать определение итератора. Чем различаются прямые и обратные итераторы? Привести пример использования обратного итератора.
3. Общая схема работы компилятора.
4. Дать определение недостижимого символа грамматики. Привести пример грамматики, содержащей недостижимый символ.
5. Привести примеры использования статической, динамической и стековой памяти.
6. Определить тип по Хомскому заданной грамматики $G = (VT, VN, P, S)$. Определить порождаемый язык и тип языка по Хомскому.

$$\begin{aligned}
 P: \quad S &\rightarrow AB\perp \\
 AB &\rightarrow aABcB \mid X \\
 Xc &\rightarrow cX \\
 XB &\rightarrow bX \\
 X\perp &\rightarrow cb\perp
 \end{aligned}$$

7. Написать грамматику для языка L_1 . Вставить в нее действия по переводу цепочек языка L_1 в соответствующие цепочки языка L_2 (в ходе анализа цепочек языка L_1 методом рекурсивного спуска). В качестве действий допустимо только использование оператора cout <<

$$\begin{aligned}
 L_1 &= \{ a^n \alpha b^n, \alpha \in (c, d)^+, n \geq 0 \} \\
 L_2 &= \{ 0^m 1^{n+k}, \text{ где } m \text{ — количество символов } c \text{ в цепочке } \alpha, \\
 &\quad k \text{ — количество символов } d \text{ в цепочке } \alpha \}
 \end{aligned}$$

8. Построить ДС, соответствующую заданной P -грамматике $G = (VT, VN, P, S)$.

$$\begin{aligned}
 P: \quad S &\rightarrow Sa \mid Aa \mid Sb \mid a \mid b \\
 A &\rightarrow Sa \mid Ab \mid a
 \end{aligned}$$

Если ДС задает НКА, то по алгоритму преобразования НКА в КА построить КА и написать соответствующую ему грамматику, иначе написать анализатор по заданной грамматике.

9. Дана КС-грамматика $G = (VT, VN, P, S)$. Определить, применим ли к ней метод рекурсивного спуска. Ответ обосновать.

$$\begin{aligned}
 S &\rightarrow aSB \mid bAf \mid \varepsilon \\
 A &\rightarrow bAc \mid cS \\
 B &\rightarrow cB \mid d
 \end{aligned}$$

10. Записать на ПОЛИЗе фрагмент программы на С:

```
do { x = (a + b) / x * ((-a) / (b + x) - 3); S += x; } while (x > c);
```